Comput. Methods Appl. Mech. Engrg. 267 (2013) 23–42

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

A domain decomposition strategy for reduced order models. Application to the incompressible Navier–Stokes equations

Joan Baiges [a,b,*], Ramon Codina [b,a], Sergio Idelsohn [a,1]

[a] Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Edifici C1, Campus Nord UPC, C/ Gran Capità S/N, 08034 Barcelona, Spain
[b] Universitat Politécnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 8 May 2013
Received in revised form 30 July 2013
Accepted 5 August 2013
Available online 20 August 2013

Keywords:
Reduced-order models
Domain decomposition
Navier–Stokes equations
Hyper-reduction

ABSTRACT

In this work, a domain decomposition strategy for non-linear hyper-reduced-order models is presented. The basic idea consists of restricting the reduced-order basis functions to the nodes belonging to each of the subdomains into which the physical domain is partitioned. An extension of the proposed domain decomposition strategy to a hybrid full-order/reduced-order model is then described. The general domain decomposition approach is particularized for the reduced-order finite element approximation of the incompressible Navier–Stokes equations with hyper-reduction. When solving the reduced incompressible Navier–Stokes equations, instabilities in the form of large gradients of the recovered reduced-order unknown at the subdomain interfaces may appear, which is the motivation for the design of additional stability terms giving rise to penalty matrices. Numerical examples illustrate the behavior of the proposed method for the simulation of the reduced-order systems, showing the capability of the approach to adapt to configurations which are not present in the original snapshot set.

1. Introduction

Reduced-order models applied to numerical design in engineering are a tool that is receiving increasing attention in the computational mechanics community. The key feature of reduced-order models 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 reduced-order models to several engineering problems like circuit design [28], multiscale modeling in solid mechanics [42] or metal forming processes [31]. Reduced-order models are also particularly interesting for computational fluid dynamics, where they have been applied to model the Navier–Stokes equations [12,18,19,29,24,39,40] and to applications like shape optimization [1,9,26,34] and flow control [4,6,20].

However, despite the important reduction in computational cost achieved by reduced-order models, the use of this kind of methods in day to day engineering is still not extended. One of the main reasons for this is the lack of robustness of reduced-order models with respect to changes in the parameters which characterize the numerical simulation. This lack of robustness causes the reduced model to be valid only in a small parameter region close to the parameter values for which the reduced model was built [4], requiring the snapshot collection and the reduced model to be updated when an optimization process leads to a parameter configuration which becomes too separated from the starting parameter set.

[*] Corresponding author at: Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Edifici C1, Campus Nord UPC, C/ Gran Capità S/N, 08034 Barcelona, Spain. Tel.: +34 934016206.

E-mail addresses: jbaiges@cimne.upc.edu (J. Baiges), ramon.codina@upc.edu (R. Codina), sergio@cimne.upc.edu (S. Idelsohn).

[1] ICREA Research Professor.

0045-7825/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cma.2013.08.001

The aim of the present paper is to develop a strategy which allows to improve the behavior of non-linear reduced models (where hyper-reduction is used for the reconstruction of the reduced-order equations) in parameter configurations which are not present in the snapshot set from which the reduced model is built. In order to do this, a domain decomposition (DD) strategy for reduced-order models (ROM) is proposed, which will be applied to reduced models built from finite element full-order models. Domain decomposition methods for reduced-order models have been used for different simulation problems in the past. Early approaches to domain decomposition techniques and reduced order models are presented in [30,27], where hybrid full-order/reduced-order strategies are described. In [10] several strategies for the iterative coupling of partitioned reduced-order systems are studied. In [3], a domain decomposition method is applied together with balanced truncation for the solution of convection–diffusion problems. This formulation is extended in [2] to the optimization of linear Stokes problems, and different partitions are introduced where one of the partitions is solved by means of a full-order system. A modular strategy for reduced-order modeling for fluid dynamics is presented in [41], where tools for coupling reduced models from various flow configurations are developed. A domain decomposition approach for solid mechanics problems using hyper-reduction techniques is presented in [33], where depending on the approximation error some of the local problems are solved by using the full-order equations.

Contrary to other DD for ROM approaches, the domain decomposition method we propose is obtained simply by restricting the reduced model basis functions to be non-null only in the nodes of the computational mesh belonging to the considered subdomain. For the finite element discretizations, this means that there are no interface nodes, but instead we have interface elements (that is, elements which have nodes from more than one subdomain). This definition of the partitioned problem directly ensures the continuity of the recovered reduced-order solution at the interfaces. Plus, it gives a straightforward definition of the DD-ROM, and allows the original computational tools developed for the reduced model to be used in the new DD setting, including the reconstruction of the reduced-order equations by using hyper-reduction techniques in the case of non-linear problems. Also, there is no need to use the classical domain-decomposition iteration by subdomain schemes, because the DD-ROM model is written in terms of the partitioned reduced bases in a monolithic way, although this kind of strategies could also be used.

An important issue is how to build the various reduced bases for the computational subdomains. In order to do this, two different approaches are presented which are based, on the one hand, in partitioning a global reduced basis into sub-bases, and, on the other, in partitioning the snapshots and building the local sub-bases from them. The features of each of these approaches are discussed.

One of the advantages of the method we propose is the ease for building a hybrid full-order/reduced-order model as a particular case of the general DD-ROM method. This extension to a hybrid model is directly obtained by taking as basis functions of the full-order subdomain the finite element shape functions of the nodes of the computational mesh. The hybrid system can then be built, with the only particularity that the reduced-order equations need to be tested against the reduced-order basis functions, while in the full-order equations the contribution of the reduced-order subdomain needs to be written in terms of the reduced-order unknowns. The proposed hybrid DD-ROM model can be easily used together with hyper-reduced models or reduced-models in which a Petrov–Galerkin projection is used for writing the reduced equations.

Depending on the nature of the simulated problem and the time integration scheme selected for the reduced models, instabilities in the form of large gradients of the recovered reduced-order unknowns at the subdomain interfaces may appear when the proposed domain decomposition strategy is straightforwardly applied; this is the motivation for the design of additional stability terms, introduced as penalty matrices. This penalization strategy turns out to be able to completely eliminate the oscillations at the subdomain borders.

The numerical simulation of the incompressible Navier–Stokes equations is of great importance in engineering, also in design problems where reduced models may play an important role. However, the particular nature of the Navier–Stokes equations causes that general problem aimed numerical strategies usually need to be adapted when applied to the solution of incompressible flows. This is why we focus in particularizing the general domain decomposition approach for reduced-order problems to the case of the finite element approximation of the incompressible Navier–Stokes equations. The main difference with respect to the original approach is that only the equality of velocities is enforced at the interface or overlapping regions, while no additional constraints are set on the pressure field.

The paper is organized as follows. In Section 2 we describe a reduced-order model strategy for general variational problems and the proper orthogonal decomposition (POD) method. In Section 3, a monolithic approach for domain decomposition in reduced-order methods is presented. The localized global POD (LG-POD) and local POD (L-POD) approaches for building the subdomain reduced bases are discussed. In Section 4 the method for stabilizing general domain-decomposition reduced models based on penalizing the equations at the interface/overlapping between subdomains is presented. The relationship between the original non-partitioned reduced-order approach and the domain decomposition approach with stabilization is shown. In Section 5 the extension to a hybrid full-order/reduced-order model is described. Finally, in Section 6, the particularities of the application of the previously described methods to the incompressible Navier–Stokes equations are explained. In Section 7, numerical examples illustrate the behavior of the proposed method for the simulation of the reduced-order incompressible Navier–Stokes equations in a domain decomposition setting. The stability of the method is assessed in a domain decomposition reduced-order problem, and a numerical example is presented where the capability of the hybrid full-order/reduced-order approach to adapt to flow configurations which are not present in the original snapshot set is shown. Finally, some conclusions and remarks close the paper in Section 8.

2. Reduced order approximation of variational problems

In this section a reduced order modeling (ROM) approach for the approximation of non-linear, time dependent variational problems is summarized. This formulation is to be used in the following sections for the domain decomposition strategy for reduced models.

2.1. POD for the reduced-order modeling of general problems

Let us define a global unknown $\boldsymbol{U} \in \mathbb{R}^M$, where M can be identified with the number of nodes of the computational mesh if we are considering the equations arising from a finite element problem. Suppose that after linearizing and fully discretizing in time and space a given variational problem, the following matrix form is obtained which allows to obtain the vector of nodal unknowns $\boldsymbol{U}$ at a given iteration of the non-linear procedure, for a certain time step:

$$\boldsymbol{A}\boldsymbol{U} = \boldsymbol{F}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{M\times M}$ is the matrix of the system whose solution is $\boldsymbol{U}$, and $\boldsymbol{F} \in \mathbb{R}^M$ the RHS vector. Let us emphasize that $\boldsymbol{A}$ and $\boldsymbol{F}$ might have a dependence on the values of $\boldsymbol{U}$ in previous iterations or time steps.

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

$$\boldsymbol{U} \approx \boldsymbol{\Phi}\boldsymbol{\alpha}, \tag{2}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ is the basis for $\mathcal{U}$ and N is the dimension of the reduced order model, with $N < M$. $\boldsymbol{\alpha} \in \mathbb{R}^N$ are the components in $\mathcal{U}$ expressed in the reference system defined by $\boldsymbol{\Phi}$. There are several methods for obtaining the reduced order basis $\boldsymbol{\Phi}$. In the reduced order method we propose, the reduced basis is obtained by means of a proper orthogonal decomposition (POD) procedure [14,25,22]. A relevant property of POD bases is that the recovered basis functions are orthonormal.

After introducing the approximation (2) in (1) an overdetermined system with M equations and N unknowns is obtained. As explained in [11,13], if matrix $\boldsymbol{A}$ is symmetric and positive definite, a least-squares strategy for approximating the linearized overdetermined system:

$$\boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{\alpha} = \boldsymbol{F},$$

leads to:

$$\boldsymbol{\Phi}^T\boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{\alpha} = \boldsymbol{\Phi}^T\boldsymbol{F}. \tag{3}$$

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 [11].

2.2. Hyper-reduction

The system defined in equation (3) is the reduced order model associated to the original system (1). However, although the system of equations (3) is of dimension N (which typically corresponds to few degrees of freedom), matrix $\boldsymbol{A}$ and vector $\boldsymbol{F}$ still need to be built at each time step, and the cost of this is of order M, the dimension of the original system (1). Recently, a general approach for dealing with the non-linear terms of reduced order models has appeared [32,21,8,15,35,36,5,37,38], giving rise to what are known as *hyper-reduced* order models. In these methods, the equations governing the problem of interest are only integrated at certain reduced-order integration points, instead of integrating them in the Gauss points of the full-order problem. The non-linear and parameter-dependent terms can then be recovered by means of a least-squares procedure from the values at the reduced-order integration points where the function to be approximated is computed. The use of hyper-reduction approaches for reconstructing $\boldsymbol{A}$ and $\boldsymbol{F}$ allows one to reduce the computational cost associated to the construction of the ROM system to $\mathcal{O}(N)$.

In [7] an explicit hyper-reduction strategy for the incompressible Navier–Stokes equations is presented. This is also the hyper-reduction approach used in the sections devoted to the Navier–Stokes equations. The basic idea consists of first, deriving an explicit time integration scheme for the reduced-order incompressible Navier–Stokes equations. This is possible because the snapshots for the POD decomposition do already fulfill a stabilized continuity equation. Secondly, a hyper-reduction technique for the reconstruction of the non-linear right-hand side of the system can be applied. On the other hand, the system matrix needs not to be reconstructed, since it is linear. The attained computational cost reduction by using this method can reach a 99% of the original computational cost.

3. Monolithic domain decomposition strategy for reduced order models (DD-ROM)

As we have already explained, the joint use of domain decomposition strategies and reduced order models can have several advantages. In this section we present a general framework for monolithic domain decomposition strategies for reduced order models. Let us consider the splitting of the computational domain Ω into two subdomains Ω_k, $k = 1, 2$, and the associated local unknowns $\boldsymbol{U}^k \in \mathbb{R}^{M_k}$, $M = M_1 + M_2$. If the domain decomposition is applied to the equations arising from a finite element problem, the partition into subdomains is done by assigning each of the nodes (and nodal unknowns) of the finite element mesh to a subdomain. This means that there are no interface nodes, instead we define interface elements as those elements who own nodes from different subdomains. Let us define a local reduced order basis $\boldsymbol{\phi}^k$ consisting of the reduced basis functions $\boldsymbol{\phi}_i^k \in \mathbb{R}^{M_k}$, $i = 1, \dots, N_k$, to approximate $\boldsymbol{U}^k$ in each subdomain. Note that the number of basis functions in each subdomain is not necessarily the same. The possible ways to construct this basis are discussed later. This local basis can be extended to the global domain by defining $\boldsymbol{\Phi}_i^k \in \mathbb{R}^M$:

$$\boldsymbol{\Phi}_i^1 := \begin{bmatrix} \boldsymbol{\phi}_i^1 \\ \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{\Phi}_i^2 := \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\phi}_i^2 \end{bmatrix}, \tag{4}$$

where the null terms correspond to components of the global system which lie outside Ω_k. Taking this into account, the unknown $\boldsymbol{U}$ is approximated as:

$$\boldsymbol{U} \approx \sum_{i=1}^{N} (\boldsymbol{\Phi}_i^1 \alpha_i^1 + \boldsymbol{\Phi}_i^2 \alpha_i^2) = (\boldsymbol{\Phi}^1 \boldsymbol{\alpha}^1 + \boldsymbol{\Phi}^2 \boldsymbol{\alpha}^2), \quad \boldsymbol{\Phi}^k \in \mathbb{R}^{M \times N}, \boldsymbol{\alpha}^k \in \mathbb{R}^N \; k = 1, 2, \tag{5}$$

where $\boldsymbol{\alpha}^k$ are the solution coefficients for subdomain k.

Let $\boldsymbol{A} \in \mathbb{R}^{M \times M}$ be the matrix of the system whose solution is $\boldsymbol{U} \in \mathbb{R}^M$, and $\boldsymbol{F} \in \mathbb{R}^M$ the RHS vector. They can be partitioned into the components associated to each subdomain Ω_k, $k = 1, 2$, so that

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}|_{11} & \boldsymbol{A}|_{12} \\ \boldsymbol{A}|_{21} & \boldsymbol{A}|_{22} \end{bmatrix}, \quad \boldsymbol{A}|_{kl} \in \mathbb{R}^{M_k \times M_l}, \quad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}|_1 \\ \boldsymbol{F}|_2 \end{bmatrix}, \quad \boldsymbol{F}|_k \in \mathbb{R}^{M_k}.$$

The monolithic approach for the domain decomposition ROM is obtained by introducing the union of the extensions of the local bases to the global domain as the global reduced order basis:

$$\begin{aligned} &(\boldsymbol{\Phi}^1)^T \boldsymbol{A}(\boldsymbol{\Phi}^1 \boldsymbol{\alpha}^1 + \boldsymbol{\Phi}^2 \boldsymbol{\alpha}^2) = (\boldsymbol{\Phi}^1)^T \boldsymbol{F}, \\ &(\boldsymbol{\Phi}^2)^T \boldsymbol{A}(\boldsymbol{\Phi}^1 \boldsymbol{\alpha}^1 + \boldsymbol{\Phi}^2 \boldsymbol{\alpha}^2) = (\boldsymbol{\Phi}^2)^T \boldsymbol{F}. \end{aligned} \tag{6}$$

If we also consider the decomposition of $\boldsymbol{A}$ and $\boldsymbol{F}$ the final reduced order system can be written in terms of the local bases $\boldsymbol{\phi}^k$:

$$\begin{aligned} &(\boldsymbol{\phi}^1)^T (\boldsymbol{A}|_{11} \boldsymbol{\phi}^1 \boldsymbol{\alpha}^1 + \boldsymbol{A}|_{12} \boldsymbol{\phi}^2 \boldsymbol{\alpha}^2) = (\boldsymbol{\phi}^1)^T \boldsymbol{F}|_1, \\ &(\boldsymbol{\phi}^2)^T (\boldsymbol{A}|_{21} \boldsymbol{\phi}^1 \boldsymbol{\alpha}^1 + \boldsymbol{A}|_{22} \boldsymbol{\phi}^2 \boldsymbol{\alpha}^2) = (\boldsymbol{\phi}^2)^T \boldsymbol{F}|_2. \end{aligned}$$

The off diagonal block matrices correspond to the coupling terms and are null except for the contribution of the unknowns ubicated at the domain interfaces. It can be observed that the cost of computing the ROM system is not larger than in the monolithic approach. However, the size of the reduced system is larger (dimension $2N$).

There are different possibilities for building the local reduced order bases which are discussed in the next subsections. An important point is that each algebraic local basis function $\boldsymbol{\Phi}_i^k$ arises from a function defined in space. This spatial function is a

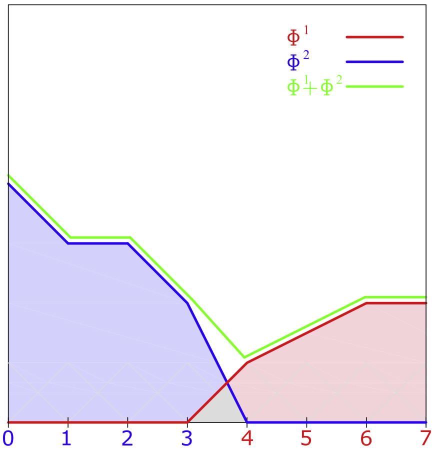

Fig. 1. Local basis functions for the domain decomposition approach. The green function is the sum of the local basis function of the left subdomain (blue) and the local basis function of the right subdomain (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

linear combination of the finite element shape functions of the nodes of subdomain Ω_k. As a consequence, each of the components in $\mathbf{\Phi}_i^k$ corresponds to a nodal value of the spatial field to be represented on the finite element mesh. This is illustrated in Fig. 1, where examples of local basis functions for a one-dimensional problem and linear finite elements are shown. Let us also emphasize that, if the original finite element shape functions are continuous, any local (and global) basis function will also be continuous, as a consequence of the definition of the extension of the basis functions to the global domain (4). This will also hold for the combination of local basis functions, even if these belong to different subdomains. In Fig. 1 the blue basis function belongs to the left subdomain, the red basis function belongs to the right subdomain. The green line represents the addition of the blue and the red basis functions. Since both of the original functions are continuous, the green function is also continuous. Note also that there is an overlapping region where both the left and right basis functions are non-zero.

3.1. Localized global POD (LG-POD)

In this approach the local, subdomain-restricted bases are obtained from the original global domain POD basis. Firstly, a global POD basis is obtained from the global array of snapshots. Secondly, the global POD basis is straightforwardly partitioned and the local bases for each subdomain are obtained. The strategy can be summarized as follows:

- Compute an initial global POD basis $\mathbf{\Phi}^0 \in \mathbb{R}^{M\times N}$.
- Compute the local POD bases $\boldsymbol{\phi}^k \in \mathbb{R}^{M_k\times N}$ as the restriction of the original basis $\mathbf{\Phi}^0$ to each subdomain:

$$\boldsymbol{\phi}^k = \mathbf{\Phi}^0|_k.$$

- The associated global basis $\mathbf{\Phi} \in \mathbb{R}^{M\times 2N}$ is directly recovered as:

$$\mathbf{\Phi} = \left[\mathbf{\Phi}^1, \mathbf{\Phi}^2\right]. \tag{7}$$

Note that this strategy does not guarantee the orthonormality of the resulting local bases because the norm of the partitioned local basis functions is not unitary. Moreover, even if the functions of the local basis are normalized, they cannot be guaranteed to be orthogonal. However, we have checked in our numerical examples that this approach yields well-conditioned systems of equations and accurate results in the studied cases.

Despite its drawbacks, the simplicity of this straightforward approach allows a better understanding of some of the topics to be discussed later.

3.2. Local POD (L-POD)

The local POD strategy consists in performing a POD for the part of the snapshots corresponding to each of the subdomains. Instead of taking the global POD and decomposing it into the contribution to each of the subdomains, the snapshots are first partitioned according to the domain decomposition strategy and the local basis $\boldsymbol{\phi}^k$ is obtained from these partitioned snapshots. The global basis is again defined as $\mathbf{\Phi} = \left[\mathbf{\Phi}^1, \mathbf{\Phi}^2\right]$. Note that the number of local basis functions in each subdomain does not necessarily coincide, $N_1 \neq N_2$, $N = N_1 + N_2$.

There are some advantages which favor the use of this strategy instead of the localized global POD:

- The local basis can be ensured to be orthonormal at the algebraic level. By construction, each of the basis functions which conform the local POD has unitary norm and is orthogonal to all the basis functions in its subdomain at the algebraic level. Moreover, due to the domain decomposition approach, the projection of a local basis of a given subdomain onto the space conformed by the basis functions of any other subdomain is also zero. This ensures that if we consider the POD decomposition globally, the union of the local bases is also an orthonormal basis. This also ensures that the space spanned by the union of the local bases is greater (or equal) than the space spanned by the localized global POD.
- The computation of the singular value decomposition of the local snapshots for each subdomain requires less memory than the computation of the singular value decomposition of the global snapshots.

In the case we are using hyper reduced models which require additional POD bases for reconstructing the system matrix and right-hand side, we can proceed in the same way.

Once the localized reduced order bases have been defined, the monolithic domain decomposition reduced order model is obtained by using as reduced basis the union of the local reduced bases. The fact that the basis functions are local makes the computational cost diminish with respect to the global approach with the same number of basis functions, because the operations can be done at the local level. However, the number of functions is usually larger in the domain decomposition approach, because a sufficient number of components needs to be assigned to the reduced basis of each subdomain in order to properly represent the solution in that subdomain.

4. Stabilization through overlapping and penalty terms

Despite the simplicity of the previously described domain decomposition strategy, some instabilities appear at the interfaces in the form of increasingly large velocity and pressure gradients when this approach is applied to the incompressible Navier–Stokes equations. These instabilities are more apparent when an explicit time integration scheme is used for the time marching of the reduced order model. In this case it is clear that the proposed scheme is equivalent to an explicit iteration-by-subdomain strategy, which is known to have convergence and stability issues. The instabilities do also manifest for the implicit ROM with the Petrov–Galerkin projection [11], though.

4.1. Penalty term for the LG-POD

The motivation of the stabilization strategy we propose arises when the LG-POD strategy described in Section 3.1 is used to compute the local reduced order bases. In this case one would expect to obtain the same solution coefficients for the reduced order solution of each subdomain. The coefficients should coincide with the solution coefficients of the global approach (that is, without domain decomposition), because the local basis functions are simply the restriction of the global basis functions to a given subdomain. However, it can be seen that this is not the actual case:

One-domain ROM system. The algebraic system for the one-domain ROM problem can be written in terms of the partitioned domains as:

$$\left[\boldsymbol{\Phi}^0|_1^T, \boldsymbol{\Phi}^0|_2^T\right]\begin{bmatrix}\boldsymbol{A}|_{11} & \boldsymbol{A}|_{12}\\ \boldsymbol{A}|_{21} & \boldsymbol{A}|_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}^0|_1\\ \boldsymbol{\Phi}^0|_2\end{bmatrix}\boldsymbol{\alpha} = \left[\boldsymbol{\Phi}^0|_1^T, \boldsymbol{\Phi}^0|_2^T\right]\begin{bmatrix}\boldsymbol{F}|_1\\ \boldsymbol{F}|_2\end{bmatrix}.$$

Defining $\boldsymbol{a}_{kl} = (\boldsymbol{\Phi}^k)^T\boldsymbol{A}\boldsymbol{\Phi}^l \in \mathbb{R}^{N\times N}$ and $\boldsymbol{f}_k = (\boldsymbol{\Phi}^k)^T\boldsymbol{F} \in \mathbb{R}^N$, we may write this system as

$$(\boldsymbol{a}_{11} + \boldsymbol{a}_{12} + \boldsymbol{a}_{21} + \boldsymbol{a}_{22})\boldsymbol{\alpha} = \boldsymbol{f}_1 + \boldsymbol{f}_2. \tag{8}$$

Two-domain ROM system. Using approximation (5), vectors $\boldsymbol{\alpha}^k \in \mathbb{R}^N k = 1, 2$, may be obtained from Eq. (6). Using the above notation:

$$\boldsymbol{a}_{11}\boldsymbol{\alpha}^1 + \boldsymbol{a}_{12}\boldsymbol{\alpha}^2 = \boldsymbol{f}_1, \tag{9}$$
$$\boldsymbol{a}_{21}\boldsymbol{\alpha}^1 + \boldsymbol{a}_{22}\boldsymbol{\alpha}^2 = \boldsymbol{f}_2. \tag{10}$$

Obviously, the solution to this system *is not* necessarily $\boldsymbol{\alpha}^1 = \boldsymbol{\alpha}^2$, which would correspond to (8). In fact, numerical experiments indicate that this solution displays instabilities for some problems and time integration schemes. Note that here we assume that it is possible to prescribe $\boldsymbol{\alpha}^1 = \boldsymbol{\alpha}^2$. A remark will be made when applying this ideas to the incompressible Navier–Stokes equations.

Penalizing the two-domain ROM system. The idea now is to modify system (9) and (10) by introducing a penalty term that yields $\boldsymbol{\alpha}^1 = \boldsymbol{\alpha}^2$ in the limit. Let $\varepsilon > 0$ be a given small parameter, and let $\boldsymbol{M}_k \in \mathbb{R}^{N\times N}$, $k = 1, 2$, be two invertible matrices to be determined. Treating $\boldsymbol{\alpha}^1 = \boldsymbol{\alpha}^2$ as a penalized constraint in (9) and (10) yields

$$\boldsymbol{a}_{11}\boldsymbol{\alpha}^1 + \boldsymbol{a}_{12}\boldsymbol{\alpha}^2 + \frac{1}{\varepsilon}\boldsymbol{M}_1(\boldsymbol{\alpha}^1 - \boldsymbol{\alpha}^2) = \boldsymbol{f}_1, \tag{11}$$
$$\boldsymbol{a}_{21}\boldsymbol{\alpha}^1 + \boldsymbol{a}_{22}\boldsymbol{\alpha}^2 + \frac{1}{\varepsilon}\boldsymbol{M}_2(\boldsymbol{\alpha}^2 - \boldsymbol{\alpha}^1) = \boldsymbol{f}_2. \tag{12}$$

Let us show that the solution converges to that of (8) as $\varepsilon \to 0$. From (11) it is found that

$$\boldsymbol{\alpha}^1 = \left(\boldsymbol{a}_{11} + \frac{1}{\varepsilon}\boldsymbol{M}_1\right)^{-1}\left[\boldsymbol{f}_1 - \left(\boldsymbol{a}_{12} - \frac{1}{\varepsilon}\boldsymbol{M}_1\right)\boldsymbol{\alpha}^2\right],$$

which inserted in (12) yields

$$\left[-\left(\boldsymbol{a}_{21} - \frac{1}{\varepsilon}\boldsymbol{M}_2\right)\left(\boldsymbol{a}_{11} + \frac{1}{\varepsilon}\boldsymbol{M}_1\right)^{-1}\left(\boldsymbol{a}_{12} - \frac{1}{\varepsilon}\boldsymbol{M}_1\right) + \left(\boldsymbol{a}_{22} + \frac{1}{\varepsilon}\boldsymbol{M}_2\right)\right]\boldsymbol{\alpha}^2 = \boldsymbol{f}_2 - \left(\boldsymbol{a}_{21} - \frac{1}{\varepsilon}\boldsymbol{M}_2\right)\left(\boldsymbol{a}_{11} + \frac{1}{\varepsilon}\boldsymbol{M}_1\right)^{-1}\boldsymbol{f}_1. \tag{13}$$

We have that

$$\left(\boldsymbol{a}_{11} + \frac{1}{\varepsilon}\boldsymbol{M}_1\right)^{-1} = \varepsilon\boldsymbol{M}_1^{-1}\left(I + \varepsilon\boldsymbol{M}_1^{-1}\boldsymbol{a}_{11}\right)^{-1} = \varepsilon\boldsymbol{M}_1^{-1} - \varepsilon^2\boldsymbol{M}_1^{-2}\boldsymbol{a}_{11} + \mathcal{O}(\varepsilon^3).$$

Using this in (13) yields

$$(\boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{a}_{11} + \boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{a}_{12} + \boldsymbol{a}_{21} + \boldsymbol{a}_{22})\boldsymbol{\alpha}^2 + \mathcal{O}(\varepsilon) = \boldsymbol{f}_1 + \boldsymbol{M}_2\boldsymbol{M}_1^{-1}\boldsymbol{f}_2 + \mathcal{O}(\varepsilon),$$

from where it is seen that the solution of system (11) and (12) converges to that of (8) as $\varepsilon \to 0$ if

- $\boldsymbol{M}_2\boldsymbol{M}_1^{-1} = \boldsymbol{I}$ (for example if $\boldsymbol{M}_1 = \boldsymbol{M}_2$) or if
- a Petrov–Galerkin weighting is used for the one-domain ROM, of the form

$$\left[\boldsymbol{M}_2\boldsymbol{M}_1^{-1}(\boldsymbol{\Phi}^1)^T + (\boldsymbol{\Phi}^2)^T\right]\begin{bmatrix}\boldsymbol{A}|_{11} & \boldsymbol{A}|_{12}\\ \boldsymbol{A}|_{21} & \boldsymbol{A}|_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}^0|_1\\ \boldsymbol{\Phi}^0|_2\end{bmatrix}\boldsymbol{\alpha} = \left[\boldsymbol{M}_2\boldsymbol{M}_1^{-1}(\boldsymbol{\Phi}^1)^T + (\boldsymbol{\Phi}^2)^T\right]\begin{bmatrix}\boldsymbol{F}|_1\\ \boldsymbol{F}|_2\end{bmatrix}.$$

In this case, $\boldsymbol{\alpha}$ should be identified with $\boldsymbol{\alpha}^2$ ($\boldsymbol{\alpha}$ could be identified with $\boldsymbol{\alpha}^1$ proceeding similarly).

In the practical cases we take:

$$\boldsymbol{M}_1 = \boldsymbol{M}_2 = \begin{bmatrix}\boldsymbol{\Phi}^0|_1\\ \boldsymbol{\Phi}^0|_2\end{bmatrix}^T\begin{bmatrix}\boldsymbol{\Phi}^0|_1\\ \boldsymbol{\Phi}^0|_2\end{bmatrix} = \boldsymbol{I}, \tag{14}$$

due to the orthonormality of the global POD basis. However, other choices are possible for $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$.

The introduction of the $\boldsymbol{M}$ matrices to the reduced order formulation allows one to obtain a stable solution in the practical cases. The stabilization parameter ε is chosen so that, on the one hand, the penalty terms are sufficiently large to provide the desired stabilization effects, and on the other, the norm of $\frac{1}{\varepsilon}\boldsymbol{M}$ is proportional to the norm of $\boldsymbol{A}$. In this way we ensure that the resulting system does not become ill-conditioned. In the numerical examples we have done so by computing the $\frac{1}{\varepsilon}$ parameter for each subdomain as:

$$\frac{1}{\varepsilon} = \frac{1}{\gamma}\sqrt{\left(\sum_{i=1}^{N}(\boldsymbol{M}_{ii})^2\right)\Bigg/\left(\sum_{i=1}^{N}\left([\boldsymbol{\Phi}^T\boldsymbol{A}\boldsymbol{\Phi}]_{ii}\right)^2\right)}, \tag{15}$$

where γ is a dimensionless algorithmic constant.

4.2. Extension to L-POD through overlapping

In this section we extend the penalty terms defined in the previous section to the L-POD reduced bases. In this case the equality $\boldsymbol{\alpha}^1 = \boldsymbol{\alpha}^2$ cannot be imposed, because the local reduced order bases are not defined as the restriction of a global basis but are computed independently. The only way to establish a linkage between $\boldsymbol{\alpha}^1$ and $\boldsymbol{\alpha}^2$ is by means of an *overlapping region.* As in classical iteration by subdomain strategies, the overlapping region $\Omega_\cap$ is the part of Ω which belongs to both Ω_1 and Ω_2. In our approach, in which the partitioning is obtained by assigning the nodes of the finite element mesh to Ω_1 and Ω_2, overlapping is achieved by allowing some nodes close to the interface to belong to both Ω_1 and Ω_2. The local reduced bases are computed by performing the POD of the restriction of the snapshots to Ω_k, but the obtained basis functions need to be corrected. Suppose that the original overlapping local POD bases $\boldsymbol{\Phi}^{01} \in \mathbb{R}^{M\times N_1}$ and $\boldsymbol{\Phi}^{02} \in \mathbb{R}^{M\times N_2}$ are:

$$\boldsymbol{\Phi}^{01} = \begin{bmatrix}\boldsymbol{\phi}^1\\ \boldsymbol{\phi}^{1\cap}\\ \boldsymbol{0}\end{bmatrix}, \quad \boldsymbol{\Phi}^{02} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{\phi}^{2\cap}\\ \boldsymbol{\phi}^2\end{bmatrix}, \tag{16}$$

where

$$\boldsymbol{\phi}^k \in \mathbb{R}^{M_k\times N_k}, \tag{17}$$

is the restriction of the local basis functions in Ω_k to the part of the subdomain without overlapping (M_k components), and

$$\boldsymbol{\phi}^{k\cap} \in \mathbb{R}^{M_{k\cap}\times N_k}, \tag{18}$$

corresponds to the restriction of $\boldsymbol{\Phi}^{0k}$ to the overlapping domain $\Omega_\cap$ ($M_{k\cap}$ components). Note that $M_\cap := M_{1\cap} = M_{2\cap}$, and now $M = M_1 + M_2 + M_\cap$.

In this case the corrected bases are:

$$\boldsymbol{\Phi}^1 = \begin{bmatrix}\boldsymbol{\phi}^1\\ \beta\boldsymbol{\phi}^{1\cap}\\ \boldsymbol{0}\end{bmatrix}, \quad \boldsymbol{\Phi}^2 = \begin{bmatrix}\boldsymbol{0}\\ (1-\beta)\boldsymbol{\phi}^{2\cap}\\ \boldsymbol{\phi}^2\end{bmatrix}, \tag{19}$$

where $\beta \in [0,1]$ is a weighting parameter. Note that the limits $\beta = 0$ and $\beta = 1$ correspond to the non overlapping case. If several subdomains overlap in a certain region, then each subdomain is assigned a weighting parameter β^k and we must ensure that $\sum\beta^k = 1$. The motivation for this correction is the requirement that the resulting global reduced basis (obtained as the union of the local bases for each subdomain) is capable of representing the global snapshot set if N is equal to the number of snapshots. This is shown in Fig. 2, where some illustrative basis functions for a one-dimensional problem are depicted.

Supposing that the problem defined in (1) allows to do so, the stabilization penalty term imposes that the solution at the overlapping region recovered from $\boldsymbol{\phi}^{1\cap}$ (prior to the introduction of the weighting parameter β) is equal to the solution recovered from $\boldsymbol{\phi}^{2\cap}$:

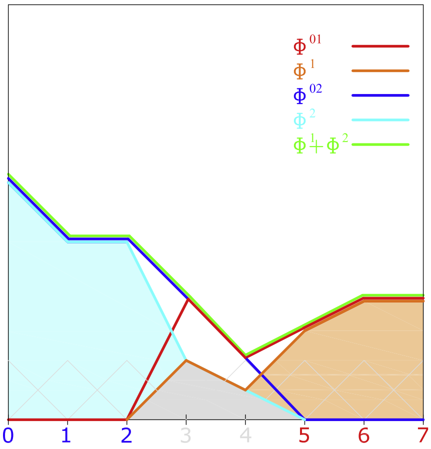

Fig. 2. Overlapping local basis functions. $\beta = 0.5$. The overlapping nodes are depicted in gray. In this particular case the value of the basis functions at the overlapping nodes coincides, which is not necessarily the case for L-POD.

$$\boldsymbol{\phi}^{1\cap}\boldsymbol{\alpha}^1 = \boldsymbol{U}^{1\cap} = \boldsymbol{U}^{2\cap} = \boldsymbol{\phi}^{2\cap}\boldsymbol{\alpha}^2 \in \mathbb{R}^{M\cap}, \tag{20}$$

where now we take $\boldsymbol{\alpha}^k \in \mathbb{R}^{N_k}$ the ROM degrees of freedom for each subdomain, which, using the terminology of the LG-POD approach, would in fact be $\boldsymbol{\alpha}^k|_k$.

This condition can be equivalently written as:

$$\begin{aligned}
&(\boldsymbol{\Phi}^{1\cap})^T\boldsymbol{\Phi}^{1\cap}\boldsymbol{\alpha}^1 - (\boldsymbol{\Phi}^{1\cap})^T\boldsymbol{\Phi}^{2\cap}\boldsymbol{\alpha}^2 = \boldsymbol{0},\\
&(\boldsymbol{\Phi}^{2\cap})^T\boldsymbol{\Phi}^{1\cap}\boldsymbol{\alpha}^1 - (\boldsymbol{\Phi}^{2\cap})^T\boldsymbol{\Phi}^{2\cap}\boldsymbol{\alpha}^2 = \boldsymbol{0},
\end{aligned} \tag{21}$$

where

$$\boldsymbol{\Phi}^{k\cap} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\phi}^{k\cap} \\ \boldsymbol{0} \end{bmatrix}. \tag{22}$$

Introducing (21) as a penalized constraint in the ROM system we get:

$$\boldsymbol{a}_{11}\boldsymbol{\alpha}^1 + \boldsymbol{a}_{12}\boldsymbol{\alpha}^2 + \frac{1}{\varepsilon}(\boldsymbol{M}_{11}\boldsymbol{\alpha}^1 - \boldsymbol{M}_{12}\boldsymbol{\alpha}^2) = \boldsymbol{f}_{\ 1}, \tag{23}$$

$$\boldsymbol{a}_{21}\boldsymbol{\alpha}^1 + \boldsymbol{a}_{22}\boldsymbol{\alpha}^2 + \frac{1}{\varepsilon}(\boldsymbol{M}_{21}\boldsymbol{\alpha}^1 - \boldsymbol{M}_{22}\boldsymbol{\alpha}^2) = \boldsymbol{f}_{\ 2}, \tag{24}$$

where

$$\boldsymbol{M}_{kl} = (\boldsymbol{\Phi}^{k\cap})^T\boldsymbol{\Phi}^{l\cap} \in \mathbb{R}^{N_k \times N_l}. \tag{25}$$

Note that if this strategy is applied to the LG-POD it is automatically satisfied that:

$$\boldsymbol{M}_{11} = \boldsymbol{M}_{12} = \boldsymbol{M}_{21} = \boldsymbol{M}_{22}. \tag{26}$$

An important property of the block diagonal penalty matrices $\boldsymbol{M}_{kk}$ is that they can only be guaranteed be full-rank matrices if $\Omega_\cap = \Omega$; in this case $\boldsymbol{M}_{kk} = \boldsymbol{I}$ for LG-POD. However, this stabilization strategy shows good results in the numerical examples even if $\Omega_\cap \neq \Omega$, both for L-POD and LG-POD, to which this idea can also be applied.

4.3. Ghost overlapping penalization

As explained in the previous section, it is possible to take the limit values for β, $\beta = 0$ or $\beta = 1$. This corresponds to the non-overlapping case. However, we can still keep the penalty terms $\boldsymbol{M}_{kl}$ as stabilization terms. We call this a *ghost overlapping penalization.* As it will be shown in the numerical examples section, this is also an effective strategy to stabilize the partitioned reduced order model.

Another possibility consists in using different β weighting parameters depending on which is the subdomain against whose local basis functions we are testing the equations. This results in a system of the form:

$$\begin{aligned}
&(\boldsymbol{\Phi}^1_{\beta^1})^T\boldsymbol{A}\boldsymbol{\Phi}^1_{\beta^1}\boldsymbol{\alpha}^1 + (\boldsymbol{\Phi}^2_{\beta^1})^T\boldsymbol{A}\boldsymbol{\Phi}^2_{\beta^1}\boldsymbol{\alpha}^2 + \frac{1}{\varepsilon}(\boldsymbol{M}_{11}\boldsymbol{\alpha}^1 - \boldsymbol{M}_{12}\boldsymbol{\alpha}^2) = \boldsymbol{\Phi}^1_{\beta^1}\boldsymbol{F},\\
&(\boldsymbol{\Phi}^2_{\beta^2})^T\boldsymbol{A}\boldsymbol{\Phi}^1_{\beta^2}\boldsymbol{\alpha}^1 + (\boldsymbol{\Phi}^2_{\beta^2})^T\boldsymbol{A}\boldsymbol{\Phi}^2_{\beta^2}\boldsymbol{\alpha}^2 + \frac{1}{\varepsilon}(\boldsymbol{M}_{21}\boldsymbol{\alpha}^1 - \boldsymbol{M}_{22}\boldsymbol{\alpha}^2) = \boldsymbol{\Phi}^2_{\beta^2}\boldsymbol{F},
\end{aligned} \tag{27}$$

where

$$\boldsymbol{\Phi}^1_{\beta^k} = \begin{bmatrix} \boldsymbol{\phi}^1 \\ \beta^k \boldsymbol{\phi}^{1\cap} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\Phi}^2_{\beta^k} = \begin{bmatrix} \mathbf{0} \\ (1-\beta^k)\boldsymbol{\phi}^{2\cap} \\ \boldsymbol{\phi}^2 \end{bmatrix}, \tag{28}$$

This possibility is particularly useful in the case of the mixed FOM–ROM approach described in Section 5, where we take $\beta^1 = 1$ and $\beta^2 = 0$. Note that the penalty matrices $\boldsymbol{M}_{kl}$ are independent of the weighting coefficients β^k.

In the same way, different penalty parameters ε^1 and ε^2 can be used for the equations corresponding to different subdomains. Again, in the hybrid FOM–ROM approach described in Section 5, $\varepsilon^{FOM} \neq \varepsilon^{ROM}$, particularly $\frac{1}{\varepsilon^{FOM}} = 0$.

A more general possibility which has not been studied here is considering the overlapping region for writing the reduced-order equations to be independent from the definition of the region for enforcing the penalty equations.

5. Full-order/reduced-order domain decomposition (FOM–ROM)

Another possibility is the use of a hybrid full-order/reduced-order (FOM–ROM) approach. This is convenient if a high fidelity model is required in a certain region of the domain, or if the conditions in a certain region strongly depart from the conditions at which the snapshots for building the POD bases were taken. In this cases one can choose to solve the FOM problem in one of the subdomains, while keeping the cheaper ROM approach in the less critical subdomains. Extending the described partitioned ROM strategy to a hybrid FOM–ROM domain decomposition method is straightforward: the FOM–ROM is obtained by taking as local basis for the FOM subdomain Ω_F the nodal shape functions of the finite element space for the unknown. In the ROM subdomain Ω_R a local reduced basis needs to be built, which can be either LG-POD or L-POD. The hybrid FOM–ROM system without overlapping is:

$$\begin{bmatrix} \boldsymbol{A}|_{FF} & \boldsymbol{A}|_{FR}\boldsymbol{\phi}^R \\ (\boldsymbol{\phi}^R)^T\boldsymbol{A}|_{RF} & (\boldsymbol{\phi}^R)^T\boldsymbol{A}|_{RR}\boldsymbol{\phi}^R \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^F \\ \boldsymbol{\alpha}^R \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}|_F \\ (\boldsymbol{\phi}^R)^T\boldsymbol{F}|_R \end{bmatrix}. \tag{29}$$

Let us remark that the time stepping strategies need not to be the same for the full order and the reduced order equations. For instance, if an explicit reduced order model is used for the incompressible Navier–Stokes equations [7], the $\boldsymbol{A}$ matrix and the $\boldsymbol{F}$ RHS vector for the reduced order equations are taken from the explicit model, while the equations arising from the implicit time stepping are kept for the full order equations:

$$\begin{bmatrix} \boldsymbol{A}|_{FF} & \boldsymbol{A}|_{FR}\boldsymbol{\phi}^R \\ (\boldsymbol{\phi}^R)^T\boldsymbol{A}^{\exp}|_{RF} & (\boldsymbol{\phi}^R)^T\boldsymbol{A}^{\exp}|_{RR}\boldsymbol{\phi}^R \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^F \\ \boldsymbol{\alpha}^R \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}|_F \\ (\boldsymbol{\phi}^R)^T\boldsymbol{F}^{\exp}|_R \end{bmatrix}. \tag{30}$$

If a Petrov–Galerkin projection is used, this can also be introduced in the ROM equations. For instance, the FOM–ROM system for the Petrov–Galerkin projection described in [11,13] would result in the following system:

$$\begin{bmatrix} \boldsymbol{A}|_{FF} & \boldsymbol{A}|_{FR}\boldsymbol{\phi}^R \\ \boldsymbol{A}^{PG}_{RF} & \boldsymbol{A}^{PG}_{RR}\boldsymbol{\phi}^R \end{bmatrix} \begin{bmatrix} \boldsymbol{U}^F \\ \boldsymbol{\alpha}^R \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}|_F \\ \boldsymbol{F}^{PG}_R \end{bmatrix}, \tag{31}$$

where

$$\begin{aligned} \boldsymbol{A}^{PG}_{RF} &= (\boldsymbol{\phi}^R)^T\left(\boldsymbol{A}|^T_{FR}\boldsymbol{A}|_{FF} + \boldsymbol{A}|^T_{RR}\boldsymbol{A}|_{RF}\right), \\ \boldsymbol{A}^{PG}_{RR} &= (\boldsymbol{\phi}^R)^T\left(\boldsymbol{A}|^T_{FR}\boldsymbol{A}|_{FR} + \boldsymbol{A}|^T_{RR}\boldsymbol{A}|_{RR}\right), \\ \boldsymbol{F}^{PG}_R &= (\boldsymbol{\phi}^R)^T\boldsymbol{A}|^T_{FR}\boldsymbol{F}|_F + (\boldsymbol{\phi}^R)^T\boldsymbol{A}|^T_{RR}\boldsymbol{F}|_R. \end{aligned}$$

Obviously, any hyper-reduction technique for efficiently reconstructing the ROM equations can be used. The described overlapping strategies and the use weighting coefficients need to be introduced in the previous formulation. This can be done in a straightforward manner, including the use of different weighting parameters β or ε for the FOM and the ROM equations described in (27).

6. Application to the finite element approximation of the Navier–Stokes equations

6.1. Incompressible Navier–Stokes equations

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:

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

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 conditions have to be appended to this problem.

Let now $V = H^1(\Omega)^d$, and $V_0 = \{\boldsymbol{v} \in V | \boldsymbol{v} = \boldsymbol{0} \, \text{on} \Gamma_D\}$. Let also $Q = L^2(\Omega)$ 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:

$$(\boldsymbol{v}, \partial_t \boldsymbol{u}) + B([\boldsymbol{v}, q], [\boldsymbol{u}, p]) = \langle \boldsymbol{v}, \boldsymbol{f} \rangle \quad \forall [\boldsymbol{v}, q] \in V_0 \times Q \tag{32}$$

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 below, $(\cdot,\cdot)$ denotes the L^2 product in Ω and $\langle f, g\rangle$ denotes the integral of the product of the functions f and g in Ω, whereas $\langle\cdot,\cdot\rangle_\omega$ is used for the integral restricted to a subdomain ω.

6.2. 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 (32) is small the Galerkin method fails and stabilized finite element methods need to be used. On the other hand, the velocity and pressure spaces $V_{h,0}$ and Q_h need to fulfill an inf–sup condition in order for the discrete version of the problem defined in (32) to be well-posed. 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 (32) and on the other, allow to circumvent the inf–sup condition and to freely choose the interpolation spaces V_h and Q_h.

The stabilized formulations we use are derived from a multiscale splitting of the spaces for the unknowns *bfu* and p into the finite element part of the solution and the subscales (see [23], 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: for each t, find $\boldsymbol{u}_h(t) \in V_h, p_h(t) \in Q_h$ such that:

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h) + B([\boldsymbol{v}_h, q_h], [\boldsymbol{u}_h, p_h]) + \sum_K \tau_K \langle \boldsymbol{u}_h\cdot\nabla\boldsymbol{v}_h + \nu\Delta\boldsymbol{v}_h + \nabla q_h, \boldsymbol{r}([\boldsymbol{u}_h, p_h])\rangle_K = \langle \boldsymbol{v}_h, \boldsymbol{f}\rangle \tag{33}$$

for all $\boldsymbol{v}_h \in V_{h,0}, q_h \in Q_h$. Boundary conditions need to be appended to this problem. In (33):

$$\boldsymbol{r}([\boldsymbol{u}_h, p_h]) = \partial_t \boldsymbol{u}_h - \nu\Delta\boldsymbol{u}_h + \boldsymbol{u}_h\cdot\nabla\boldsymbol{u}_h + \nabla p_h - \boldsymbol{f} \tag{34}$$

is the residual of the momentum equation and τ_K is the stabilization parameter:

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

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 [16]. As explained in [17], a more accurate method can be obtained replacing $\boldsymbol{r}([\boldsymbol{u}_h, p_h])$ in (33) by its projection orthogonal to V_h. A discretization scheme for approximating the time derivatives needs to be added to the formulation (33). An iterative scheme for solving the non-linear nature of (33) is also required. For a given time step, the method we favor for dealing with the non-linearity is Picard's method: despite having a slower convergence rate than Newton's method, the convergence radius of Picard's method is larger for the incompressible Navier–Stokes equations. After discretizing in time and introducing the non-linear iterative scheme, (33) can be written in matrix form as:

$$\boldsymbol{A}(\boldsymbol{U}_{n+1}^{i-1})\boldsymbol{U}_{n+1}^{i} = \boldsymbol{F}(\boldsymbol{U}_n, \boldsymbol{U}_{n-1}), \tag{35}$$

where the superscripts indicate iteration counter and the subscripts indicate the time step. (35) is the counterpart of (1) for the incompressible flow equations identifying $\boldsymbol{U} = [\boldsymbol{u}, p]$. Once the system (35) is obtained the reduced order and domain decomposition strategies described in Sections 2,2,3,4,5 can be used to solve a reduced problem for the incompressible Navier–Stokes equations.

6.3. Domain decomposition ROM strategy

The use of the domain decomposition ROM strategy to the particular problem of the incompressible Navier–Stokes equations is straightforward if a ROM approach is used in all the subdomains. On the other hand, some care needs to be taken when a FOM approximation is used in one of the subdomains while a ROM approximation is used in its neighbor subdomains. As in the original domain decomposition strategy, a penalization term through overlapping is convenient in this FOM–ROM approach. However, it is necessary to distinguish between the velocity and the pressure unknowns of the incompressible Navier–Stokes equations in this case: only the equality between the FOM and the ROM velocities in the overlapping region is imposed, and no condition is required on the FOM pressure field. This is so because the pressure field can be understood as the Lagrange multiplier enforcing the incompressibility constraint, and as such it is not possible to enforce the pressure value over the overlapping domain. Taking this into account we may rewrite system (29) in this particular case by separating the velocity and pressure unknowns in the FOM domain, still without considering overlapping and the imposition of continuity:

$$\begin{bmatrix} \boldsymbol{A}|_{FF_{uu}} & \boldsymbol{A}|_{FF_{up}} & \boldsymbol{A}|_{FR_u}\boldsymbol{\phi}^R \\ \boldsymbol{A}|_{FF_{pu}} & \boldsymbol{A}|_{FF_{pp}} & \boldsymbol{A}|_{FR_p}\boldsymbol{\phi}^R \\ (\boldsymbol{\phi}^R)^T\boldsymbol{A}|_{RF_u} & (\boldsymbol{\phi}^R)^T\boldsymbol{A}|_{RF_p} & (\boldsymbol{\phi}^R)^T\boldsymbol{A}|_{RR}\boldsymbol{\phi}^R \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^F \\ \boldsymbol{p}^F \\ \boldsymbol{\alpha}^R \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}|_{F_u} \\ \boldsymbol{F}|_{F_p} \\ (\boldsymbol{\phi}^R)^T\boldsymbol{F}|_R \end{bmatrix}. \tag{36}$$

Supposing that an overlapping region exists, we can enforce the equivalence between the FOM and the ROM velocities in the overlapping domain. We denote by $\boldsymbol{\phi}_u^R$ the velocity components of the local basis functions for the nodes interior to the ROM domain and by $\boldsymbol{\phi}_p^R$ the pressure components:

$$\boldsymbol{\phi}^R = \begin{bmatrix} \boldsymbol{\phi}_u^R \\ \boldsymbol{\phi}_p^R \end{bmatrix} \in \mathbb{R}^{M_R \times N_R}. \tag{37}$$

The same decomposition can be introduced for the overlapping region $\Omega_\cap$:

$$\boldsymbol{\phi}^{R\cap} = \begin{bmatrix} \boldsymbol{\phi}_u^{R\cap} \\ \boldsymbol{\phi}_p^{R\cap} \end{bmatrix} \in \mathbb{R}^{M_\cap \times N_R}. \tag{38}$$

Velocity and pressure FOM unknowns need also to be decomposed into unknowns in the overlapping region and the rest of the FOM domain:

$$\begin{bmatrix} \boldsymbol{u}^F \\ \boldsymbol{p}^F \\ \boldsymbol{u}^{F\cap} \\ \boldsymbol{p}^{F\cap} \end{bmatrix} \in \mathbb{R}^{M_F + M_\cap}, \tag{39}$$

where:

$$\boldsymbol{u}^F \in \mathbb{R}^{M_{F_u}}, \quad \boldsymbol{p}^F \in \mathbb{R}^{M_{F_p}}, \quad \boldsymbol{u}^{F\cap} \in \mathbb{R}^{M_{\cap u}}, \quad \boldsymbol{p}^{F\cap} \in \mathbb{R}^{M_{\cap p}}.$$

M_{F_u} is the dimension of the velocity components in the interior of the FOM domain, M_{F_p} the dimension of the pressure components and $M_F = M_{F_u} + M_{F_p}$. Analogously, $M_{\cap_u}$ is the dimension of the velocity components in the overlapping domain, $M_{\cap_p}$ the dimension of the pressure components and $M_\cap = M_{\cap_u} + M_{\cap_p}$. We can also define $M_R = M_{R_u} + M_{R_p}$. Finally, $M = M_F + M_\cap + M_R$.

The stabilization term to be (adequately) added to the left-hand side of system (36) defined on the overlapping region is:

$$\frac{1}{\varepsilon}\begin{bmatrix} \boldsymbol{M}_{F\cap_u F\cap_u} & -\boldsymbol{M}_{F\cap_u R\cap_u} \\ -\boldsymbol{M}_{R\cap_u F\cap_u} & \boldsymbol{M}_{R\cap_u R\cap_u} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^\cap \\ \boldsymbol{\alpha}^R \end{bmatrix}, \tag{40}$$

where:

$$\boldsymbol{M}_{F\cap_u F\cap_u} = (\boldsymbol{\phi}_u^{F\cap})^T \boldsymbol{\phi}_u^{F\cap} \in \mathbb{R}^{M_{\cap u} \times M_{\cap u}}, \tag{41}$$

$$\boldsymbol{M}_{F\cap_u R\cap_u} = (\boldsymbol{\phi}_u^{F\cap})^T \boldsymbol{\phi}_u^{R\cap} \in \mathbb{R}^{M_{\cap u} \times N_R}, \tag{42}$$

$$\boldsymbol{M}_{R\cap_u F\cap_u} = (\boldsymbol{\phi}_u^{R\cap})^T \boldsymbol{\phi}_u^{F\cap} \in \mathbb{R}^{N_R \times M_{\cap u}}, \tag{43}$$

$$\boldsymbol{M}_{R\cap_u R\cap_u} = (\boldsymbol{\phi}_u^{R\cap})^T \boldsymbol{\phi}_u^{R\cap} \in \mathbb{R}^{N_R \times N_R}, \tag{44}$$

where the basis functions for the FOM velocity $\boldsymbol{\phi}_u^{F\cap}$ are the finite element nodal velocity shape functions in the overlapping region. Regarding the overlapping strategy, a specific approach for the FOM–ROM case is devised. Its particularity consists in using different coefficients β_F and $\beta_R = 1 - \beta_F$ for the FOM and the ROM equations: in the FOM equations $\beta_F = 1$ is taken, while $\beta_R = 1$ is used for the ROM equations. When updating the unknown values after solving the FOM–ROM system, we take the FOM values in the overlapping area, because these are expected to be more accurate than the corresponding ROM values.

Let us now introduce the notation:

$$\boldsymbol{a}_{F_uR_p} = \boldsymbol{A}|_{F_uR_p},$$

$$\boldsymbol{a}_{F\cap_u R} = \boldsymbol{A}|_{\cap_u R}\boldsymbol{\phi}^R,$$

$$\boldsymbol{a}_{(R\cap)F_u} = (\boldsymbol{\phi}^{R\cap})^T\boldsymbol{A}|_{\cap F_u},$$

$$\boldsymbol{a}_{(R+\cap)(R+\cap)} = \begin{bmatrix}\boldsymbol{\phi}^R \\ \boldsymbol{\phi}^{R\cap}\end{bmatrix}^T \boldsymbol{A}|_{(R+\cap)(R+\cap)}\begin{bmatrix}\boldsymbol{\phi}^R \\ \boldsymbol{\phi}^{R\cap}\end{bmatrix},$$

$$\boldsymbol{f}_{\ F_u} = \boldsymbol{F}|_{F_u},$$

$$\boldsymbol{f}_{\ R} = \begin{bmatrix}\boldsymbol{\phi}^R \\ \boldsymbol{\phi}^{R\cap}\end{bmatrix}^T \boldsymbol{F}|_{(R+\cap)}.$$

Note that the domains where the various matrices and vectors are defined differ depending on whether we are considering the FOM or the ROM equations. This is illustrated in Fig. 3, where the overlapping, FOM and ROM domains are depicted for the construction of the penalty matrices, the FOM equations and the ROM equations.

Introducing (40) in (36) the final system to be solved taking into account overlapping is obtained. Defining:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{a}_{F_uF_u} & \boldsymbol{a}_{F_uF_p} & \boldsymbol{a}_{F_uF\cap_u} & \boldsymbol{a}_{F_uF\cap_p} & \boldsymbol{0} \\ \boldsymbol{a}_{F_pF_u} & \boldsymbol{a}_{F_pF_p} & \boldsymbol{a}_{F_pF\cap_u} & \boldsymbol{a}_{F_pF\cap_p} & \boldsymbol{0} \\ \boldsymbol{a}_{F\cap_uF_u} & \boldsymbol{a}_{F\cap_uF_p} & \boldsymbol{a}_{F\cap_uF\cap_u} + \frac{1}{\varepsilon}\boldsymbol{M}_{F\cap_uF\cap_u} & \boldsymbol{a}_{F\cap_uF\cap_p} & \boldsymbol{a}_{F\cap_uR} - \frac{1}{\varepsilon}\boldsymbol{M}_{F\cap_uR\cap_u} \\ \boldsymbol{a}_{F\cap_pF_u} & \boldsymbol{a}_{F\cap_pF_p} & \boldsymbol{a}_{F\cap_pF\cap_u} & \boldsymbol{a}_{F\cap_pF\cap_p} & \boldsymbol{a}_{F\cap_pR} \\ \boldsymbol{a}_{(R\cap)F_u} & \boldsymbol{a}_{(R\cap)F_p} & -\frac{1}{\varepsilon}\boldsymbol{M}_{R\cap_uF\cap_u} & \boldsymbol{0} & \boldsymbol{a}_{(R+\cap)(R+\cap)} + \frac{1}{\varepsilon}\boldsymbol{M}_{R\cap_uR\cap_u} \end{bmatrix}, \quad \tilde{\boldsymbol{U}} = \begin{bmatrix} \boldsymbol{u}^F \\ \boldsymbol{p}^F \\ \boldsymbol{u}^{F\cap} \\ \boldsymbol{p}^{F\cap} \\ \boldsymbol{\alpha}^R \end{bmatrix}, \quad \tilde{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{f}_{\ F_u} \\ \boldsymbol{f}_{\ F_p} \\ \boldsymbol{f}_{\ F\cap_u} \\ \boldsymbol{f}_{\ F\cap_p} \\ \boldsymbol{f}_{\ R} \end{bmatrix}, \qquad (45)$$

the final system is:

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{F}}. \qquad (46)$$

After solving the FOM–ROM system, the FOM domain unknowns are used in order to update the global unknowns in $\Omega_\cap$.

7. Numerical examples

In this section we present some numerical examples which illustrate the behavior of the domain decomposition approaches described in this paper.

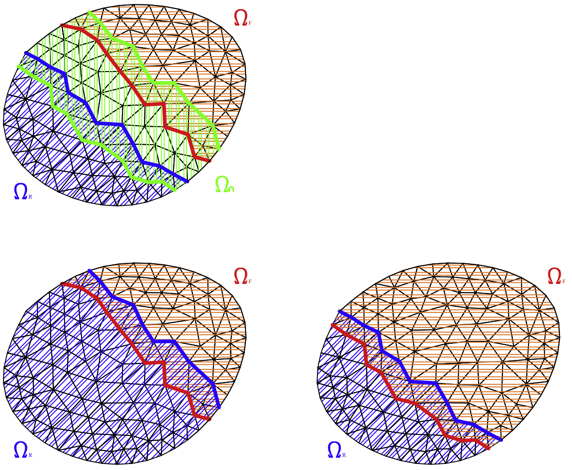

Fig. 3. Red: Ω_F. Blue: Ω_R. Green: $\Omega_\cap$. Top: penalization matrices are defined over the $\Omega_\cap$ domain. Bottom left: partitioning for the ROM equations. Bottom right: partitioning for the FOM equations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.1. Low Reynolds flow past a cylinder

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 which allow the flow to move in the direction parallel to the walls are set at $y = 0$ and $y = 8$, and the velocity is set to $\mathbf{0}$ at the cylinder surface. The viscosity has been set to $\nu = 1$, which yields a Reynolds number 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.

The domain has been partitioned into 4 subdomains, as illustrated in Fig. 4, where some LG-POD basis functions are depicted. $\gamma = 0.01$ in Eq. (15) has been used. Regarding the ROM, 50 velocity–pressure snapshots have been taken equally spaced during the FOM simulation, and the 10 first reduced basis functions have been kept for the reduced model. After the partition into subdomains, the total number of basis functions is 40. The hyper-reduction technique for rebuilding the ROM system described in [7] is applied to reduce the computational cost.

Figs. 5 and 6 compare the performance of the DD-ROM with the FOM model; it can be observed that results for the DD-ROM almost exactly match the full-order results: the relative L^2-error for the velocity and pressure time history in the last oscillation period of the simulation in Figs. 5 and 6 are 2.7% (velocity) and 0.43% (pressure) for the LG-POD and 2.3% (velocity) and 0.38% (pressure) for the L-POD. Fig. 7 shows a comparison of the behavior of L-POD model with overlapping penalization (considering 2 layers of elements as overlapping) and L-POD ghost overlapping penalization (L-POD-GHOST), where now a second order backward differences scheme has been used for the time integration. It can be observed that both stabilization approaches work very similarly, the relative L^2 error for the L-POD-GHOST being now 2.4% for the velocity and 0.41% for the pressure.

The time for the DD-ROM computations is 0.65 s, which is approximately 1% of the computational cost of the FOM model, which takes 65.9 s to run. The ROM without domain partitioning (10 global basis functions) takes 0.39 s. The cost of the DD-ROM is larger, the cause for this being that the dimension of the resulting reduced order system is 4 times larger.

7.2. Three-dimensional flow past two cylinders

In this section we simulate the three-dimensional flow past two cylinders. The domain simulation is a rectangular prism of size $16 \times 8 \times 4$. Two unit diameter cylinders are considered with axes at $(4, 4, z)$ and $(8, 4, z)$. The income velocity modulus is 1, slip boundary conditions are applied at the lateral walls and zero velocity is imposed on 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 $\delta t = 0.1$, the simulation is run for 200 time steps, a backward Euler integration scheme is used. A 248460 linear tetrahedra finite element mesh is used.

The domain has been partitioned into 4 subdomains, which result from dividing the domain in two halves in the *xz* and *yz* planes. In each of the subdomains, 10 basis functions (obtained from a total of 100 snapshots taken equally spaced in time during the FOM simulation) have been used for the reduced-order domain-decomposition simulation. $\gamma = 0.01$ in Eq. (15) has been used. Again, the hyper-reduction technique for rebuilding the ROM system described in [7] is applied to reduce the computational cost. The computational cost of the DD-ROM computations is 47.9 s, approximately 1.15% of the full-order model computational cost, which is 4174.9 s.

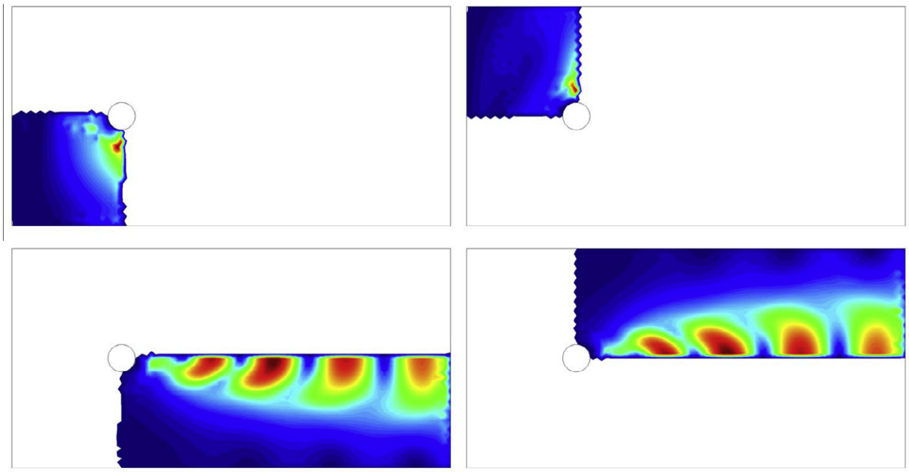

Fig. 4. Support of the local basis functions for the flow past a cylinder.

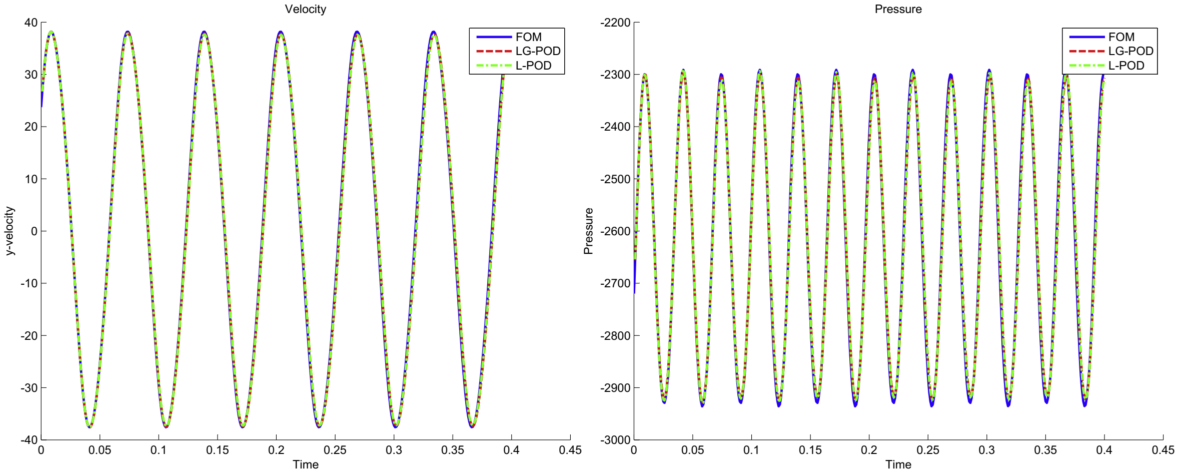

Fig. 5. Flow past a cylinder. Comparison of FOM and DD-ROM results at coordinates $(6.5, 4)$. Left, vertical velocity time history. Right, pressure time history.

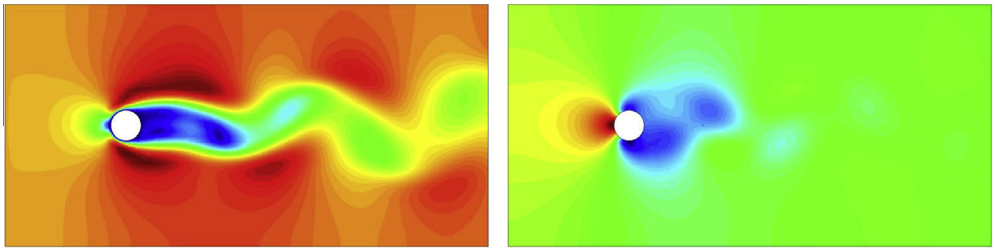

Fig. 6. Flow past a cylinder. Velocity (left) and pressure (right) fields at the end of the DD-ROM simulation.

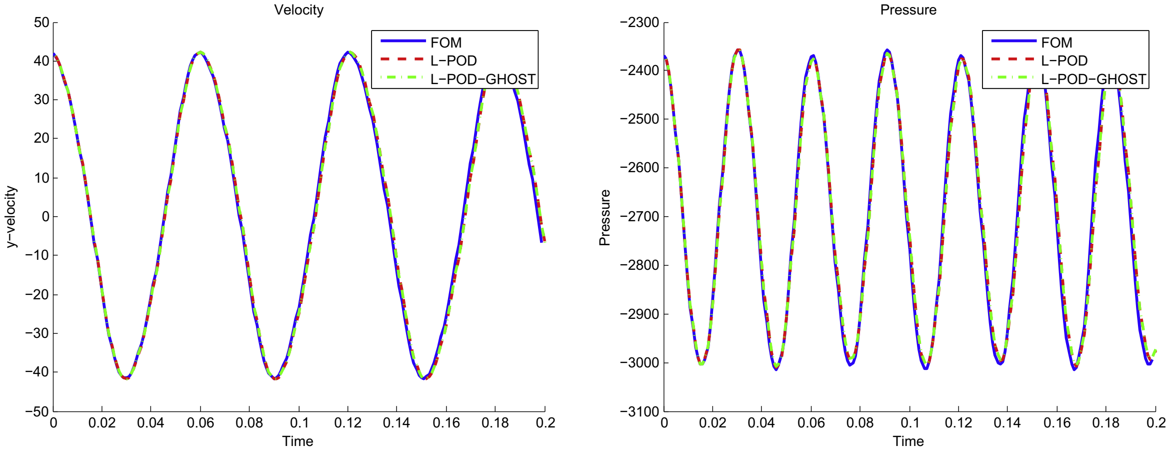

Fig. 7. Flow past a cylinder. Comparison of FOM and L-POD with overlapping penalization and L-POD with ghost-overlapping penalization results at coordinates $(6.5, 4)$. Left, vertical velocity time history. Right, pressure time history.

Fig. 8 compares the velocity and pressure fields for the full-order and the domain-decomposition reduced-order model at the end of the simulation. It can be appreciated that the DD-ROM model is capable of successfully representing the solution of the full-order model, the difference between both models are difficult to appreciate. Note that as expected the transition between subdomains is smooth and cannot be appreciated in the numerical solution.

Fig. 9 shows a comparison of the velocity in the y direction and the pressure at a point behind the first strut with coordinates $(6, 4, 0)$. Even if the point is ubicated at the frontier between two subdomains, the velocity and pressure history are recovered with a good accuracy: 1.7% relative error for the velocity history in the last oscillation period and 0.7% relative error for the pressure time history in the last oscillation period.

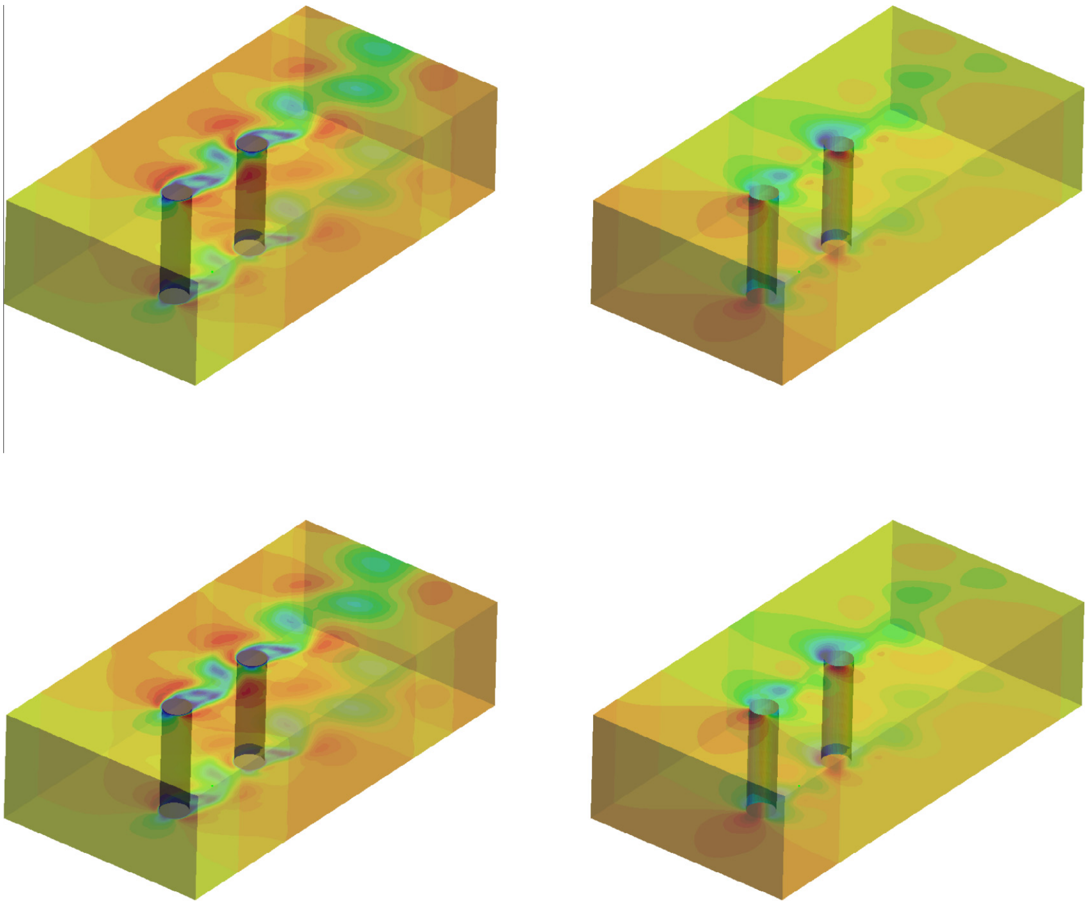

Fig. 8. Three-dimensional flow past two cylinders. Velocity (left) and pressure (right) fields. Comparison between full-order (top) and domain-decomposition reduced-order model (bottom).

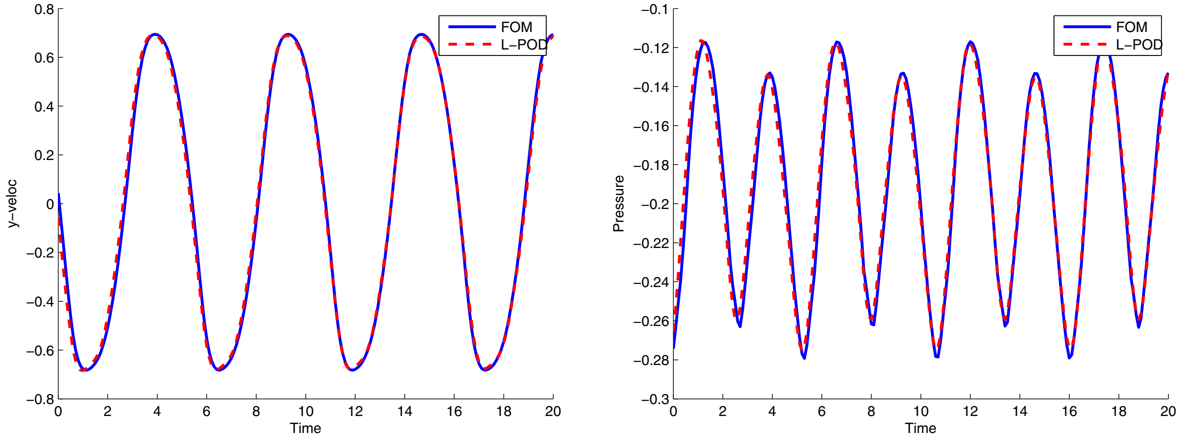

Fig. 9. Comparison of the velocity (left) and pressure (right) time evolution at a point behind the first strut $(6, 4, 0)$.

7.3. Flow injection in a rectangular cylinder

In this numerical example we show the capability of the proposed FOM–ROM strategy to adapt to flow configurations which were not present in the original snapshot set. The initial problem set is the incompressible flow past a rectangular cylinder at $\mathrm{Re} = 100$. The computational domain consists of a 24×12 domain with a square cylinder with a side of size 1. The square cylinder is centered at coordinates $(8, 6)$. The horizontal inflow velocity is set to 1. Slip boundary conditions which allow the flow to move in the direction parallel to the walls are set at $y = 0$ and $y = 12$, and velocity is set to $\mathbf{0}$ on the cylinder surface in the direction normal to the surface, a tangential force (computed by using a wall-law approach) is used to model the velocity in the tangential direction. The viscosity has been set to $\nu = 0.01$, which yields a Reynolds number $\mathrm{Re} = 100$ based on the dimension of the cylinder and the inflow velocity. A second order backward differences scheme has been used for the time integration with time step $\delta t = 0.1$. In this example, a relatively fine 67224 linear element mesh has been used to solve the problem.

An initial run of the full-order model is performed for the snapshot collection and no domain decomposition strategy is applied in the initial run. The FOM model takes 849.36 s to run. After the snapshot collection procedure, the ROM is capable of reproducing the FOM solution with a good accuracy for the velocity field (2,1% of relative error in the L^2-norm for the last oscillation period), the pressure amplitude being underpredicted (but with only with 0.8% of relative error in the last oscillation period), and a very low computational cost (3.07 s, 0.37% of the original computational cost), as illustrated in Fig. 10. For the ROM run, 10 basis functions are used, which are obtained from the POD decomposition of the original 50 snapshot collection.

As illustrated in Fig. 10, the reduced-order model is capable of reproducing the solution of the full-order model for the configuration in which the snapshots were taken. However, let us now consider the flow injection in the downstream side of the cylinder illustrated in Fig. 11, which is introduced in order to modify the flow. The velocity in the injection region (whose length is 0.2) is 0.1 in the direction normal to the cylinder surface. Fig. 12 illustrates the behavior of the reduced order model when the injection is considered. Despite its very low computational cost compared to the FOM model, it is clear that the ROM is incapable of reproducing the new flow configuration; the reason for this is that the snapshot set from which the ROM base was built does not contain the solution with the flow injection.

Let us now consider the FOM–ROM strategy described in the previous sections. We will decompose the physical domain into two subdomains, based on our a priori knowledge of the boundary conditions of the problem: the first subdomain corresponds to the region surrounding the square cylinder of the rectangle $(7,10) \times (5,7)$. In this subdomain a FOM approach is going to be taken, and the Navier–Stokes equations are going to be solved with full accuracy. The second subdomain covers the rest of the computational domain. Since this region does not involve the critical area where the vortexes are formed, it is going to be solved by means of the less accurate ROM strategy. The ROM basis are obtained from a set of 100 snapshots, from which a L-POD basis of 10 basis functions is obtained. $\gamma^{ROM} = 0.035$ in Eq. (15) has been used for the ROM equations, while $\gamma^{FOM} = 0$ has been used for the FOM equations. Also, the ghost overlapping penalization described in Section 4.3 has been used, with $\beta^{FOM} = 1$ and $\beta^{ROM} = 0$. This choice of β and γ parameters is basically saying that, in the overlapping region, we recover the final unknown as the one in the FOM domain (which is expected to yield higher accuracy). As it will be shown, the combination of both strategies (FOM and ROM) allows to recover a solution which is close to the full FOM solution, but at a much lower computational cost.

Fig. 12 shows a comparison of the vertical velocity and pressure at a point at the wake of the cylinder with coordinates $(5,4)$, for the FOM, the ROM and the FOM–ROM models. It is interesting to note that the ROM model is not able to capture the physics of the problem; this is natural since the ROM basis does not contain the solution of the injection case. The FOM–ROM model, on the other hand, is capable of a quite accurate solution of the system evolution in the short term in the FOM domain (13.3% relative error for the velocity time history in the last oscillation period and 4.6% error in the pressure). Fig. 13 compares the velocity and pressure fields of the FOM and the FOM–ROM models. We can observe that in the region surrounding the cylinder (FOM region) the velocity and pressure fields are very similar, in the ROM region the velocity fields slightly differ, with more intense vortexes or bulbs in the FOM simulation. This is due to the difficulties for the ROM model for representing the injected velocity and pressure fields (the used snapshots are *bad*for the injection case). Despite this evident lack

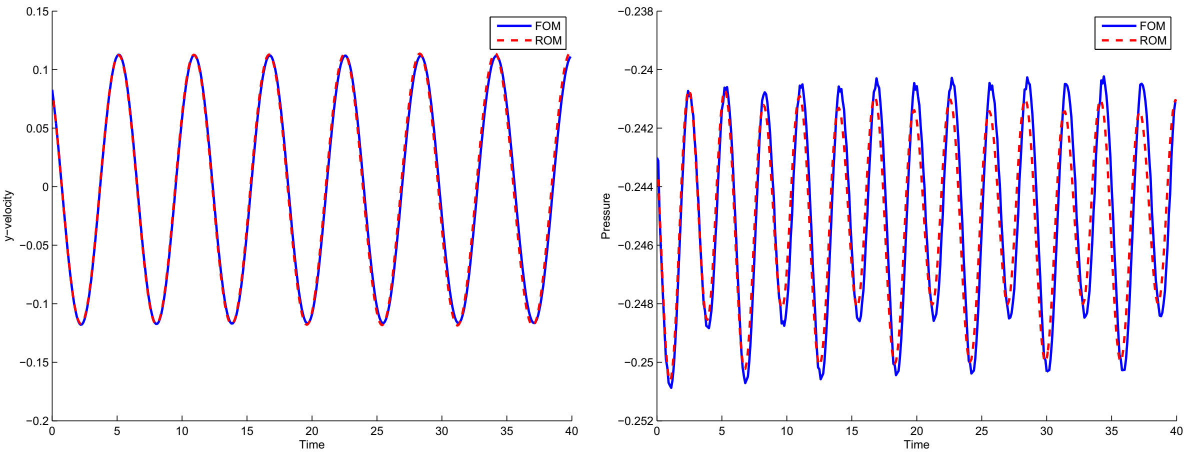

Fig. 10. Comparison of the FOM and ROM velocities at $(5,4)$ for the initial configuration.

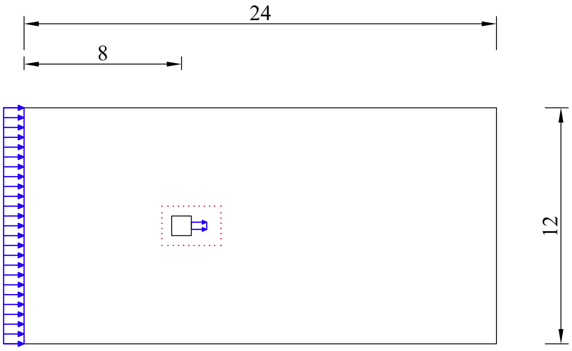

Fig. 11. Flow injection configuration. The red dotted line denotes the FOM domain for the FOM–ROM model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

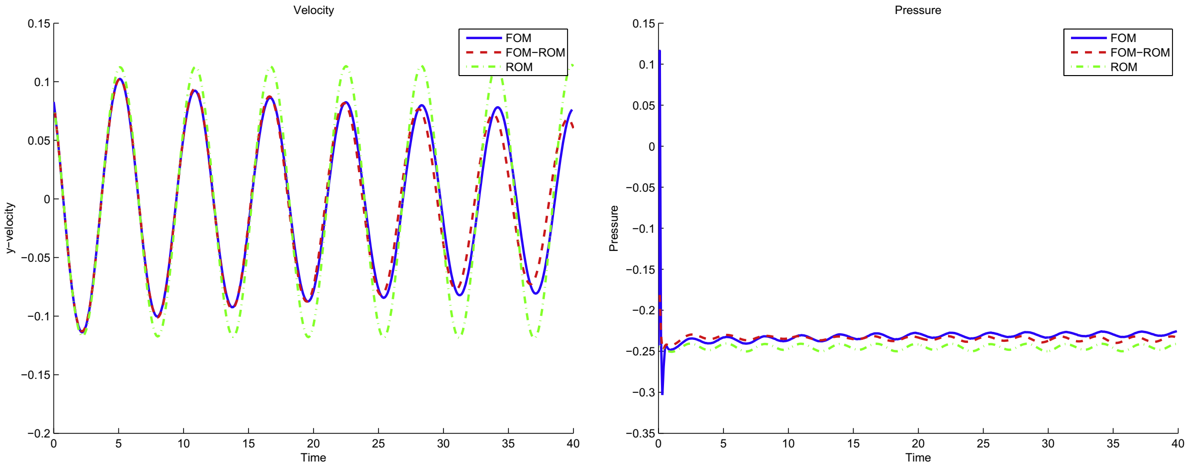

Fig. 12. Comparison of the vertical velocity (left) and pressure (right) at $(5,4)$ for the FOM, FOM–ROM and ROM models for the injection case.

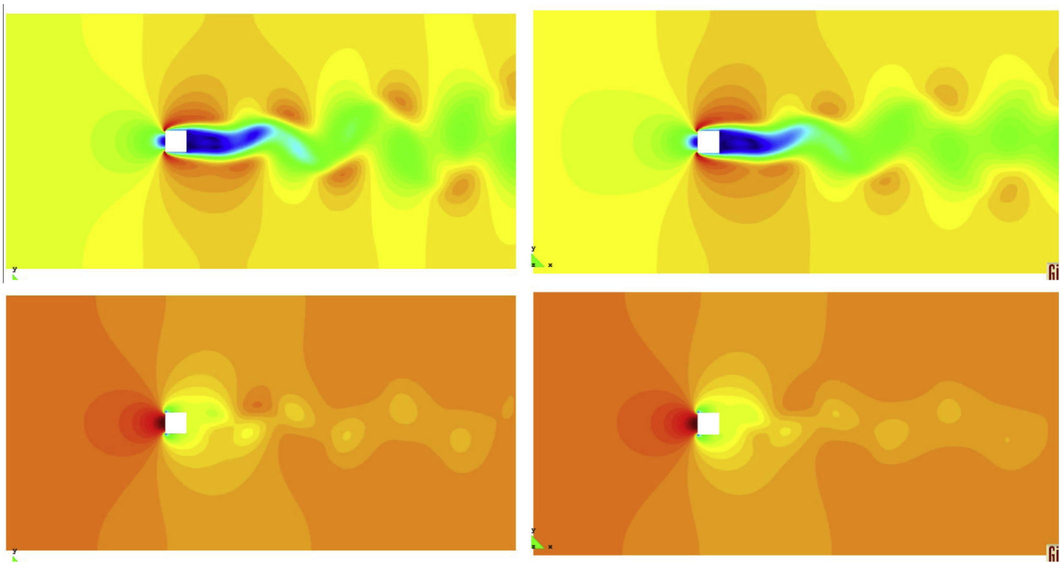

Fig. 13. Comparison of the velocity (top) and pressure (bottom) fields after 400 steps. Left: FOM. Right: FOM–ROM.

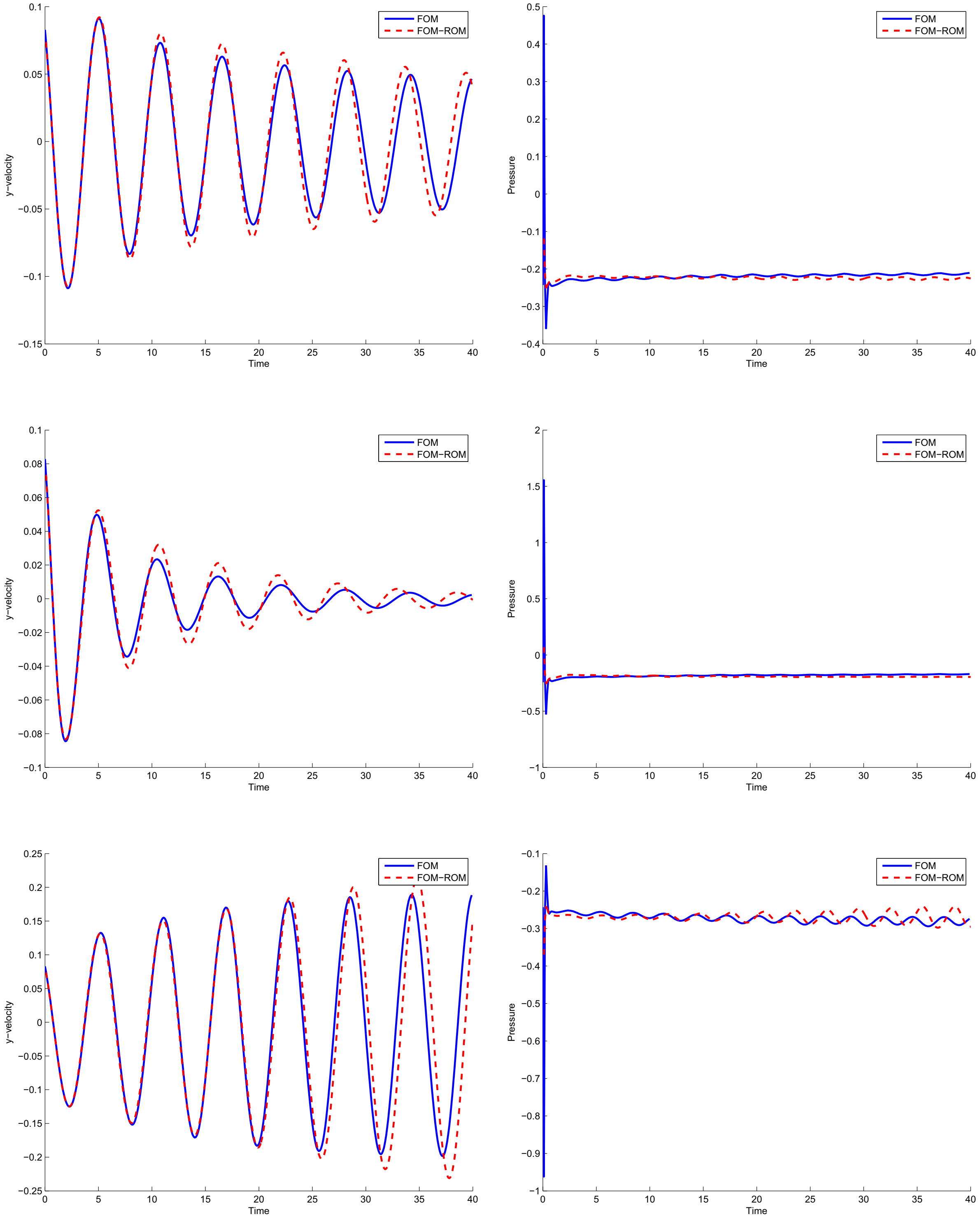

Fig. 14. Comparison of the vertical velocity (left) and pressure (right) at $(5, 4)$ for the FOM, FOM–ROM and ROM models for the injection case. Injection velocities, from top to bottom: 0.2, 0.5, -0.2.

of optimality of the snapshot set, the FOM–ROM model is capable of properly representing the solution in the FOM region. Fig. 14 shows a comparison between the FOM simulation and FOM–ROM model for several injection velocities. The accuracy of the FOM–ROM model decreases as the absolute value of the injection velocity increases. This is due to the fact that the larger the injection velocity, the more different the flow becomes from the original FOM simulation without injection.

Regarding the computational cost, the FOM–ROM approach takes 55.56 s to run, which is only 6.7% of the original FOM computational cost.

8. Conclusions

A domain decomposition strategy for non-linear hyper-reduced-order models has been presented. The basic idea consists of restricting the reduced-order basis functions to the nodes of each of the subdomains into which the physical domain has been partitioned. As a consequence, there are no interface nodes, but interface elements instead. This definition of the partitioned problem directly ensures the continuity of the recovered reduced-order solution. The local POD bases can be obtained by localizing a global POD basis (LG-POD), or by computing a local POD decomposition for the partitioned snapshots in each subdomain (L-POD). Instabilities in the form of large gradients of the recovered reduced-order unknown at the subdomain interfaces may appear when the proposed domain decomposition strategy is straightforwardly applied, which is the motivation for the design of additional stability terms in the form of penalty matrices. The penalty matrices are defined so that equality of the unknown at the interface (LG-POD) or at an overlapping region (L-POD) is reinforced in a weak sense.

An extension of the proposed domain decomposition strategy to a hybrid full-order/reduced-order model (FOM–ROM) is then described. This extension is directly obtained by taking as basis functions of the full-order subdomain the finite element shape functions of the nodes of the computational mesh.

The general domain decomposition approach for partitioned problems is particularized for the finite element approximation of the incompressible Navier–Stokes equations. The main feature with respect to the general approach is that only the equality of velocities is enforced at the interface or overlapping regions, while no additional constraints are set on the pressure field.

Numerical examples illustrate the behavior of the proposed method for the simulation of the hyper-reduced-order incompressible Navier–Stokes equations in a domain decomposition setting. The stability of the method is assessed in a domain decomposition reduced-order problem, and a numerical example is presented where the capability of the hybrid FOM–ROM to improve the performance of the ROM in flow configurations which are not present in the original snapshot set is shown.

Acknowledgement

This work was partially supported by the *Ministerio de Economía y Competitividad* of the Spanish Government under the Plan Nacional de Investigación 2012: AYA2012-33490, and 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 Governement through a "Juan de la Cierva" postdoctoral grant.

References

[1] I. Akhtar, J. Borggaard, A. Hay, Shape sensitivity analysis in flow models using a Finite-Difference approach, Math. Prob. Eng. 2010 (2010) 1–23.
[2] H. Antil, M. Heinkenschloss, R.H.W. Hoppe, Domain decomposition and balanced truncation model reduction for shape optimization of the stokes system, Optim. Methods Softw. 26 (4–5) (2011) 643–669.
[3] H. Antil, M. Heinkenschloss, R.H.W. Hoppe, D.C. Sorensen, Domain decomposition and model reduction for the numerical solution of PDE constrained optimization problems with localized optimization variables, Comput. Visual. Sci. 13 (6) (2010) 249–264.
[4] E. Arian, M. Fahl, E.W. Sachs, Trust-Region proper orthogonal decomposition for flow control, Technical Report, Institute for Computer Applications in Science and Engineering (ICASE), 2000, pp. 2000–2101.
[5] P. Astrid, Reduction of process simulation models: a proper orthogonal decomposition approach, Ph.D. Thesis, Department of Electrical Engineering, Eindhoven University of Technology, 2004.
[6] B.R. Noack, M. Morzynski, G. Tadmor, Reduced-Order Modelling for Flow Control, Springer, 2011.
[7] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 72 (12) (2013) 1219–1243.
[8] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C.R. Math. 339 (9) (2004) 667–672.
[9] M. Bergmann, L. Cordier, J.P. Brancher, Drag minimization of the cylinder wake by Trust-Region proper orthogonal decomposition, Notes Numer. Fluid Mech. Multidiscip. Des. 95 (2007) 19.
[10] M. Buffoni, H. Telib, A. Iollo, Iterative methods for model reduction by domain decomposition, Comput. Fluids 38 (6) (2009) 1160–1167.
[11] T. Bui-Thanh, K. Willcox, O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM J. Sci. Comput. 30 (6) (2008) 3270.
[12] J. Burkardt, M. Gunzburger, H. Lee, POD and CVT-based reduced-order modeling of Navier–Stokes flows, Comput. Methods Appl. Mech. Eng. 196 (1–3) (2006) 337–355.
[13] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Int. J. Numer. Methods Eng. 86 (2) (2011) 155–181.
[14] A. Chatterjee, An introduction to the proper orthogonal decomposition, Curr. Sci. 78 (7) (2000) 808–817.
[15] S. Chaturantabut, D.C. Sorensen, Discrete empirical interpolation for nonlinear model reduction, Technical Report TR09-05, Rice University, Houston Texas, 2009.
[16] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Comput. Methods Appl. Mech. Eng. 190 (2001) 2681–2706.
[17] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Eng. 191 (2002) 4295–4321.

[18] B. Galletti, C.H. Bruneau, L. Zannetti, A. Iollo, Low-order modelling of laminar flow regimes past a confined square cylinder, J. Fluid Mech. 503 (2004) 161–170.

[19] B. Glaz, L. Liu, P.P. Friedmann, Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework, AIAA J. 48 (10) (2010) 2418–2429.

[20] W.R. Graham, J. Peraire, K.Y. Tang, Optimal control of vortex shedding using low-order models. Part I – open-loop model development, Int. J. Numer. Methods Eng. 44 (7) (1999) 945–972.

[21] M.A. Grepl, Y. Maday, N.C. Nguyen, A.T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM: Math. Modell. Numer. Anal. 41 (03) (2007) 575–605.

[22] P. Holmes, J.L. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, 1998.

[23] T.J.R. Hughes, Multiscale phenomena: Green's function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations, Comput. Methods Appl. Mech. Eng. 127 (1995) 387–401.

[24] I. Kalashnikova, M.F. Barone, Stable and efficient Galerkin reduced order models for non-linear fluid flow, in: AIAA-2011-3110, 6th AIAA Theoretical Fluid Mechanics Conference, Honolulu, 2011.

[25] D.D. Kosambi, Statistics in function space, J. Indian Math. Soc. 7 (1943) 76–88.

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

[27] P.A. LeGresley, Application of proper orthogonal decomposition to design decomposition methods, Ph.D. Thesis, Department of Aeronautics and Astronautics, Stanford University, 2005.

[28] F. Lihong, Review of model order reduction methods for numerical simulation of nonlinear circuits, Appl. Math. Comput. 167 (1) (2005) 576–591.

[29] D.J. Lucia, P.S. Beran, Projection methods for reduced order models of compressible flows, J. Comput. Phys. 188 (1) (2003) 252–280.

[30] D.J. Lucia, P.I. King, P.S. Beran, Reduced order modeling of a two-dimensional flow with moving shocks, Comput. Fluids 32 (7) (2003) 917–938.

[31] C. Maier, A. Epureanu, V. Marinescu, F.M. Bogdan, 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, Stevens Point, Wisconsin, USA, World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 133–136.

[32] N.C. Nguyen, J. Peraire, An efficient reduced-order modeling approach for non-linear parametrized partial differential equations, Int. J. Numer. Methods Eng. 76 (1) (2008) 27–55.

[33] T. Rabczuk, S.P.A. Bordas, P. Kerfriden, O. Goury, A partitioned model order reduction approach to rationalise computational expenses in multiscale fracture mechanics, 2012. arXiv: 1208.3940v2 [physics.class-ph].

[34] G. Rozza, T. Lassila, A. Manzoni, Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map, in: J.S. Hesthaven, E.M. Ranquist (Eds.), Spectral and High Order Methods for Partial Differential Equations, vol. 76, Springer, Berlin, Heidelberg, 2011, pp. 307–315.

[35] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, J. Comput. Phys. 202 (1) (2005) 346–366.

[36] D. Ryckelynck, Hyper-reduction of mechanical models involving internal variables, Int. J. Numer. Methods Eng. 77 (1) (2009) 75–89.

[37] A. Verhoeven, T. Voss, P. Astrid, E.J.W. ter Maten, T. Bechtold, Model order reduction for nonlinear problems in circuit simulation, PAMM 7 (1) (2007) 1021603–1021604.

[38] Arie Verhoeven, Jan Maten, Michael Striebel, Robert Mattheij, Model order reduction for nonlinear ic models, in: Adam Korytowski, Kazimierz Malanowski, Wojciech Mitkowski, Maciej Szymkat (Eds.), System Modeling and Optimization, IFIP Advances in Information and Communication Technology, vol. 312, Springer, Berlin, Heidelberg, 2009, pp. 476–491.

[39] K. Veroy, A.T. Patera, Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds, Int. J. Numer. Methods Fluids 47 (8–9) (2005) 773–788.

[40] Z. Wang, I. Akhtar, J. Borggaard, T. Iliescu, Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. arXiv:1106.3585, 2011.

[41] M. Wicke, M. Stanton, A. Treuille, Modular bases for fluid dynamics, ACM Trans. Graph. 28 (3) (2009) 39:1–39:8.

[42] J. Yvonnet, Q. He, The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains, J. Comput. Phys. 223 (1) (2007) 341–368.