INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2013; **72**:1219–1243
Published online 11 February 2013 in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.3777

Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations

Joan Baiges[1,2,*,†], Ramon Codina[2,1] and Sergio Idelsohn[3,1]

[1]*Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Edifici C1, Campus Nord UPC C/ Gran Capità S/N 08034 Barcelona, Spain*
[2]*Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain*
[3]*Institució Catalana de Recerca i Estudis Avançats, Barcelona, Spain*

SUMMARY

In this paper, we present an explicit formulation for reduced-order models of the stabilized finite element approximation of the incompressible Navier–Stokes equations. The basic idea is to build a reduced-order model based on a proper orthogonal decomposition and a Galerkin projection and treat all the terms in an explicit way in the time integration scheme, including the pressure. This is possible because the reduced model snapshots do already fulfill the continuity equation. The pressure field is automatically recovered from the reduced-order basis and solution coefficients. The main advantage of this explicit treatment of the incompressible Navier–Stokes equations is that it allows for the easy use of hyper-reduced order models, because only the right-hand side vector needs to be recovered by means of a gappy data reconstruction procedure. A method for choosing the optimal set of sampling points at the discrete level in the gappy procedure is also presented. Numerical examples show the performance of the proposed strategy.

Received 12 June 2012; Revised 7 January 2013; Accepted 15 January 2013

KEY WORDS: reduced-order modeling; Navier–Stokes; finite element; explicit; POD: proper orthogonal decomposition; stabilized method

1. INTRODUCTION

Reduced-order models (ROMs) applied to numerical design in engineering are a tool that is receiving increasing attention in the computational mechanics community. The key feature of ROMs is their capability for drastically reducing the computational cost of numerical simulations, while maintaining a sufficient accuracy from the engineering point of view. This has led many researchers to apply ROMs to several engineering problems such as circuit design [1], multiscale modeling in solid mechanics [2], or metal forming processes [3].

Reduced-order models are also particularly interesting for computational fluid dynamics, where they have been applied to model the Navier–Stokes equations [4–10] and to applications such as shape optimization [11–14] and flow control [15–17].

One of the issues of the traditional approach to ROMs is that the computational cost of solving the reduced-order problem is not reduced in such a drastic way as one would expect when going from a full-order model (FOM) with thousands of degrees of freedom to a reduced system with only tens of degrees of freedom. The cause for this is that in order to solve the reduced-order equations, the full-order equations need to be built first and then projected onto the reduced-order subspace. Although this allows to save the cost of finding the solution of the full-order system of equations,

*Correspondence to: Joan Baiges, Dept. Resistència de Materials i Estructures a l'Enginyeria, Universitat Politècnica de Catalunya, Barcelona, Spain.

[†]E-mail: jbaiges@cimne.upc.edu

it is inconvenient for nonlinear problems where the full-order equations vary from iteration to iteration and consequently need to be continuously rebuilt.

Recently, a general approach for dealing with the non-linear terms of ROMs has appeared [18–23], giving rise to what are known as hyper-ROMs. In these methods, the nonlinear and parameter-dependent terms are recovered by means of a least-squares procedure from a series of sampling points where the function to be approximated is computed. These procedures allow to effectively reduce the amount of computations required to build the reduced-order system and result in a ROM whose computational cost is directly proportional to its number of degrees of freedom.

However, applying this kind of method to the Navier–Stokes equations is not straightforward, because not only a reduced-order basis is required for reconstructing the right-hand side of the system of equations but also the associated matrix needs to be rebuilt. The reconstruction of the left-hand side matrix (or of the result of multiplying the matrix times the ROM basis functions) has been successfully carried out by using hyper-reduced approaches [19, 20, 24]. However, a less costly method in terms of both memory (specially if a specific reduced-order basis for the matrix reconstruction needs to be stored) and amount of computations can be obtained if a strategy which avoids the reconstruction of the matrix is devised. This is the reason why we propose an explicit formulation for the reduced-order Navier–Stokes equations, which yields an easy-to-reconstruct linear matrix. The basic idea is to build a ROM based on a proper orthogonal decomposition and a Galerkin projection and treat all the terms in an explicit way in the time integration scheme. This results in a ROM where only the right-hand side of the system needs to be rebuilt by using a gappy data reconstruction approach at each time step. This is possible because the reduced model snapshots do already fulfill the stabilized continuity equation. As we will show, the pressure field can be automatically recovered from the reduced-order basis and solution coefficients.

When using hyper-ROMs, the quality of the recovered right-hand side vector can very strongly depend on the selected sampling points. A method for choosing the sampling points is presented which is particularly convenient for the sampling of the right-hand side of the Navier–Stokes system of equations. The method consists of choosing the sampling points such that the distance between the right-hand side snapshots and the recovered snapshots is minimized, with the restriction that the coordinates of sampling points must coincide with the coordinates of the finite element mesh nodes.

The paper is organized as follows. In Section 2, we present the finite element approximation of the incompressible Navier–Stokes equations and the stabilizing terms which allow to use equal velocity-pressure interpolations and stabilize the convective term. In Section 3, we describe ROM strategies and the proper orthogonal decomposition method. We also introduce HROMs and we present a new strategy for choosing sampling points in gappy data reconstruction processes. In Section 4, we describe the proposed explicit formulation for the incompressible Navier–Stokes equations and we apply it to ROMs and hyper-reduced models. Finally, some numerical examples show the performance of the proposed methods in Section 5. Some remarks and conclusions close the paper in Section 6.

2. FINITE ELEMENT APPROXIMATION OF THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

2.1. Problem statement

Let us consider the transient incompressible Navier–Stokes equations, which consist of finding $\boldsymbol{u} : \Omega \times (0,T) \longrightarrow \mathbb{R}^d$ and $p : \Omega \times (0,T) \longrightarrow \mathbb{R}$ such that

$$\begin{aligned} \partial_t \boldsymbol{u} - \nu \Delta \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p &= \boldsymbol{f} \quad \text{in } \Omega, \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } \Omega, \\ \boldsymbol{u} &= \bar{u} \quad \text{on } \Gamma_D, \\ -p\boldsymbol{n} + \nu \boldsymbol{n} \cdot \nabla \boldsymbol{u} &= \boldsymbol{0} \quad \text{on } \Gamma_N. \end{aligned}$$

DOI: 10.1002/fld

for $t > 0$, where $\partial_t \boldsymbol{u}$ is the local time derivative of the velocity field. $\Omega \subset \mathbb{R}^d$ is a bounded domain, with $d = 2, 3$, ν is the viscosity, and $\boldsymbol{f}$ the given source term. Appropriate initial and boundary conditions have to be appended to this problem.

Let now be $V = H^1(\Omega)^d$ and $V_0 = \{\boldsymbol{v} \in V | \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_D\}$. Let also be $Q = L^2(\Omega)$ and $\mathcal{D}'(0, T; Q)$ be the distributions in time with values in Q. The variational problem consists of finding $[\boldsymbol{u}, p] \in L^2(0, T; V) \times \mathcal{D}'(0, T; Q)$ such that

$$(\boldsymbol{v}, \partial_t \boldsymbol{u}) + B([\boldsymbol{v}, q], [\boldsymbol{u}, p]) = \langle \boldsymbol{v}, \boldsymbol{f} \rangle \quad \forall [\boldsymbol{v}, q] \in V \times Q, \tag{1}$$

with

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{on } \Gamma_D,$$

where

$$B([\boldsymbol{v}, q], [\boldsymbol{u}, p]) := \langle \boldsymbol{v}, \boldsymbol{u} \cdot \nabla \boldsymbol{u} \rangle + \nu(\nabla \boldsymbol{v}, \nabla \boldsymbol{u}) - (p, \nabla \cdot \boldsymbol{v}) + (q, \nabla \cdot \boldsymbol{u}).$$

Here and in the succeeding text, $(\cdot, \cdot)$ denotes the L^2 product in Ω. In general, the integral of two functions g_1 and g_2 over a domain ω will be denoted by $\langle g_1, g_2 \rangle_\omega$, the $L^2(\omega)$ inner product by $(\cdot, \cdot)_\omega$ and the norm in a function space X by $\| \cdot \|_X$, with the simplifications $\| \cdot \|_{L^2(\Omega)} \equiv \| \cdot \|$ and $(\cdot, \cdot)_\Omega \equiv (\cdot, \cdot)$.

2.2. *Stabilized finite element approximation*

Let $\mathcal{P}_h = \{K\}$ be a finite element partition of Ω from which we construct the finite element spaces $V_h \subset V, V_{h,0} \subset V_0$ and $Q_h \subset Q$. It is well known that when the viscosity coefficient ν in (1) is small, the Galerkin method fails and stabilized finite element methods need to be used. On the other hand, the velocity and pressure spaces V_h and Q_h need to fulfill an inf-sup condition

$$\inf_{p_h \in Q_h} \sup_{\boldsymbol{u}_h \in V_h} \frac{(p_h, \nabla \cdot \boldsymbol{u}_h)}{||\boldsymbol{u}_h||_{V_h} ||p_h||_{Q_h}} \geqslant C > 0, \tag{2}$$

in order for the discrete version of the problem defined in (1) to be well-posed. Boundary conditions need to be appended to (2). These issues motivate the use of stabilized finite element formulations which, on the one hand, deal with the stability problems due to the convective nature of problem (1) and on the other hand, allow us to circumvent the inf-sup condition (2) and to freely choose the interpolation spaces V_h and Q_h (in particular, stabilized formulations allow equal interpolation spaces for the velocity and the pressure, which are otherwise prevented by the inf-sup condition).

The stabilized formulations we use are derived from a multiscale splitting of the spaces for the unknowns $\boldsymbol{u}$ and p into the finite element part of the solution and the subscales (see [25] where the original variational multiscale method was developed). This general approach can be used to deal with several kinds of problems for which the Galerkin method is unstable. When applied to the incompressible Navier–Stokes equations, the formulation we use reads as follows: for each t, find $\boldsymbol{u}_h(t) \in V_h$, $p_h(t) \in Q_h$ such that

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + B([\boldsymbol{v}_h, q_h], [\boldsymbol{u}_h, p_h]) + \sum_K \tau_K (\boldsymbol{u}_h \cdot \nabla \boldsymbol{v}_h + \nu \Delta \boldsymbol{v}_h + \nabla q_h, \boldsymbol{r}([\boldsymbol{u}_h, p_h]))_K = \langle \boldsymbol{v}_h, \boldsymbol{f} \rangle, \tag{3}$$

for all $\boldsymbol{v}_h \in V_{h,0}$, $q_h \in Q_h$. Boundary conditions need to be appended to this problem. In (3),

$$\boldsymbol{r}([\boldsymbol{u}_h, p_h]) = \partial_t \boldsymbol{u}_h - \nu \Delta \boldsymbol{u}_h + \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h + \nabla p_h - \boldsymbol{f}, \tag{4}$$

is the residual of the momentum equation, $(\cdot, \cdot)_K$ is used to denote the L^2 product in element K, and τ_K is the stabilization parameter

$$\tau_K = \left(c_1 \frac{\nu}{h^2} + c_2 \frac{|\boldsymbol{u}_h|_K}{h} \right)^{-1},$$

DOI: 10.1002/fld

where $|\boldsymbol{u}_h|_K$ is the mean velocity modulus in element K, h is the element size, and c_1 and c_2 are stabilization constants. We take $c_1 = 4$ and $c_2 = 2$ for linear elements, which can be justified as in [26]. As explained [27], a more accurate method can be obtained replacing $\boldsymbol{r}([\boldsymbol{u}_h, p_h])$ in (3) by its projection orthogonal to V_h.

An important feature of the stabilized form is that it is consistent, that is, if the exact solution for the problem is replaced, (3) holds. This is due to the fact that the stabilization terms depend on the residual, and the residual vanishes when the exact solution is replaced in the discrete stabilized weak form of the problem. Consistency is an important property that the ROM for the stabilized equations should retain. Details for the motivation of the formulation described and stability and convergence properties can be found in [26].

A discretization scheme for approximating the time derivatives needs to be added to the formulation (3). An important point when solving the incompressible Navier–Stokes equations is that this scheme needs to be *implicit* at least in the pressure term because no pressure time derivative appears in (3), which would allow us to recover p at a given time step from the values of the pressure at the previous step. Plus, if an explicit time integration scheme is used for the terms involving the velocity unknowns, there is no guarantee that the obtained velocity at the new time step will fulfill the incompressibility constraint. As a consequence, we treat (3) in an implicit way. Supposing that the velocity and pressure at time step n $\left[\boldsymbol{u}_h^n, p_h^n\right]$ are known, we may solve (3) for example with $\partial_t \boldsymbol{u}_h$ being discretized using a backward differences in time scheme

$$\begin{aligned} \partial_t \boldsymbol{u}_h &\approx \delta_t \boldsymbol{u}_h^{n+1}, \\ \delta_t \boldsymbol{u}_h^{n+1} &:= \begin{cases} \frac{1}{\delta t}\left(\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n\right) & \text{First-order scheme} \\ \frac{1}{\delta t}\left(\frac{3}{2}\boldsymbol{u}_h^{n+1} - 2\boldsymbol{u}_h^n + \frac{1}{2}\boldsymbol{u}_h^{n-1}\right) & \text{Second-order scheme} \end{cases} \end{aligned} \tag{5}$$

where δt is the time step size. Usually, a second-order scheme is used for approximating the time derivative in the Galerkin term $B\left([\boldsymbol{v}_h, q_h], \left[\boldsymbol{u}_h^{n+1}, p_h^{n+1}\right]\right)$, whereas a first-order scheme is used for approximating the derivative in the residual in the stabilizing terms $\boldsymbol{r}\left(\left[\boldsymbol{u}_h^{n+1}, p_h^{n+1}\right]\right)$, because this term is multiplied by τ_K and this parameter is precisely of the order of the critical time step of explicit schemes [28]. Note that none of the terms of the fully discrete form of the problem involves the pressure at time step n, p_h^n.

3. PROPER ORTHOGONAL DECOMPOSITION-BASED REDUCED-ORDER MODELS

In this section, a general strategy for reducing a given variational problem will be described.

Suppose that after fully discretizing in time and space, given a variational problem, we end up with the following matrix form which allows us to obtain the vector of nodal unknowns $\boldsymbol{U}_{n+1}$ for a given time step $n + 1$

$$\boldsymbol{F}(\boldsymbol{U}_{n+1}, \boldsymbol{U}_n, \boldsymbol{U}_{n-1}, ..) := \boldsymbol{A}(\boldsymbol{U}_{n+1})\boldsymbol{U}_{n+1} - \boldsymbol{B}_n(\boldsymbol{U}_n)\boldsymbol{U}_n - \boldsymbol{B}_{n-1}(\boldsymbol{U}_{n-1})\boldsymbol{U}_{n-1} - \ldots - \boldsymbol{C} = \boldsymbol{0}. \tag{6}$$

If we denote by M the number of unknowns of the space-discrete problem, then $\boldsymbol{U}_{n+1}, \boldsymbol{U}_n, \ldots, \boldsymbol{F}, \boldsymbol{C} \in \mathbb{R}^M$. Note that matrices $\boldsymbol{A}, \boldsymbol{B}_n, \boldsymbol{B}_{n-1}, \ldots \in \mathbb{R}^{M \times M}$ might have a dependence on the vectors of unknowns $\boldsymbol{U}$, thus yielding a nonlinear problem. On the other hand, the discretization of the time derivative at time instant t_n usually involves values of the unknowns only at the first few previous time steps. We can group the constant terms and the terms corresponding to previous time steps into

$$\boldsymbol{R}(\boldsymbol{U}_n, \boldsymbol{U}_{n-1}, \ldots) := \boldsymbol{B}_n(\boldsymbol{U}_n)\boldsymbol{U}_n + \boldsymbol{B}_{n-1}(\boldsymbol{U}_{n-1})\boldsymbol{U}_{n-1} + \ldots + \boldsymbol{C}.$$

The previous nonlinear matrix form needs to be linearized. There are several possibilities for doing so, for example, Picard's or Newton's linearization methods. In the following, Picard's linearization is used because the radius of convergence of Picard's method is larger than the radius of convergence of Newton's method for the incompressible Navier–Stokes equations, despite the

DOI: 10.1002/fld

convergence rate of Newton's method being higher [29, 30]. If Picard's linearization is used, (6) is iteratively solved in the following manner

$$\boldsymbol{A}\left(\boldsymbol{U}_{n+1}^{i-1}\right)\boldsymbol{U}_{n+1}^{i}=\boldsymbol{R}(\boldsymbol{U}_n,\boldsymbol{U}_{n-1},\ldots), \tag{7}$$

where i denotes the current iteration. A similar system would arise for Newton's method of the form

$$\boldsymbol{J}\left(\boldsymbol{U}_{n+1}^{i-1}\right)\boldsymbol{U}_{n+1}^{i}=\boldsymbol{R}_N\left(\boldsymbol{U}_{n+1}^{i-1},\boldsymbol{U}_n,\boldsymbol{U}_{n-1},\ldots\right),$$

where $\boldsymbol{J}\in\mathbb{R}^{M\times M}$ is the Jacobian and $\boldsymbol{R}_N\in\mathbb{R}^{M}$ is the right-hand side vector for Newton's method. Note that, as in Picard's method, both $\boldsymbol{J}$ and $\boldsymbol{R}_N$ are nonlinear and depend on the values of the unknown at previous iterations or time steps.

In order to build a ROM, the previous full-order system can be projected onto a low-dimensional subspace $\mathcal{U}\subset\mathbb{R}^{M}$. Vectors $\boldsymbol{U}$ are now approximated by

$$\boldsymbol{U}\simeq\tilde{\boldsymbol{U}}=\boldsymbol{\Phi}_u\boldsymbol{U}_\Phi, \tag{8}$$

where $\boldsymbol{\Phi}_u\in\mathbb{R}^{M\times N}$ is the basis for $\mathcal{U}$ and N is the dimension of the ROM with $N<M$. $\boldsymbol{U}_\Phi\in\mathbb{R}^{N}$ are the components in $\mathcal{U}$ expressed in the reference system defined by $\boldsymbol{\Phi}_u$. The reduced-order basis $\boldsymbol{\Phi}_u$ is obtained by means of a proper orthogonal decomposition (POD) procedure [31–33]. The POD method consists in storing a sample of vectors $\boldsymbol{U}$ which are representatives of the solution of the problem along its time evolution. This sample is called the snapshot set and it must be chosen carefully in order to obtain an accurate ROM. The next step consists in performing a singular value decomposition of the snapshot set and keeping the first N resulting basis functions. These functions are optimal in the sense that the space spanned by the POD basis minimizes over all subspace of dimension N the Frobenius norm of the projection error of the snapshot matrix. Another relevant property is that the recovered basis functions are orthonormal. Also, the reduced-order basis functions fulfill the boundary conditions of the FOM, which ensures that any solution field which is a linear combination of the reduced-order basis also fulfills the boundary conditions.

After introducing the approximation (8) in (7), an overdetermined system with M equations and N unknowns is obtained. As explained in [19, 24], if matrix $\boldsymbol{A}$ is symmetric and positive definite, a least-squares strategy for approximating the linearized overdetermined system

$$\boldsymbol{A}\left(\boldsymbol{U}_{n+1}^{i-1}\right)\boldsymbol{\Phi}_u\boldsymbol{U}_{\Phi,n+1}^{i}=\boldsymbol{R}(\boldsymbol{U}_n,\boldsymbol{U}_{n-1},\ldots),$$

leads to

$$\boldsymbol{\Phi}_u^T\boldsymbol{A}\left(\boldsymbol{U}_{n+1}^{i-1}\right)\boldsymbol{\Phi}_u\boldsymbol{U}_{\Phi,n+1}^{i}=\boldsymbol{\Phi}_u^T\boldsymbol{R}(\boldsymbol{U}_n,\boldsymbol{U}_{n-1},\ldots), \tag{9}$$

which can be proven to minimize the error in the norm given by matrix $\boldsymbol{A}$

$$\boldsymbol{U}_{\Phi,n+1}^{i}=\arg\min_{\boldsymbol{U}_\Phi\in\mathcal{U}}\|\boldsymbol{\Phi}_u\boldsymbol{U}_\Phi-\boldsymbol{A}\left(\boldsymbol{U}_{n+1}^{i-1}\right)^{-1}\boldsymbol{R}\left(\boldsymbol{U}_n,\boldsymbol{U}_{n-1},\ldots\right)\|_{\boldsymbol{A}\left(\boldsymbol{U}_{n+1}^{i-1}\right)},$$

where $\|\cdot\|_{\boldsymbol{A}\left(\boldsymbol{U}_{n+1}^{i-1}\right)}$ denotes the norm with respect to matrix $\boldsymbol{A}$ at time step n, iteration $i-1$.

If $\boldsymbol{A}$ is not symmetric and positive definite, a Petrov–Galerkin projection is required in order to ensure the optimality of the recovered solution [24]. In the ROM to be developed in the next section, a symmetric and positive definite matrix is obtained, and the described Galerkin projection is used.

3.1. Hyper-reduced models

The system defined in Equation (9) is the ROM associated to the original system (6). However, although the system of Equation (9) is of dimension N (which typically corresponds to few degrees of freedom), matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and vector $\boldsymbol{C}$ still need to be built at each time step, and the cost of building these matrices is of order M, the dimension of the original system (6). If this straightforward approach is adopted, solving the reduce-order system (9) leads to saving some computational effort with respect to solving (6) (mainly because the linear system to be solved is of dimension $N\times N$ instead of $M\times M$). However, these are not the drastic computational savings expected from the ROM.

DOI: 10.1002/fld

For linear problems with constant coefficients, this is not an issue, because matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are constant in time (they do not depend on vectors $\boldsymbol{U}$). In this case, the products

$$\boldsymbol{\Phi}_u^T \boldsymbol{A} \boldsymbol{\Phi}_u,\ \boldsymbol{\Phi}_u^T \boldsymbol{B} \boldsymbol{\Phi}_u,\ \boldsymbol{\Phi}_u^T \boldsymbol{C},$$

can be pre-computed in the off-line stage (that is, prior to the ROM computations), whereas in the online stage, all the operations are associated to the reduced-order system. In the case of quadratically nonlinear terms (such as the convective term in the Navier–Stokes equations) some strategies have been developed [24, 34, 35] which avoid the full-order computation of the nonlinear matrices.

However, these approaches are not suitable for problems in which nonlinear matrices have a more general dependency on the unknowns (which is the case of the stabilization terms of the formulation presented in Section 2 through the stabilization parameter τ_K) or the parameters of the simulation (such as parameters controlling the shape of the computational domain in shape optimization problems).

Recently, a more general approach for dealing with the nonlinear terms of ROMs has appeared [18, 20, 22, 23], giving rise to what are known as *hyper-reduced*-order models. In these methods, the nonlinear and parameter-dependent terms are recovered by means of a least-squares procedure from a series of sampling components where the function to be approximated is computed. When applied to finite element analysis, this translates in a method in which the function to be approximated is computed at certain points of the finite element mesh and then extrapolated to the remaining points by means of the least-square extrapolation.

As it will be explained in the next section, in the explicit formulation for the Navier–Stokes equations we propose, we follow an approach similar to the one presented in [19], in which the functions to be approximated are the matrix (times the reduced-order basis) and the right-hand side of the linearized system of equations. The method for approximating the right-hand side of the vector is the gappy-pod presented in [36]. The main advantage of the strategy we propose is that, because all the terms of the formulation are treated in an explicit way, only vectors need to be recovered, instead of matrix times reduced-order basis arrays. This results in a much cheaper ROM both in terms of memory and computational cost per time step.

Let us consider a vector of nodal values at nodal points of a finite element mesh, $\boldsymbol{V} \equiv [V_1, V_2, \ldots, V_M] \in \mathbb{R}^M$. Let us also consider a reduced-order basis for $\boldsymbol{V}$ obtained by means of a POD of a set of snapshots for $\boldsymbol{V}$, so that $\boldsymbol{V}$ can be approximated as

$$\boldsymbol{V} \simeq \boldsymbol{\Phi}_v \boldsymbol{V}_\Phi,$$

where $\boldsymbol{\Phi}_v \in \mathbb{R}^{M \times N}$ is the reduced-order basis which defines the low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^M$ and N is the dimension of the ROM. $\boldsymbol{V}_\Phi \in \mathbb{R}^N$ are the components of $\boldsymbol{V}$ expressed in the reference system defined by $\boldsymbol{\Phi}_v$. Let us also consider that we only know the nodal values for $\boldsymbol{V}$ at some sampling components $V_{i(k)}$, $1 \leqslant k \leqslant n_s$, where n_s is the number of sampling components of the vector, $i(k)$ denotes the kth sampling component. We now want to recover the reduced-order basis coefficients $\boldsymbol{V}_\Phi$ of the reduced-order basis $\boldsymbol{\Phi}_v$ for vector $\boldsymbol{V}$. This is in fact the situation we will face each time we want to solve a time step for the hyper-ROM, where we will only sample the right-hand side for certain sampling nodes of the finite element mesh (and the degrees of freedom associated to these nodes).

In order to recover $\boldsymbol{V}_\Phi$, we can solve the least-squares minimization problem

$$\boldsymbol{V}_\Phi = \arg \min_{\boldsymbol{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^{N} \left(\Phi_{v,i(k)j} a_j - V_{i(k)} \right)^2, \tag{10}$$

where $\boldsymbol{\Phi}_{v,i(k)j}$ denotes the basis vector j evaluated at the kth sampling component, $i(k)$.

The previous strategy provides the tools required to extrapolate the right-hand side vector and the matrix of the linear system of equations arising from the finite element problem. The main advantage is that in order to do so, only the nodal values at certain (few) sampling components are needed. In the case, we need to extrapolate the matrix, a cheaper (in terms of computational cost) approach is to approximate the matrix times the reduced-order basis vectors

$$\boldsymbol{A}(\boldsymbol{U}_{n+1}) \boldsymbol{\Phi}_u, \tag{11}$$

DOI: 10.1002/fld

This approach has been used in [19] in order to approximate the Jacobian matrix in Newton's method times the reduced-order basis functions; $\boldsymbol{J}(\boldsymbol{U}_{n+1})\boldsymbol{\Phi}_u$. Note that this strategy involves reconstructing a matrix of dimension $M \times N$. Even if this approach is taken, approximating (11) can be computationally expensive in terms of memory, because a reduced-order basis for each of the vectors needs to be stored. Another possibility is to use the reduced-order basis for the right-hand side vector in order to represent the system matrix times the reduced-order basis functions (11), (see [19, 21]), but this turns out to provide inaccurate results when applied to the stabilized finite element approximation of the incompressible Navier–Stokes equations (3): we have checked experimentally in several examples that the reconstructed matrix is not accurate enough and leads to unstable results. This is the reason why in Section 4, we will focus in finding an explicit formulation for the ROM; in this way, matrix $\boldsymbol{A}(\boldsymbol{U}_{n+1})$ will not only be linear (without any dependence on $\boldsymbol{U}_{n+1}$) but also diagonal. In this case, the extrapolation procedure can be applied to matrix $\boldsymbol{A}$ exactly as in the right-hand side vector case, or the diagonal of $\boldsymbol{A}$ can be computed once at the beginning of the reduced-order simulation.

3.2. A discrete version of the best points interpolation method

When using HROMs and the strategy described in the previous section, the quality of the recovered right-hand side vector highly depends on the selected sampling components. Several strategies have been developed for choosing these sampling components. In the missing point estimator method [37], the sampling components are chosen such that the hyper-reduced basis functions continue to be close to orthonormal. Another possibility is the strategy presented in [21] where nonlinear differential operators and their Fréchet derivatives are approximated by means of an Empirical Interpolation Method (EIM) [18]. In [20], the discrete EIM (DEIM) is presented, where the sampling components are selected iteratively by imposing that the error growth at each iteration is limited. Some of the advantages of the DEIM method are that it relies only on the reduced-order basis in order to choose the sampling components, and its *discrete* nature, which means that the reduced basis are treated as array vectors instead of being treated as functions defined in the physical domain. However, there is no guarantee that the error of the recovered right-hand side is minimal in any norm.

Another possibility is the best points interpolation method (BPIM) approach presented in [23], where the sampling points are chosen so that the distance between the projection of the *right-hand side snapshots* onto the reduced basis subspace and the recovered right-hand side is minimized.

In the BPIM, sampling points do not necessarily coincide with nodal points of the finite element mesh. This means that the computational cost of the sampling is larger in the best points interpolation method (all the nodal points belonging to the element in which the sampling coordinates lie need to be fully assembled). But on the other hand, it yields more accurate results than the *empirical interpolation* family of methods because it guarantees that the distance between the recovered right-hand sides and the optimal right-hand side is minimized in a certain norm. The BPIM performs very well in the case of smooth right-hand side vectors or smooth functions because the optimal sampling coordinates are very easily found by using the Marquardt method [38]. However, in the case of right-hand side vectors arising from the finite element analysis of the incompressible Navier–Stokes equations, this is not the case (see Figure 1 where a typical right-hand side vector

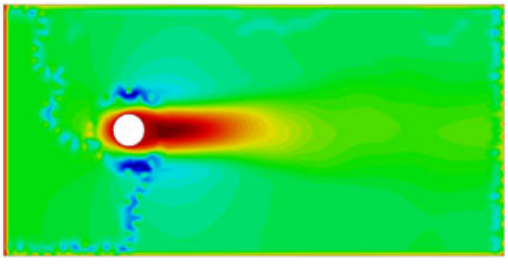

Figure 1. Right-hand side velocity components snapshot field, see Section 5.1.

DOI: 10.1002/fld

distribution over nodal points is depicted). This is the reason why we propose the *Discrete version of the Best Point Interpolation Method (DBPIM)* which is described next.

Similarly to the BPIM, the method consists of minimizing the error between the recovered right-hand side vector snapshots and the actual snapshots. However, in the strategy we use, we force the sampling coordinates to coincide with nodal points of the finite element mesh. Plus, once a component associated to a node of the finite element mesh is selected, all the degrees of freedom associated to that node are included in the sampling selection. Moreover, because of the lack of smoothness of the vectors which are being approximated (Figure 1), we do not use a Marquardt related strategy in order to advance to the optimal set of sampling nodes. Instead, we use an algorithm which advances from one set of sampling nodes to the next one by evaluating the error of the recovered snapshots at the neighbor points in the finite element mesh and replaces a sampling node with its neighbor if the error diminishes. The DBPIM algorithm is detailed in Algorithm 1 for a scalar unknown (where each sampling node is associated to a single sampling component). If we have a vector unknown, we proceed in the same way and we impose that, for each sampling node, all the degrees of freedom associated to it are sampling components.

Algorithm 1 Discrete Best Points Interpolation Method for a scalar problem

$\boldsymbol{\Phi}_v \boldsymbol{V}_{\Phi}^{\alpha} = \Pi_{\mathcal{V}}(\boldsymbol{V}^{\alpha}), \alpha = 1, N_{\text{snapshots}}$
Choose an initial set of sampling components $\boldsymbol{i} \in \mathbb{N}^{n_s} \mid 1 \leqslant i(k) \leqslant M,\ k = 1, \ldots n_s$, (for example by using the DEIM method).
Solve: $\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}(\boldsymbol{i}) = \arg\min_{\boldsymbol{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^{N} (\Phi_{v,i(k)j} a_j - V_{i(k)}^{\alpha})^2, \alpha = 1, N_{\text{snapshots}}$
$e(\boldsymbol{i}) = \sum_{j=1}^{N_{\text{snapshots}}} ||\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}(\boldsymbol{i}) - \boldsymbol{V}_{\Phi}^{\alpha}||^2$
while set of sampling points has changed **do**
  for $m = 1 : n_s$ **do**
    for $l = 1 : N_{\text{neigh}}(i(m))$ **do**
      if the lth neighbour of $i(m)$ has not been previously tested **then**
        Temporarily replace sampling node $i(m)$ by its lth neighbour
        Solve: $\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}(\boldsymbol{i}) = \arg\min_{\boldsymbol{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^{N} \left(\Phi_{v,i(k)j} a_j - V_{i(k)}^{\alpha}\right)^2, \alpha = 1, N_{\text{snapshots}}$
        $e_{\text{temp}}(\boldsymbol{i}) = \sum_{\alpha=1}^{N_{\text{snapshots}}} ||\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}(\boldsymbol{i}) - \boldsymbol{V}_{\Phi}^{\alpha}||$
        if $e_{\text{temp}} < e$ **then**
          $e = e_{\text{temp}}$
          Permanently replace sampling node $i(m)$ by its lth neighbour
          Restart l loop
        end if
      end if
    end for
  end for
end while

The first step of the DBPIM algorithm consists of finding the projection $\Pi_{\mathcal{V}}$ of the snapshots onto the reduced-order subspace defined by the reduced-order basis, $\mathcal{V}$. For each snapshot, this yields the coefficients $\boldsymbol{V}_{\Phi}^{\alpha}$. In the second step, we choose an initial set of sampling nodes, which can be carried out by using the DEIM method. If the DEIM method is used, it will give us a set of sampling components. For a scalar problem, each component corresponds to a node of the finite element mesh. If the unknown is a vector field, then the nodes associated to the DEIM sampling components are selected as initial sampling nodes, and the number of sampling nodes is equal to the number of reduced basis functions. Otherwise, we always choose the number of sampling nodes to be equal to the number of basis functions times an (usually low) integer. After defining the initial set of sampling nodes, the degree(s) of freedom associated to these sampling nodes become sampling components. For this initial set of sampling nodes, we recover the approximated coefficients $\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}$ by means of the

DOI: 10.1002/fld

previously described least-squares strategy. The error associated to a set of sampling components $\boldsymbol{i} \in \mathbb{N}^{n_s}$, whose kth component is indicated as $i(k) \in 1, \ldots, M$, is obtained by computing the difference between the exact and the approximated $\boldsymbol{V}_{\Phi}^{\alpha}$ coefficients

$$e(\boldsymbol{i}) = \sum_{\alpha=1}^{N\text{snapshots}} ||\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}(\boldsymbol{i}) - \boldsymbol{V}_{\Phi}^{\alpha}|| \tag{12}$$

where

$$\boldsymbol{V}_{\Phi}^{\alpha,\text{aprox}}(\boldsymbol{i}) = \arg \min_{\boldsymbol{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^{N} \left(\Phi_{v,i(k)j} a_j - V_{i(k)}^{\alpha} \right)^2, \alpha = 1, N_{\text{snapshots}} \tag{13}$$

For this definition of the error, we can define the optimal set of sampling components as

$$\boldsymbol{b} = \arg \min_{\boldsymbol{i} \in \mathbb{N}^{n_s} \mid 1 \leqslant i(k) \leqslant M,\ k=1,\ldots n_s} e(\boldsymbol{i}) \tag{14}$$

In order to obtain the set of sampling points to be used in the reduced-order simulation, we proceed as follows. For each sampling node of the finite element mesh, we loop over its neighbors in the computational mesh and we temporarily replace the sampling node by each of them. If the error of the new set is lower than the original error, the sampling node is permanently replaced by its neighbor. This procedure is repeated while the set of sampling points changes due to the algorithm (**while** loop in Algorithm 1). Although this **while** loop is required in order to ensure the local optimality of the final set of sampling points, the algorithm can be executed just once (as it has been carried out in the numerical examples of this paper) and a sufficiently good set of sampling points for the ROM is obtained.

This strategy ensures that the error associated to the sampling node set decreases monotonically, departing from the error of the initial set of sampling nodes. However, the algorithm can only guarantee that a local optimum is obtained. Arriving to the global optimum defined in (14) depends on the initial set of sampling nodes. Despite this, when applied to the numerical examples in Section 5, it results in a good performance of the HROMs. Note also that contrary to the DEIM method, where the number of sampling nodes need to coincide with the number of reduced-order basis, the number of sampling nodes is arbitrary in the DBPIM. The DBPIM can take several steps (understood as replacements of each of the sampling nodes by its neighbors) to arrive to the (locally) optimal set of sampling nodes, but the cost of each of the steps is of $\mathcal{O}(N_{\text{snapshots}} \times n_s)$ which is typically low compared with the dimensions of the original system. Moreover, these operations can be performed at the off-line stage after the POD procedure is carried out.

If edge or face unknowns are present in the finite element formulation of the problem, edges or faces of the finite element problem should be treated as a *nodes* in the DBPM algorithm. A sampling set of nodes and edges or faces would arise from Algorithm 1 should edge or face unknowns exist.

4. AN EXPLICIT FORMULATION FOR THE REDUCED-ORDER MODEL

As explained in Section 3.1, it is convenient to work with an as explicit as possible formulation in order to obtain an efficient and accurate ROM. This means that we need to send as many terms as possible to the right-hand side in (9) (matrices $\boldsymbol{B}$ and $\boldsymbol{C}$). However, in Section 2.2, we have stressed the need of an implicit time integration scheme for the stabilized incompressible Navier–Stokes equations. In this section, we will see how to address the problem of devising an explicit time integration scheme for the reduced-order finite element approximation of the stabilized incompressible Navier–Stokes equations.

4.1. Explicit formulation for a pure Galerkin method

In the case a pure Galerkin formulation is used, devising an explicit formulation is quite straightforward: the reduced-order basis is divergence free because of the fact that all the snapshots fulfill the

DOI: 10.1002/fld

weak form of the continuity equation

$$(\hat{q}, \nabla \cdot \hat{\boldsymbol{u}}) = 0, \quad \forall \hat{q} \in \hat{Q},$$

where $\hat{Q} \subset Q_h$ is the pressure subspace defined by the pressure part of the POD basis functions and for each time t, $\hat{\boldsymbol{u}}(t) \in \hat{V} \subset V_h$ is the weakly divergence free velocity subspace defined by the velocity part of the POD basis functions. As a consequence, the terms $(\hat{p}, \nabla \cdot \hat{\boldsymbol{v}})$ and $(\hat{q}, \nabla \cdot \hat{\boldsymbol{u}})$ can be eliminated from the original bilinear form, yielding the following pressure-free weak form of the problem

$$(\hat{\boldsymbol{v}}, \partial_t \hat{\boldsymbol{u}}) + (\hat{\boldsymbol{v}}, \hat{\boldsymbol{u}} \cdot \nabla \hat{\boldsymbol{u}}) + \nu(\nabla \hat{\boldsymbol{v}}, \nabla \hat{\boldsymbol{u}}) = \langle \hat{\boldsymbol{v}}, \boldsymbol{f} \rangle, \quad \forall \hat{\boldsymbol{v}} \in \hat{V}. \tag{15}$$

4.2. Difficulties of pressure elimination in a variational multiscale stabilized formulation

When we turn to the stabilized form of the problem (3), getting rid of the pressure and developing an explicit time integration scheme for the ROM requires some additional steps. The first reason for this is that the reduced-order basis is not exactly divergence free but fulfills instead the linear stabilized continuity equation

$$(\hat{q}, \nabla \cdot \hat{\boldsymbol{v}}) + \sum_K \tau_K (\nabla \hat{q}, \hat{\boldsymbol{r}})_K = 0, \quad \forall \hat{q} \in \hat{Q}, \tag{16}$$

for any $\hat{\boldsymbol{v}} \in \hat{V}$ belonging to the reduced-order basis.

Despite of this, (16) can be considered a good approximation to the incompressibility constraint (the residual $\hat{\boldsymbol{r}}$ will be small if the finite element solution is a good approximation to the exact solution), that is, it would be reasonable to assume the approximation

$$(\hat{q}, \nabla \cdot \hat{\boldsymbol{v}}) \approx 0. \quad \forall \hat{q} \in \hat{Q} \tag{17}$$

for any $\hat{\boldsymbol{v}} \in \hat{V}$ belonging to the reduced-order basis. However, it is still not possible to eliminate the pressure from the momentum equation because introducing (17) in the momentum equation yields

$$(\hat{\boldsymbol{v}}, \partial_t \hat{\boldsymbol{u}}) + (\hat{\boldsymbol{v}}, \hat{\boldsymbol{u}} \cdot \nabla \hat{\boldsymbol{u}}) + \nu(\nabla \hat{\boldsymbol{v}}, \nabla \hat{\boldsymbol{u}}) + \sum_K \tau_K (\hat{\boldsymbol{u}} \cdot \nabla \hat{\boldsymbol{v}} + \nu \Delta \hat{\boldsymbol{v}}, \boldsymbol{r}([\hat{\boldsymbol{u}}, \hat{p}]))_K = \langle \hat{\boldsymbol{v}}, \boldsymbol{f} \rangle. \tag{18}$$

The previous expression involves the pressure dependent term $\tau_K(\hat{\boldsymbol{u}} \cdot \nabla \hat{\boldsymbol{v}} + \nu \Delta \hat{\boldsymbol{v}}, \nabla \hat{p})$. One can choose to simply drop this term from the reduced equations, but this introduces a consistency error in the formulation of the ROM which strongly manifests in the practical cases as an over-diffusive behavior of the ROM.

4.3. Proposed strategy

Instead, we will present a strategy in which all terms involving the pressure are treated in an explicit way. This is possible in the ROM because

- All reduced basis functions do already fulfill the stabilized continuity equation (16). Because the constraint introduced by the continuity equation is linear, any function in the span of this basis will also satisfy this equation.
- If basis functions are taken to be joint velocity-pressure basis functions (that is $\boldsymbol{\Phi}_u$ contains the coefficients of functions in $\hat{V} \times \hat{Q}$), then the pressure at time step $n+1$ is automatically recovered from coefficients $\boldsymbol{U}_{\Phi,n+1}$ and the reduced-order basis $\boldsymbol{\Phi}_u$ even if all the terms involving the pressure are treated in an explicit way in the reduced order formulation.

The second item in the previous list involves that, after each time step in the ROM, the velocity and pressure values need to be recovered from the reduced basis functions and the obtained reduced basis coefficients. In fact, for the HROMs, only the values at the sampling components are required, which allows to keep the number of operations in this step of $\mathcal{O}(n_s)$.

DOI: 10.1002/fld

The variational formulation for the first-order ROM that we propose is

$$\left(\hat{\boldsymbol{v}}, \delta_t \hat{\boldsymbol{u}}^{n+1}\right) + \left(\hat{\boldsymbol{v}}, \hat{\boldsymbol{u}}^{n*} \cdot \nabla \hat{\boldsymbol{u}}^{n*}\right) + \nu(\nabla \hat{\boldsymbol{v}}, \nabla \hat{\boldsymbol{u}}^{n*}) - (\hat{p}^{n*}, \nabla \cdot \hat{\boldsymbol{v}}) + \sum_K \tau_K \left(\hat{\boldsymbol{u}}^n \cdot \nabla \hat{\boldsymbol{v}} + \nu \Delta \hat{\boldsymbol{v}}, \delta_t \hat{\boldsymbol{u}}^n - \nu \Delta \hat{\boldsymbol{u}}^n + \hat{\boldsymbol{u}}^n \cdot \nabla \hat{\boldsymbol{u}}^n + \nabla \hat{p}^n - \boldsymbol{f}^n\right)_K = \langle \hat{\boldsymbol{v}}, \boldsymbol{f}^n \rangle . \tag{19}$$

where the terms $\hat{\boldsymbol{u}}^{n*}$ and $\hat{p}^{n*}$ are a second-order approximation of the state at $n+1$, the velocity and the pressure at $n+1$ given by

$$\begin{aligned} \hat{\boldsymbol{u}}^{n*} &= 2\hat{\boldsymbol{u}}^n - \hat{\boldsymbol{u}}^{n-1}, \\ \hat{p}^{n*} &= 2\hat{p}^n - \hat{p}^{n-1}. \end{aligned} \tag{20}$$

In the case of the second-order ROM, we use the same variational formulation (19), but the terms $\hat{\boldsymbol{u}}^{n*}$ and $\hat{p}^{n*}$ are now a third-order approximation of the state at $n+1$ given by

$$\begin{aligned} \hat{\boldsymbol{u}}^{n*} &= \frac{12}{5}\hat{\boldsymbol{u}}^n - \frac{9}{5}\hat{\boldsymbol{u}}^{n-1} + \frac{2}{5}\hat{\boldsymbol{u}}^{n-2}, \\ \hat{p}^{n*} &= \frac{12}{5}\hat{p}^n - \frac{9}{5}\hat{p}^{n-1} + \frac{2}{5}\hat{p}^{n-2}. \end{aligned} \tag{21}$$

Note that for the first-order explicit scheme, we propose to use the second-order extrapolation (20) and for the second-order scheme, the third-order extrapolation (21). The following remarks are in order

- In the case of the first-order ROM, we have observed that extrapolation (20) is in fact the only essential for the convective term in the Galerkin contribution. The viscous and pressure terms can be evaluated with the unknowns at time step n.
- As it is written, (19) is formally second order in time (except if the viscous and pressure terms are evaluated at n), in spite of the fact that we use it when the FOM is computed with a first-order scheme. We have observed empirically that a straightforward first-order evaluation of the convective term in the Galerkin contribution leads to an unstable scheme.
- As it has been explained earlier, the stabilization parameter τ_K is $\tau_K = \mathcal{O}(\delta t)$ and therefore the stabilization terms in (19) are formally $\mathcal{O}(\delta t^2)$.
- In the case of the second-order ROM, the Galerkin contribution is formally $\mathcal{O}(\delta t^3)$ due to the third-order extrapolation (21) even if we use it when the FOM is computed with a second-order scheme. Similarly to the previous case, we have observed empirically that a second-order evaluation is not enough, and the resulting scheme is unstable.

In the numerical examples of Section 5, $\hat{\boldsymbol{u}}^{n*}$ is also used in the FOM in order to approximate the convective velocity and avoid the need of performing several Picard iterations for the convective term.

The time derivatives appearing in (19) are treated in two different ways

- The time derivative in the residual in the stabilization terms is approximated by using the first-order backward differences scheme in (5). Note that this term is computed using the velocities $\hat{\boldsymbol{u}}^n$ and $\hat{\boldsymbol{u}}^{n-1}$ and as a consequence belongs to the right-hand side of the system of equations.
- In order to deal with the Galerkin term time derivative, the same schemes appearing in the time derivative approximation in the implicit scheme (5) are used, depending on whether the FOM uses a first-order or second-order time integration scheme.

The important fact about the formulation in (19) is that the corresponding matrix form of the problem can be written as

$$\boldsymbol{F}(\boldsymbol{U}_{n+1}, \boldsymbol{U}_n, \boldsymbol{U}_{n-1}, ..) = \boldsymbol{A}\boldsymbol{U}_{n+1} - \boldsymbol{B}_n(\boldsymbol{U}_n)\boldsymbol{U}_n - \boldsymbol{B}_{n-1}(\boldsymbol{U}_{n-1})\boldsymbol{U}_{n-1} - \ldots - \boldsymbol{C}, \tag{22}$$

that is, matrix $\boldsymbol{A}$ is linear. $\boldsymbol{A}$ corresponds in fact to the mass matrix. Terms corresponding to $\boldsymbol{B}_n(\boldsymbol{U}_n)\boldsymbol{U}_n$ are the ones evaluated at time step n in (19), and $\boldsymbol{B}_{n-1}(\boldsymbol{U}_{n-1})\boldsymbol{U}_{n-1}$ corresponds to

DOI: 10.1002/fld

the time derivative terms evaluated at time step $n-1$ if a second-order backward differences scheme is used for the time integration. $\boldsymbol{C}$ corresponds to the forcing term. The approach we follow is to lump matrix $\boldsymbol{A}$ into a diagonal matrix. This is very advantageous because the extrapolation described in Section 3.1 needs to be applied only to vectors, instead of vectors and matrices times the reduced-order basis, even in the case in which matrix $\boldsymbol{A}$ is not constant because of a geometric dependence on the simulation parameters. Once (22) has been solved, the full-order velocity and pressure at $n+1$ can be recovered by multiplying the reduced-order basis $\boldsymbol{\Phi}_u$ by the obtained reduced-order components $\boldsymbol{U}_{\Phi,n+1}$. In the case matrix $\boldsymbol{A}$ is not diagonal, a defect correction method could be used using a diagonal matrix as preconditioner. The proposed ROM strategy is summarized in Algorithm 2, which allows the reduced model to step forward in time for the total number of time steps (n_{steps}).

Algorithm 2 Explicit Reduced-Order Model strategy

1: Simulate the Full Order Model (FOM) and take snapshots for the:
   - Velocity and pressure fields
   - Right-hand side (RHS) vector of the explicit HROM *
2: //REDUCED-ORDER MODEL (ROM)
3: Build the diagonal ROM mass matrix
4: **for** itemp $= 1{:}n_{\text{steps}}$ **do**
5: Build the FOM RHS and the ROM RHS
6: Solve the ROM momentum equation
7: Recover the pressure field from the joint velocity-pressure snapshots
8: Take snapshots for the RHS vector of the explicit HROM *
9: **end for**
10: //HYPER-REDUCED MODEL (HROM)
11: Build or extrapolate the HROM diagonal mass matrix
12: **for** itemp $= 1{:}n_{\text{steps}}$ **do**
13: Extrapolate the FOM RHS, build the HROM RHS
14: Solve the HROM momentum equation
15: Recover the pressure field from the joint velocity-pressure snapshots
16: **end for**
17: *: Snapshots for the HROM RHS can be taken either in the FOM or the ROM.

As the numerical examples in Section 5 illustrate, the proposed strategy results in a computational cost drop of up to 65% for the ROM and more than 99% for the hyper-reduced models. The cause for this is that if the number of operations for the assembly and the system of equations solution in the FOM is $\mathcal{O}(M)$ (supposing an optimal multilevel linear system solver is used), then the cost of the assembly in the ROM is also $\mathcal{O}(M)$ but the cost of solving the system is $\mathcal{O}(N)$. Finally, the hyper-reduction is achieved by diminishing the cost of the assembly to $\mathcal{O}(n_s) \sim \mathcal{O}(N)$. It is also important to note that the time step for the ROMs is not restricted by the Courant–Friedrichs–Levy condition, which generally restricts the time step size of explicit schemes. This can be explained by the fact that basis functions for the ROM are *global*, in the sense that they cover the whole computational domain, in contrast to being localized in a patch of finite elements (as in the FOM).

5. NUMERICAL EXAMPLES

In this section, we present some numerical examples which illustrate the behavior of the proposed numerical methods. In these examples, we show how the ROMs are capable of reproducing the results of the FOMs. No parameter variation is considered, that is, the snapshots are obtained from the same problem on which the reduced model is run. The numerical examples were run in an Intel Core 2 Duo (E6750, 266 GHz) machine with 2 Gb of RAM. All the computations are serial and the described algorithms were coded in our in-house code FEMUSS.

DOI: 10.1002/fld

5.1. *Two-dimensional low Reynolds flow past a cylinder*

5.1.1. Reduced-order models for $\mathrm{Re} = 100$. In this numerical example, we study the incompressible flow around a cylinder at $\mathrm{Re} = 100$. The computational domain consists of a 16×8 rectangle with a unit-diameter cylinder centered at (4 ,4). The horizontal inflow velocity is set to 100 at $x = 0$. Slip boundary conditions that allow the flow to move in the direction parallel to the walls are set at $y = 0$ and $y = 8$, and velocity is set to 0 at the cylinder surface. The viscosity has been set to $\nu = 1$, which yields a Reynolds number $\mathrm{Re} = 100$ based on the diameter of the cylinder and the inflow velocity. A backward Euler scheme has been used for the time integration with time step $\delta t = 0.001$. In this example, a coarse 7294 linear element mesh has been used to solve the problem.

Regarding the ROM, 50 velocity-pressure snapshots have been taken and the 10 first reduced basis functions have been kept for the reduced model. Here and in the following numerical examples, the snapshots collection procedure consists of taking equally spaced snapshots during the whole simulation period of the FOM, which in this case spans over six periods of the vortex shedding. The number of basis functions that are kept is chosen so that a sufficiently accurate approximation is obtained in the ROM. For the hyper-reduced model, 50 additional snapshots from the right-hand side have been taken and the corresponding 10 first reduced basis functions have been kept. The total number of sampling nodes is 30 (three per basis function). As previously explained, the Courant–Friedrichs–Levy condition can be violated in the proposed reduced-order approach. In this numerical example, the Courant–Friedrichs–Levy (CFL) number associated to the finite element mesh was $CFL \sim 3$.

Figure 2 compares the velocity and pressure fields after 100 time steps for the FOM, the ROM and the hyper-reduced-order model (HROM). No difference can be appreciated between the results of the FOM and the ROMs.

Figure 3 compares the pressure and velocity in the vertical direction at a control point in the wake behind the cylinder at coordinates (6.5, 4). Barely any difference can be appreciated between the full-order results and the solution of the ROMs.

Figure 4 compares pressure and vertical velocity spectra at the same control point. Again, ROMs almost exactly match results from the FOM, with a slight difference in the spectra decay velocity, which is faster for the ROMs.

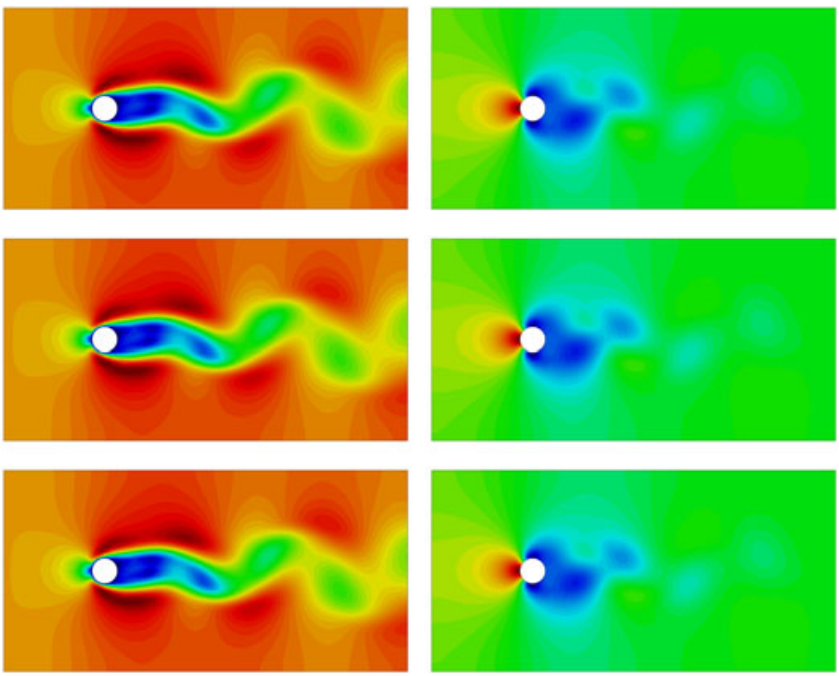

Figure 2. Velocity (left) and pressure (right) contours at $\mathrm{Re} = 100$ after 100 time steps. From top to bottom: full order model, reduced-order model, and hyper-reduced order model.

DOI: 10.1002/fld

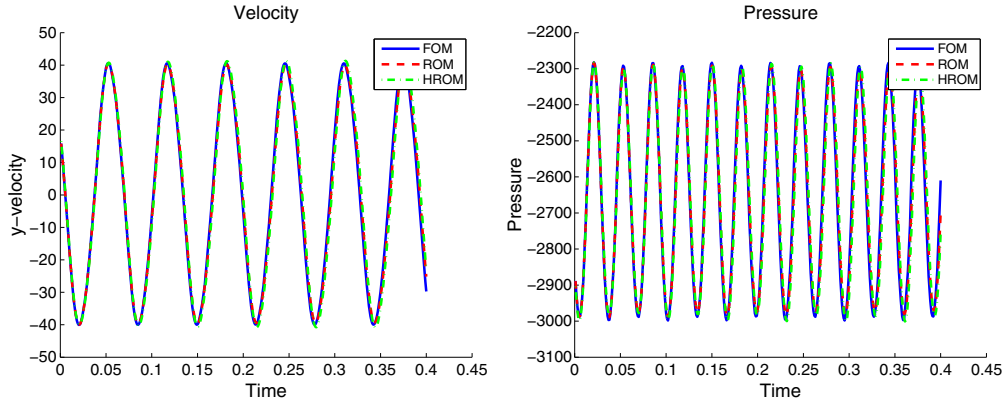

Figure 3. Velocity (left) and pressure (right) time history at a control point at the wake of the cylinder, $\mathrm{Re} = 100$.

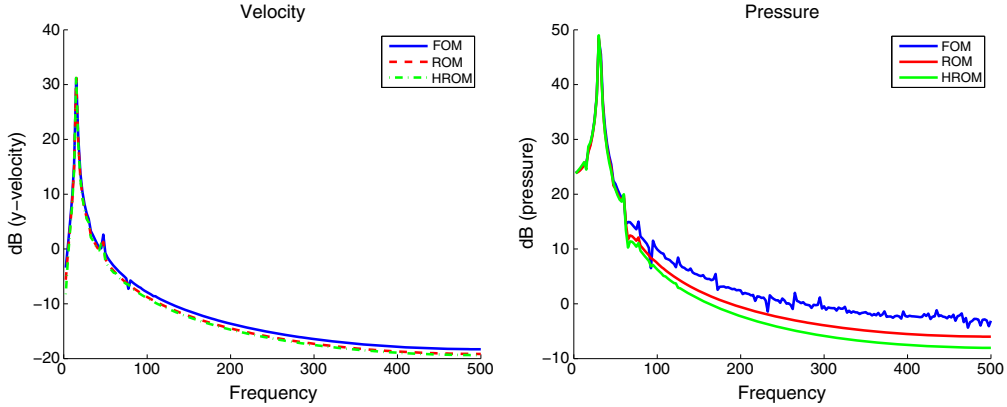

Figure 4. Velocity (left) and pressure (right) spectra at a control point at the wake of the cylinder, $\mathrm{Re} = 100$.

Regarding the computational cost, the FOM takes 65.9 s to run, the ROM, which builds the full-order system matrix but does not need to solve the associated linear systems of equations that takes 22.8 s (34.6% of the original computational cost). Finally, the hyper-reduced model, in which the computational cost depends only on the size of the ROM, takes only 0.45 s (0.7%) to run.

5.1.2. Reduced-order models for $\mathrm{Re} = 2000$. Here, we solve the same numerical example as in the previous section, but we set the viscosity to $\nu = 0.05$, which corresponds to a Reynolds number of $\mathrm{Re} = 2000$. The time step and the boundary conditions are the same as in the previous example. Regarding the ROMs, we take again 50 snapshots for both the velocity-pressure pair and right-hand side of the system and again 10 basis functions are kept. For the hyper-reduction, 30 sampling nodes are considered.

Figure 5 compares the pressure and velocity in the vertical direction at the control point in the wake behind the cylinder at coordinates (6.5, 4). Again, very few differences can be appreciated between the reduced order and the FOM: the velocity of the ROMs matches almost exactly the one from the FOM. In the case of the pressure values, the pressure for the FOM shows a slightly larger amplitude, although the frequency is exactly captured by the ROMs.

Figure 6 compares pressure and vertical velocity spectra at the same control point. Again, we can see that a very good agreement is obtained between the full order and the ROMs. However, the third harmonic frequency (~ 100) is not correctly captured by the ROMs, which show a much smaller amplitude. This can be improved by adding basis functions to the ROM. Figure 7 shows the spectra

DOI: 10.1002/fld

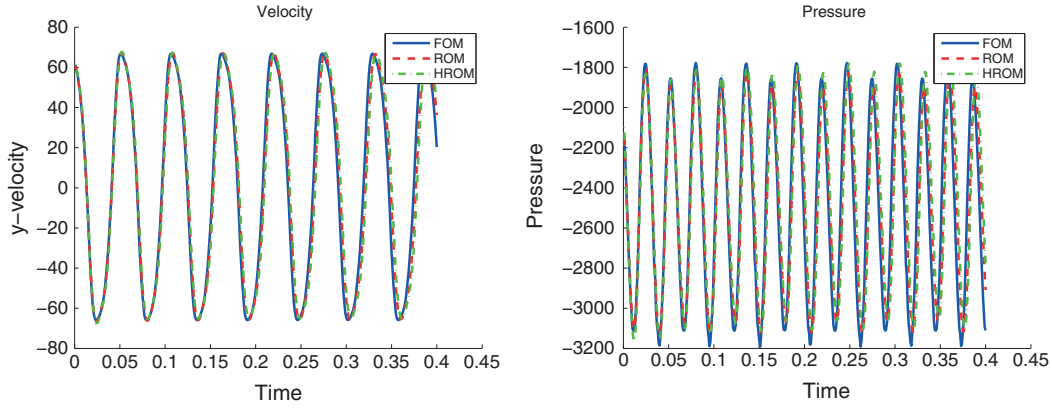

Figure 5. Velocity (left) and pressure (right) time history at a control point at the wake of the cylinder, Re = 2000.

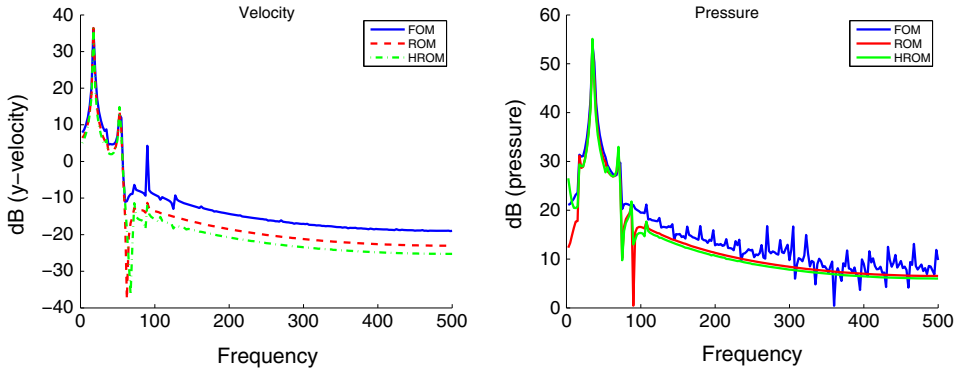

Figure 6. Velocity (left) and pressure (right) spectra at a control point at the wake of the cylinder, Re = 2000, 10 basis functions reduced-order model.

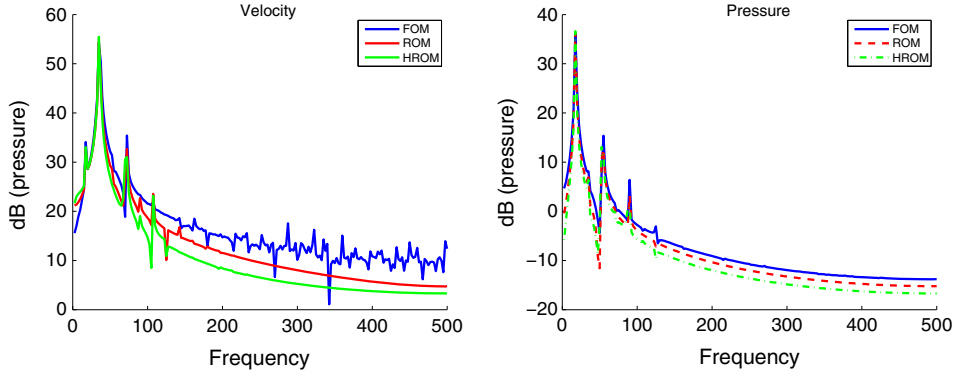

Figure 7. Velocity (left) and pressure (right) spectra at a control point at the wake of the cylinder, Re = 2000, 15 basis functions reduced-order model.

when 15 basis functions are used both for the velocity-pressure basis and for the right-hand side basis of the HROM.

However, we must note that adding an excessive number of additional basis functions to the velocity-pressure basis can result in an unstable behavior of the ROM, because of the fact that less representative basis functions are non-smooth and show local point-to-point oscillations which can cause instabilities. Regarding this issue, Figure 8 shows a *smooth* basis function (which corresponds

DOI: 10.1002/fld

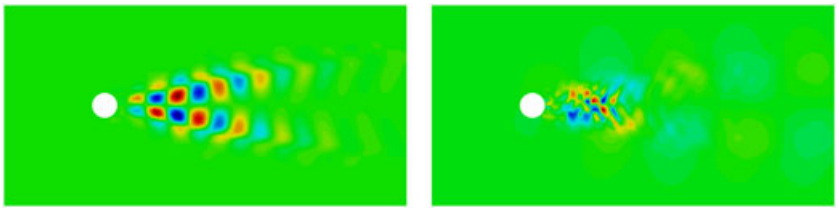

Figure 8. Smooth (left) versus noisy (right) basis functions.

to the seventh basis function obtained from the velocity–pressure snapshots) and a *noisy* basis function (which corresponds to the nineteenth basis function obtained from the snapshots). Snapshot 19 does not offer any additional information to the ROM and can cause an unstable behavior. The behavior of the ROM with respect to noisy basis functions is improved by

- Reducing the time step size.
- Refining the finite element mesh.

It is also interesting to note that the turbulent behavior that starts to manifest at the higher frequencies of the pressure spectra in the FOM is completely smoothed by the ROMs, which are only capable of capturing the lowest frequencies.

With respect to the computational cost, the FOM takes 65.2 s to run, the reduced order takes around 23 s (35%) both in 10 and 15 degrees of freedom ROMs and the hyper-reduced model takes 0.43 s (0.65%) to run in the simulation that uses 10 degrees of freedom model and 0.62 s (0.95%) in the one which uses 15 degrees of freedom model.

5.2. Two-dimensional low Reynolds flow past a National Advisory Committee for Aeronautics (NACA) airfoil

In this section, we simulate the incompressible flow around a NACA 0012 airfoil profile [39]. The computational domain is a 32×16 rectangle, with the trailing edge of an 8-unit-long airfoil placed at (16, 8). The horizontal inflow velocity is set to 1 at $x = 0$, and slip boundary conditions are applied at the superior and inferior walls of the computational domain. Velocity is prescribed to 0 at the airfoil surface.

The viscosity has been set to $\nu = 0.001$, which yields a Reynolds number $\mathrm{Re} = 1000$ based on the height of the airfoil. The time step has been set to $\delta t = 0.2$. In this numerical example, the CFL number associated to the finite element mesh was $CFL \sim 62$. A 29,945 linear element mesh has been used. The mesh is refined around the airfoil surface to be able to better capture the solution in the region surrounding the boundary layer.

5.2.1. Reduced-order model for $\alpha = 0.1$. In this numerical example, the angle of attack has been set to $\alpha = 0.1$. Regarding the ROM, 100 velocity–pressure snapshots have been taken and the 10 first reduced basis functions have been kept for the ROM. For the HROM, 100 additional snapshots from the right-hand side have been taken and the corresponding 10 first reduced basis functions have been kept. The number of sampling nodes is 30.

Figure 9 compares the velocity and pressure fields after 200 time steps for the FOM, ROM, and HROM. Very little difference can be appreciated between the results of the FOM and ROMs.

Figure 10 compares the pressure and velocity in the vertical direction at a control point in the wake behind the airfoil at coordinates (8, 0.5). It can be appreciated that the oscillation frequency is slightly larger for the FOM. The amplitudes for the vertical velocity are correctly captured. However, the amplitudes for the pressure at this control point are underestimated.

Figure 11 compares pressure and vertical velocity spectra at the same control point. It can be observed that the main frequency is underestimated by the ROMs. The velocity spectra of the ROMs almost exactly matches the FOM results for the lower most representative frequencies. Regarding the pressure, ROMs results do not match full-order results as closely, but again, the lower predominant frequency are correctly approximated.

DOI: 10.1002/fld

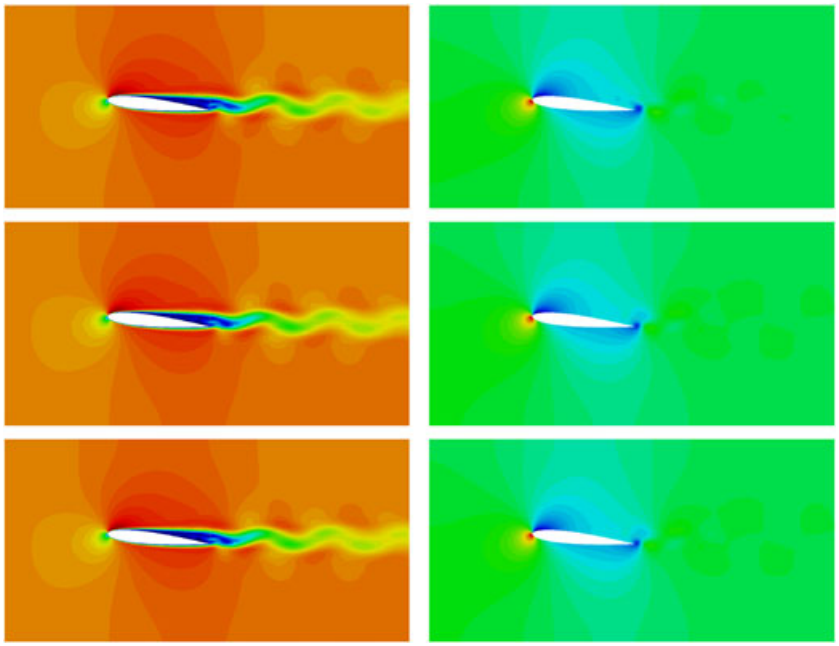

Figure 9. Velocity (left) and pressure (right) contours at Re = 1000, $\alpha = 0.1$ after 200 time steps. From top to bottom: full-order model, reduced-order model, and hyper-reduced order model.

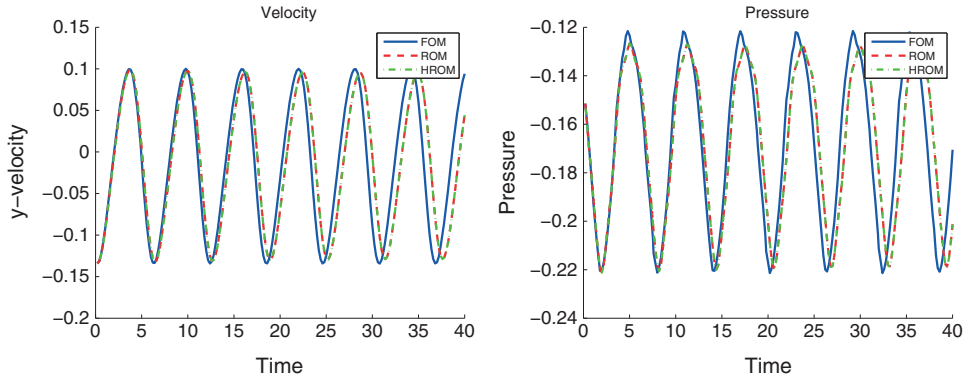

Figure 10. Velocity (left) and pressure (right) time history at a control point at the wake of the airfoil, Re = 1000, $\alpha = 0.1$.

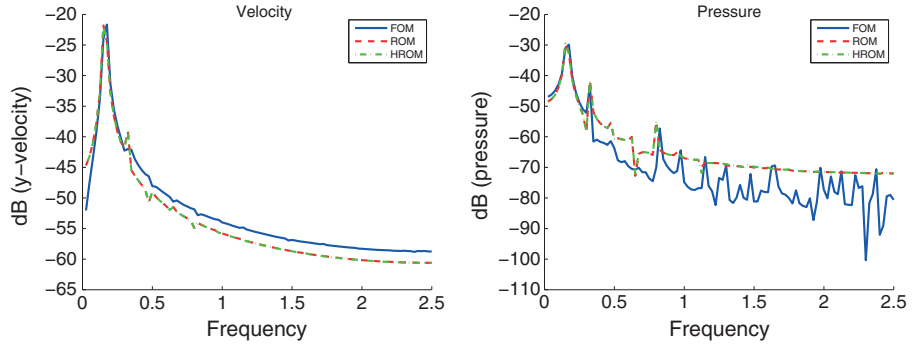

Figure 11. Velocity (left) and pressure (right) spectra at a control point at the wake of airfoil, Re = 1000, $\alpha = 0.1$.

DOI: 10.1002/fld

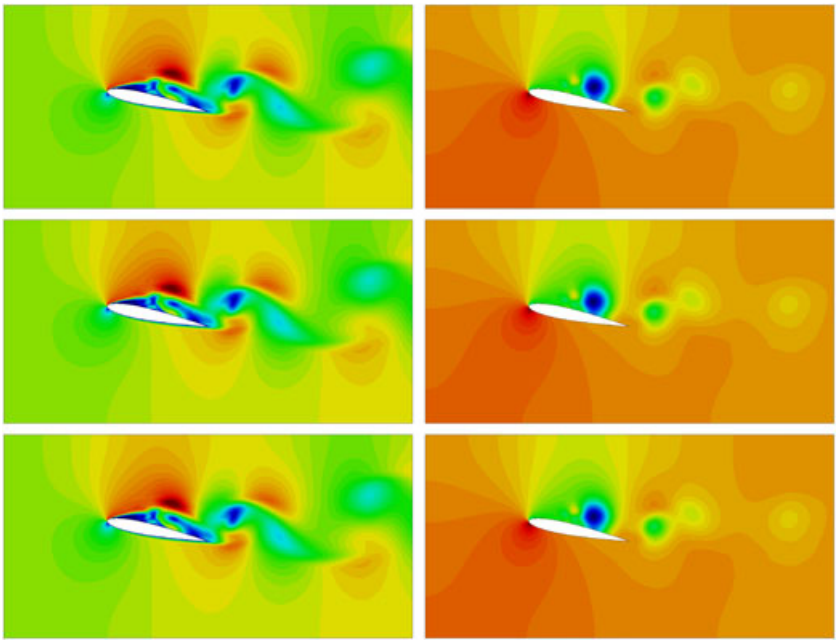

Figure 12. Velocity (left) and pressure (right) contours at Re $= 1000$, $\alpha = 0.2$ after 200 time steps. From top to bottom: full-order model, reduced-order model, and hyper-reduced-order model.

Regarding the computational cost, the FOM takes 148.9 s to run, the ROM takes 49.6 s (33%). Finally, ROM, 2, in which the computational cost depends only on the size of the ROM, takes only 0.71 s (0.45%) to run.

5.2.2. Reduced-order model for $\alpha = 0.2$, second-order time integration scheme. In this numerical example, the angle of attack has been set to $\alpha = 0.2$, and a second-order backward differences scheme has been used for the time integration.

A total of 100 velocity–pressure snapshots have been taken and the 10 first reduced basis functions have been kept for the ROM. For the HROM, 100 additional snapshots for the right-hand side have been taken and the corresponding 12 first reduced basis functions have been kept. The number of sampling nodes is 36.

Figure 12 compares the velocity and pressure fields after 200 time steps for the FOM, the reduced model, and the hyper-reduced model. In this case, ROMs almost exactly match results from the FOM.

Figures 13 and 14 show the time history and spectra for the velocity and pressure at (8, 0.5). Despite the complex flow and the high number of oscillation modes present in the solution, the ROMs manage to correctly capture the main modes amplitudes and frequencies.

It is also interesting to compare the performance of the explicit ROMs with that of an implicit ROM. For the implicit ROM, the used formulation is the same as in the FOM. Moreover, a Petrov–Galerkin projection is required because of the fact that the matrix arising from the finite element equations is not symmetric and positive definite. In this numerical example, we have used the Petrov–Galerkin projection described in [19, 24], which consists in using as left projection spaces the product $\boldsymbol{\Phi}_u^T \boldsymbol{A} \left(\boldsymbol{U}_{n+1}^{i-1}\right)^T$ instead of $\boldsymbol{\Phi}_u^T$ in (9). The same time step size as in the full order and the explicit ROMs was used for this implicit Petrov–Galerkin ROM.

Figure 15 compares the time history of the FOM with the implicit ROM, at the same control point. It can be observed that the performance of the implicit ROM (in which no hyper-reduction technique was applied) is very similar to the performance of the explicit one. The computational cost is also almost the same, 52.2 s.

DOI: 10.1002/fld

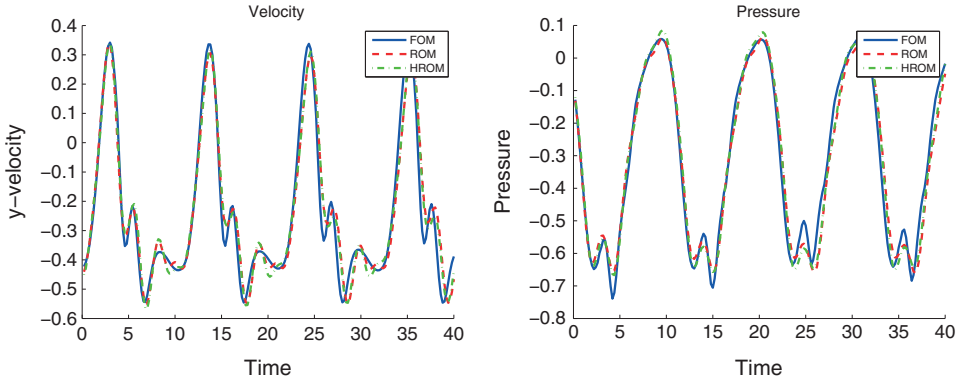

Figure 13. Velocity (left) and pressure (right) time history at a control point at the wake of the airfoil, $\mathrm{Re} = 1000$, $\alpha = 0.2$, second-order time integration.

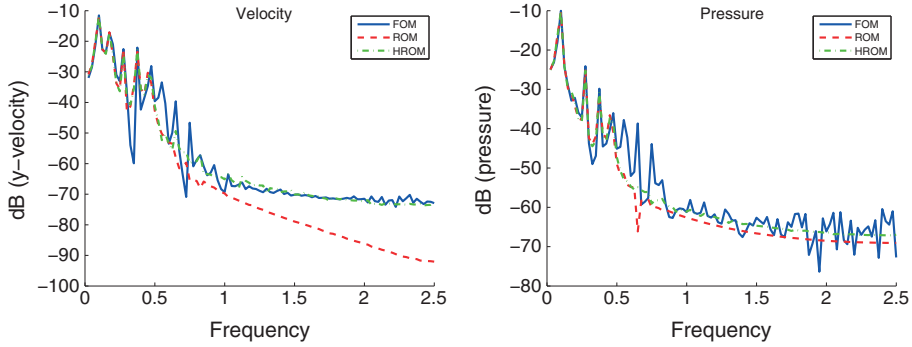

Figure 14. Velocity (left) and pressure (right) spectra at a control point at the wake of airfoil, $\mathrm{Re} = 1000$, $\alpha = 0.2$, second-order time integration.

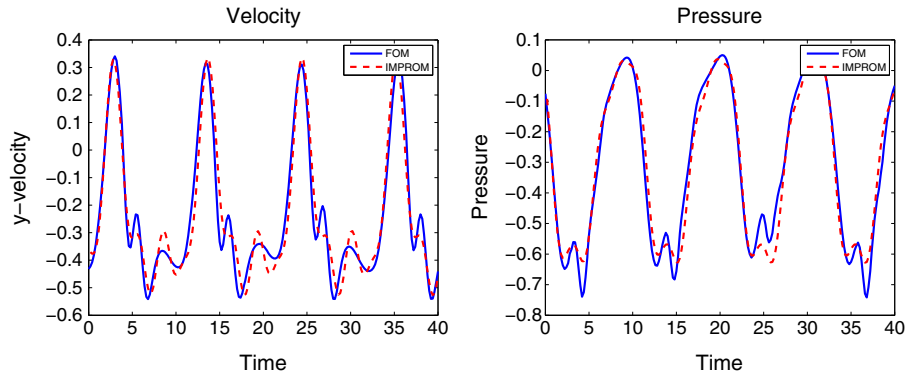

Figure 15. Velocity (left) and pressure (right) time history at a control point behind the airfoil, full order versus implicit reduced-order model (IMPROM).

5.3. Two-dimensional turbulent backward facing step, $\mathrm{Re} = 37,000$

In this numerical example, we study the incompressible flow in a backward facing step. The computational domain consists of a 44×9 rectangle. The 1-unit-height step is placed at $(4, 0)$. A horizontal inflow velocity profile with mean velocity 1 is set at $x = 0$. A wall law boundary condition is set at the top and bottom boundaries, with the wall distance characterizing the wall model being $\delta = 0.001$. The viscosity has been set to $\nu = 2.510^{-5}$, which yields a Reynolds number $Re = 37,000$ based

DOI: 10.1002/fld

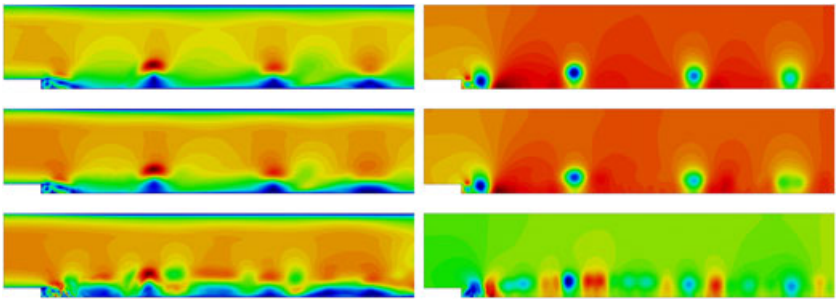

Figure 16. Velocity (left) and pressure (right) contours after 300 time steps. From top to bottom: full-order model, reduced-order model, and hyper-reduced order model.

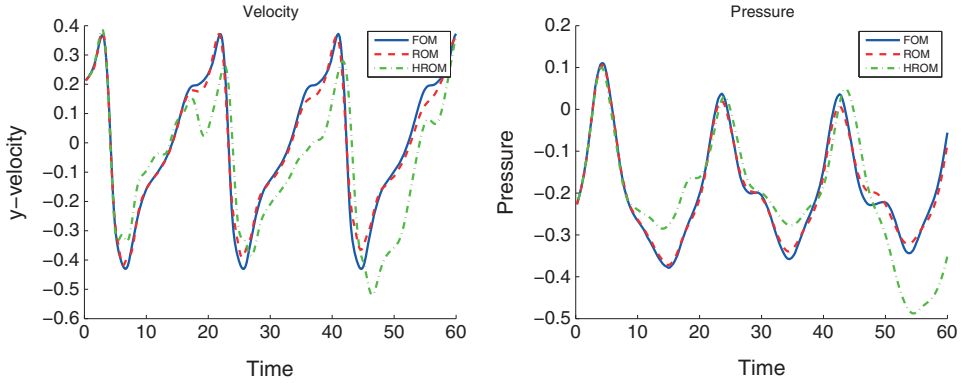

Figure 17. Velocity (left) and pressure (right) time history at a control point behind the step.

on the size of the step and the inflow velocity. A Smagorinsky turbulence model has been used for taking into account the turbulent effects, the Smagorinsky parameter being 0.1. For the time integration, a second-order backward differences scheme has been used with time step $\delta t = 0.2$. A 34,389 linear element mesh has been used to solve the problem.

Regarding the ROM, 200 velocity–pressure snapshots have been taken and the 35 first reduced basis functions have been kept for the reduced model. For the hyper-reduced model, 200 additional snapshots from the right-hand side have been taken and the corresponding 40 first reduced basis functions have been kept, and 280 sampling nodes have been considered (seven per basis function).

Figure 16 compares the velocity and pressure fields after 300 time steps. Although the ROM is able to capture the general pattern of the velocity and pressure fields, the HROM shows an unstable behavior in the area close to the vortex in the step wall.

Figure 17 compares the pressure and velocity in the vertical direction at a control point behind the step (5, 1). Because of the complex turbulent nature of the simulated flow, neither the ROM nor the hyper-reduced model are capable of exactly matching the time history evolution of the vertical velocity and the pressure at the control point. The ROM roughly represents the behavior of the FOM, but the amplitudes of the velocity and pressure oscillations quickly become smaller than the ones corresponding to the FOM. The HROM does not represent the behavior of the FOM correctly, the velocity and pressure significatively departing from the results of the FOM. However, the main frequency is correctly recovered. This can be explained by the unstable behavior observed in the region behind the step in Figure 16.

Figure 18 compares pressure and vertical velocity spectra at the same control point. The ROM correctly captures the main frequencies, the amplitudes being lower than in the FOM. The HROM, on the other hand, shows an excessive amplitude for some of main oscillation modes (see, for example, the second harmonic in the pressure spectra plot), although the frequencies are still correctly captured.

DOI: 10.1002/fld

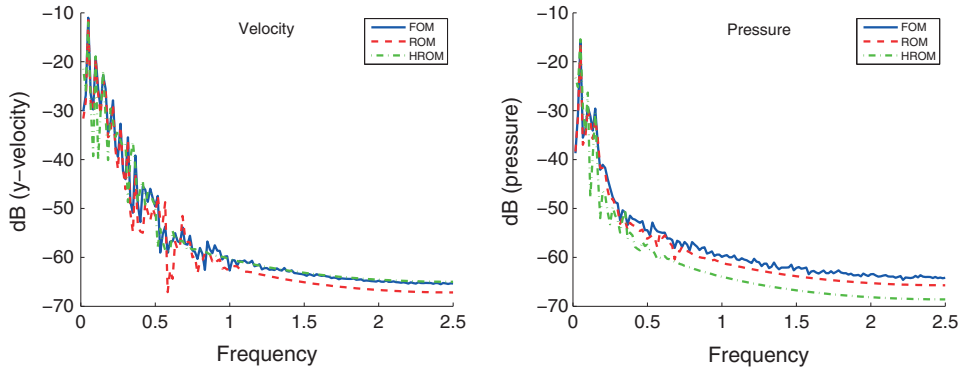

Figure 18. Velocity (left) and pressure (right) spectra at a control point behind the step.

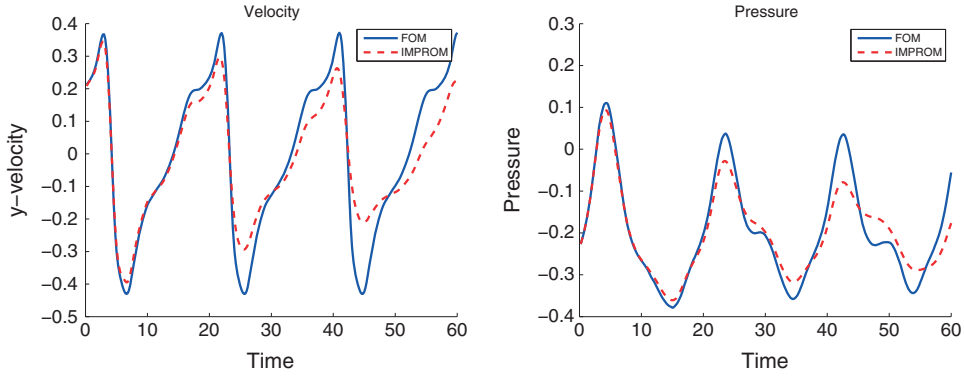

Figure 19. Velocity (left) and pressure (right) time history at a control point behind the step, full order versus (IMPROM).

Figure 19 compares the time history of the FOM with the implicit ROM at the same control point. We can observe that the implicit model does not perform better than the explicit model; in fact, the amplitude of the oscillation decreases faster in the implicit ROM. Therefore, we can conclude that the poor behavior of the explicit ROMs is due to the turbulent nature of the flow conditions.

Regarding the computational cost, the FOM takes 347.5 s to run and the ROM takes 167.2 s (48.3% of the original computational cost). Finally, the hyper-reduced model takes 5.56 s (1.6%) to run. The computational cost of the Petrov–Galerkin ROM (which was run without hyper-reduction) was 170.04 s (48.9%).

5.4. Three-dimensional incompressible flow past two cylinders

In this section, we simulate the three-dimensional flow past two cylinders. Again, a second-order backward differences scheme is used for the time integration. The simulation is performed in the $16 \times 8 \times 4$ prism. Two unit-diameter cylindrical struts are centered at (4, 4, 0) and (8, 4, 0). The income horizontal velocity is 1, slip boundary conditions are applied at the lateral walls and zero velocity is imposed at the struts surface. The viscosity is set to $\nu = 0.001$, the Reynolds number is $\mathrm{Re} = 1000$ based on the struts diameter. The time step is set to $\delta t = 0.1$. A 248,460 tetrahedra finite element mesh is used.

A total of 100 snapshots are taken for the reduced-order basis computation, of which the first 10 basis functions are kept for the ROM. Similarly, 100 right-hand side snapshots are taken for the hyper-reduced model, and the 20 first basis functions for the right-hand side are kept. The total number of sampling nodes is 60.

DOI: 10.1002/fld

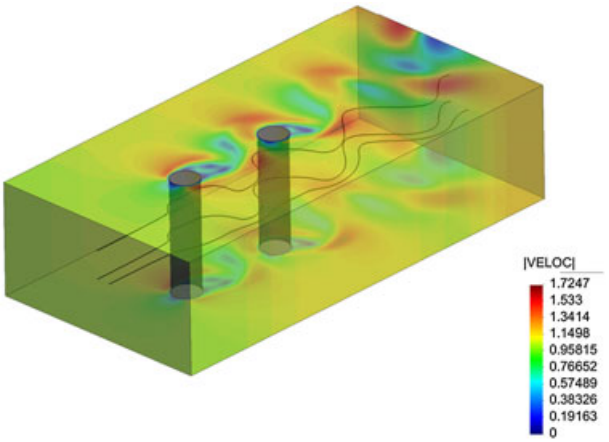

Figure 20. Velocity field and streamlines, two struts case.

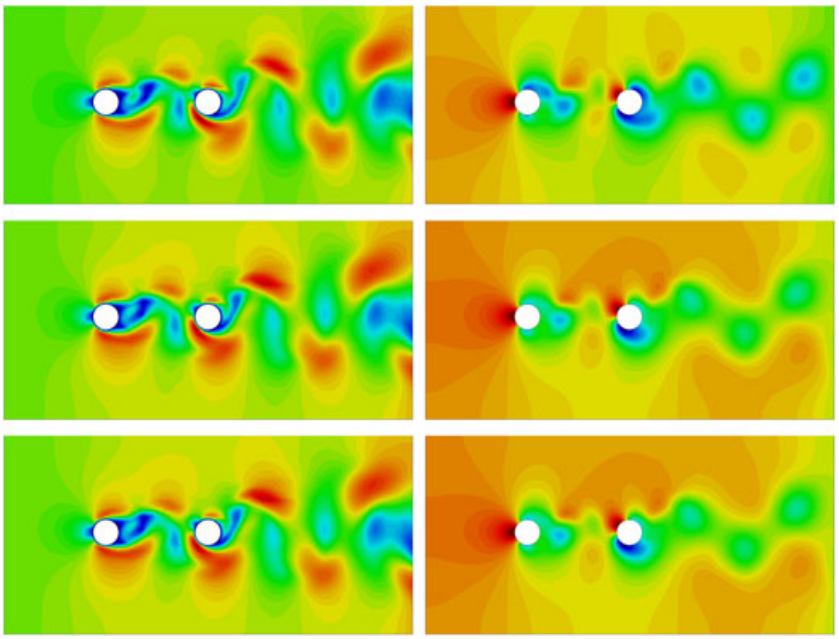

Figure 21. Velocity (left) and pressure (right) contours after 200 time steps, two struts case. From top to bottom: full-order model, reduced-order model, and hyper-reduced order model.

Figure 20 shows the velocity field and streamlines for results of the FOM computation. Figure 21 compares pressure and velocity fields after 200 time steps have been simulated for the full order and the ROMs. ROMs do not exactly match FOM results mainly because of the cumulative effect of the recovered main frequency error in the ROMs.

Figures 22 and 23 show the time history and spectra for the velocity and pressure at point (10.5, 4, 2). A good agreement is obtained between the ROMs and the FOM. In the case of the pressure field, pressure oscillation amplitudes are underestimated, but the frequency of the reduced models almost exactly matches the frequency of the FOM.

The FOM takes 2284 s to run a 200-time step simulation. The reduced model takes 703 s to run (30.7%), whereas the cost of running the hyper-reduced model is 5.15 s (0.23%), thanks to the fact that almost all the operations required for running the hyper-reduced model are of the order of the number of basis functions of the model.

DOI: 10.1002/fld

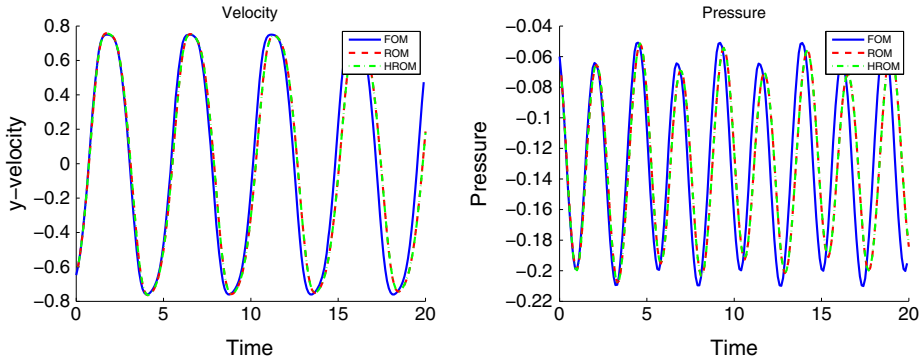

Figure 22. Velocity (left) and pressure (right) time history at a control point at the wake of the two struts.

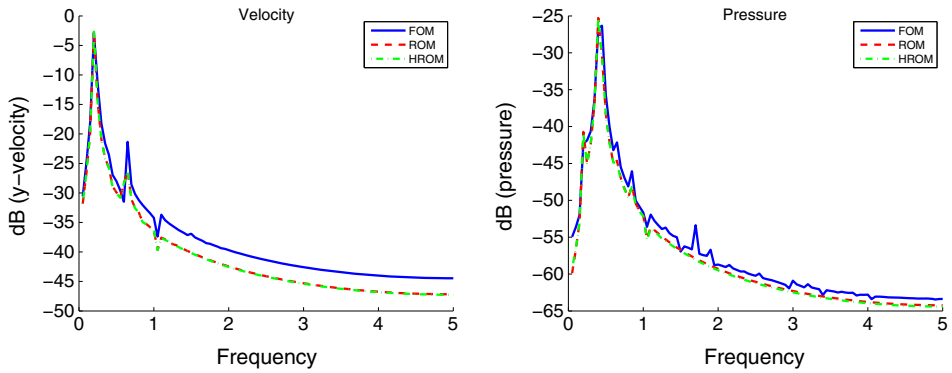

Figure 23. Velocity (left) and pressure (right) spectra at a control point at the wake of the two struts.

6. CONCLUSIONS

In this paper, we have presented an explicit formulation for ROMs of the incompressible stabilized finite element approximation of the incompressible Navier–Stokes equations. The basic idea is to treat all the terms except the mass matrix in the temporal derivative in an explicit way. This includes not only the nonlinear convective term but also the stabilization terms, which are highly nonlinear through the stabilization parameter τ. Although it is usually not possible to treat the pressure in the incompressible Navier–Stokes equations in an explicit way, this can be carried out in the ROM formulation because of the following:

- The ROM snapshots do already fulfill the continuity equation.
- The pressure field can be recovered from the reduced-order basis and the solution coefficients.

Although treating all the nonlinear terms in an explicit way does not result in any benefit when working with conventional ROMs, it allows for the easy use of HROMs, because only a single vector needs to be recovered by means of a gappy data reconstruction procedure. Despite the explicit nature of the ROM formulation, the Courant–Friedrichs–Levy condition can be violated, which can be explained by the fact that reduced basis functions expand over the whole computational domain. On the other hand, the ROM is sensitive to the addition of *noisy* basis functions, which can lead to an unstable behavior. The sensitivity of the explicit ROM to this issue can be improved by reducing the time step and refining the finite element mesh. It is also relevant to mention that we have had to treat the Galerkin contribution to the discrete variational problem with an approximation one order higher (formally) than expected, that is, second order for a first-order scheme and third order for a second-order scheme.

DOI: 10.1002/fld

Regarding the use of HROMs, we have presented a strategy for choosing the set of sampling components at the discrete level. The algorithm consists of selecting the sampling components that minimize the distance between the recovered reduced basis coefficients and the optimal coefficients (which are obtained by projecting the snapshots onto the reduced order subspace). The main advantage of the algorithm is that only values at the nodes of the finite element mesh are required for the gappy reconstruction, but these sampling nodes can be guaranteed to be locally optimal. This results in a strategy very convenient for the reconstruction of non-smooth functions, such as the right-hand side of the system of equations arising from the reduced-order strategy for the incompressible Navier–Stokes equations with the formulation used herein.

The performance of the proposed strategy has been tested in several numerical examples. The accuracy of the ROMs and the hyper-reduced models is good, showing little difference between the full-order results and the reduced-order results in non-turbulent cases. A turbulent test case has been tested where the ROMs were not able to fully reproduce the behavior of the FOM. We expect to improve the behavior of the reduced order and hyper-reduced models in turbulent flows by introducing the effect of the finite element part of the solution, which is not captured by the POD decomposition. This will hopefully enhance the stability of the reduced and hyper-reduced models with respect to turbulent phenomena.

Regarding the computational effort, the reduced-order model allows to save up to 65% of the computational cost, whereas in the case of the HROMs, the computational saving is larger than 99% the original computational cost.

ACKNOWLEDGEMENTS

This work was partially supported by the European Research Council under the Advanced Grant: ERC-2009-AdG 'Real Time Computational Mechanics Techniques for Multi-Fluid Problems'. Joan Baiges gratefully acknowledges the support received from the MICINN of the Spanish Government through a 'Juan de la Cierva' postdoctoral grant.

REFERENCES

1. Lihong F. Review of model order reduction methods for numerical simulation of nonlinear circuits. *Applied Mathematics and Computation* 2005; **167**(1):576–591.
2. Yvonnet J, He Q. The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. *Journal of Computational Physics* 2007; **223**(1):341–368.
3. Maier C, Epureanu A, Marinescu V, Bogdan FM. A new concept of the reduced order modeling in metal forming. In *Proceedings of the 6th WSEAS International Conference on Dynamical Systems and Control*, CONTROL'10. World Scientific and Engineering Academy and Society (WSEAS): Stevens Point, Wisconsin, USA, 2010; 133–136.
4. Burkardt J, Gunzburger M, Lee H. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **196**(1–3):337–355.
5. Galletti B, Bruneau CH, Zannetti L, Iollo A. Low-order modelling of laminar flow regimes past a confined square cylinder. *Journal of Fluid Mechanics* 2004; **503**:161–170.
6. Glaz B, Liu L, Friedmann PP. Reduced-Order nonlinear unsteady aerodynamic modeling using a Surrogate-Based recurrence framework. *AIAA Journal* 2010; **48**(10):2418–2429.
7. Kalashnikova I, Barone MF. Stable and efficient Galerkin reduced order models for Non-Linear fluid flow. In *AIAA-2011-3110, 6th AIAA Theoretical Fluid Mechanics Conference,* Honolulu, 2011.
8. Lucia DJ, Beran PS. Projection methods for reduced order models of compressible flows. *Journal of Computational Physics* 2003; **188**(1):252–280.
9. Veroy K, Patera AT. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids* 2005; **47**(8–9):773–788.
10. Wang Z, Akhtar I, Borggaard J, Iliescu T. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *arXiv:1106.3585*, 2011.
11. Akhtar I, Borggaard J, Hay A. Shape sensitivity analysis in flow models using a Finite-Difference approach. *Mathematical Problems in Engineering* 2010; **2010**:1–23.
12. Bergmann M, Cordier L, Brancher JP. Drag minimization of the cylinder wake by Trust-Region proper orthogonal decomposition. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* 2007; **95**:309–324.
13. Lassila T, Rozza G. Parametric free-form shape design with PDE models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(23–24):1583–1592.

DOI: 10.1002/fld

14. Rozza G, Lassila T, Manzoni A. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. In *Spectral and High Order Methods for Partial Differential Equations*, Vol. 76, Hesthaven JS, Ranquist EM (eds). Springer Berlin Heidelberg: Berlin, Heidelberg, 2011; 307–315.
15. Arian E, Fahl M, Sachs EW. Trust-Region proper orthogonal decomposition for flow control. *Technical Report*, Institute for Computers 2000:2000–2101.
16. Noack BR, Morzynski M, Tadmor G. *Reduced-Order Modelling for Flow Control*. Springer: New York, 2011.
17. Graham WR, Peraire J, Tang KY. Optimal control of vortex shedding using low-order models. Part I-open-loop model development. *International Journal for Numerical Methods in Engineering* 1999; **44**(7):945–972.
18. Barrault M, Maday Y, Nguyen NC, Patera AT. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* 2004; **339**(9):667–672.
19. Carlberg K, Bou-Mosleh C, Farhat C. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering* 2011; **86**(2):155–181.
20. Chaturantabut S, Sorensen DC. Discrete empirical interpolation for nonlinear model reduction. *Technical Report TR09-05*, Rice University, Houston Texas, 2009.
21. Drohmann M, Haasdonk B, Ohlberger M. Reduced basis approximation for nonlinear parameterized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing* 2012; **34**:937–962.
22. Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient Reduced-Basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis* 2007; **41**(03):575–605.
23. Nguyen NC, Peraire J. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *International Journal for Numerical Methods in Engineering* 2008; **76**(1):27–55.
24. Bui-Thanh T, Willcox K, Ghattas O. Model reduction for Large-Scale systems with High-Dimensional parametric input space. *SIAM Journal on Scientific Computing* 2008; **30**(6):3270–3288.
25. Hughes TJR. Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.
26. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
27. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
28. Codina R, Zienkiewicz OC. CBS versus GLS stabilization of the incompressible Navier–Stokes equations and the role of the time step as stabilization parameter. *Communications in Numerical Methods in Engineering* 2002; **18**:99–112.
29. Cuvelier C, Segal A, van Steenhoven A. *Finite Element Methods and Navier–Stokes Equations*. Reidel: Dordrecht, 1986.
30. Engelman MS, Strang G, Bathe K-J. The application of quasi-Newton methods in fluid mechanics. *International Journal for Numerical Methods in Engineering* 1981; **17**(5):707–718.
31. Chatterjee A. An introduction to the proper orthogonal decomposition. *Current Science* 2000; **78**(7):808–817.
32. Holmes P, Lumley JL, Berkooz G. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press: Cambridge, 1998.
33. Kosambi DD. Statistics in function space. *Journal of the Indian Mathematical Society* 1943; **7**:76–88.
34. Gunzburger MD, Peterson JS, Shadid JN. Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(4–6):1030–1047.
35. Ravindran SS. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids* 2000; **34**(5):425–448.
36. Everson R, Sirovich L. Karhunen-Loève procedure for gappy data. *Journal of the Optical Society of America A* 1995; **12**:1657–1664.
37. Astrid P, Weiland S, Willcox K, Backx T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control* 2008; **53**:2237–2251.
38. Marquardt DW. An algorithm for Least-Squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* 1963; **11**(2):431–441.
39. Jacobs EN, Ward KE, Pinkerton RM. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. *NACA Report,* 460, 1933.

DOI: 10.1002/fld