

Reduced-Order Modelling Strategies for the Finite Element Approximation of the Incompressible Navier-Stokes Equations

Joan Baiges, Ramon Codina and Sergio R. Idelsohn

Abstract In this chapter we present some Reduced-Order Modelling methods we have developed for the stabilized incompressible Navier-Stokes equations. In the first part of the chapter, we depart from the stabilized finite element approximation of incompressible flow equations and we build an explicit proper-orthogonal decomposition based reduced-order model. To do this, we treat the pressure and all the non-linear terms in an explicit way in the time integration scheme. This is possible due to the fact that the reduced model snapshots and basis functions do already fulfill an incompressibility constraint weakly. This allows a hyper-reduction approach in which only the right-hand-side vector needs to be reconstructed. In the second part of the chapter we present a domain decomposition approach for reduced-order models. The method consists in restricting the reduced-order basis functions to the nodes belonging to each of the subdomains. The method is extended to the particular case in which one of the subdomains is solved by using the high-fidelity, full-order model, while the other ones are solved by using the low-cost, reduced-order equations.

1 Introduction

Reduced-order models (ROM) are nowadays receiving a lot of interest from the computational mechanics community. Their most attractive feature is the capability of reproducing the response of complex physical phenomena through the solution of systems of equations which involve only very few degrees of freedom.

J. Baiges (✉) · R. Codina · S. R. Idelsohn
Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Edifici C1,
Campus Nord UPC C/ Gran Capità S/N, 08034 Barcelona, Spain
e-mail: jbaiges@cimne.upc.edu

J. Baiges, R. Codina
Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

S. R. Idelsohn
Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain

Amongst the various families of reduced-order models, Proper Orthogonal Decomposition [14, 23, 26] based ROMs consist in the training of the model by taking snapshots from a high-fidelity simulation and using them to build an orthogonal basis which is capable of accurately representing the solution through the combination of few of this basis functions. Particularly, we are interested in the application of POD models to the incompressible Navier-Stokes equations, which we originally approximate by using finite elements and a stabilized formulation. The problem of applying POD models to the incompressible flow equations has been approached by several authors [12, 19, 20, 25, 29, 39, 40] in a range of applications like shape optimization [1, 9, 27, 34] or flow control [3, 21, 32].

The major concerns when making use of a reduced-order model are, on the one hand, computational cost, and on the other, accuracy. Obviously we are looking for a cheap reduced-order model which is as accurate as possible. Unfortunately, this is not always possible. In this chapter we present some approaches we have developed for POD models for the incompressible Navier-Stokes equations which help enhance the computational cost and accuracy of the reduced-order models.

One of the a priori drawbacks of traditional POD is that a straightforward application of a POD strategy to a non-linear problem does not turn out in a drastic reduction of the required computational cost. This is so because in order to solve the reduced-order equations, the full-order, non-linear, system of equations needs to be built first and then projected onto the reduced-order space. Recently, the so-called hyper-reduction [4, 8, 15, 22, 31, 35–38] has appeared as a means to circumvent this problem. The main idea is to compute the non-linear system entries only at some few nodes of the computational mesh, and then approximate the whole system by extrapolating it from the values of the system at these entries.

In the first part of this chapter, we describe a reduced-order model which is particularly suitable for hyper-reduction [7]. 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. This results in a reduced-order model where only the right-hand side of the system needs to be rebuilt at each time step. This is possible because the reduced model snapshots do already fulfill the stabilized continuity equation and the pressure field can be automatically recovered at the end of each time step from the reduced order basis and solution coefficients. We also present a method for choosing the sampling entries from which the complete system is going to be extrapolated. 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.

Another issue which needs to be dealt with when using reduced-order models is the lack of robustness 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 [3], 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.

In the second part of this chapter we present a domain decomposition method for reduced-order models [6] which we apply to the finite element approximation of the incompressible Navier-Stokes equations. Domain decomposition methods for reduced-order models have been used for different simulation problems [2, 10, 28, 30, 33, 41]. In these partitioned approaches reduced-order models are formulated independently and then glued together in either a monolithic or an iterative way. Contrary to this approach, 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. This definition of the partitioned problem directly ensures the continuity of the recovered reduced-order solution at the interfaces. Also, there is no need to use the classical domain-decomposition iteration by subdomain schemes, because the Domain Decomposition Reduced-Order Model (DD-ROM) is written in terms of the partitioned reduced bases in a monolithic way. One of the advantages of the proposed method is the ease for generating a hybrid full-order/reduced-order model, as a particular case of the general DD-ROM method. The proposed hybrid DD-ROM model can be easily used together with hyper-reduced models, as we demonstrate in the numerical examples section.

The chapter is organized as follows. In Sect. 2 we present an explicit ROM for the finite element approximation of the incompressible Navier-Stokes equations, and a numerical example illustrates the behavior of the model for low Reynolds flow cases. In Sect. 3 we describe the hyper-reduction strategy we apply to the explicit ROM, and we also present the Discrete Best Points Interpolation Method for the selection of sampling indices for the gappy-POD reconstruction process. Finally, the domain-decomposition reduced-order model is presented in Sect. 4, where we also explain the hybrid full-order/reduced-order domain decomposition approach and present a numerical example. Some conclusions close the chapter in Sect. 5.

2 An Explicit Reduced-Order Model for the Incompressible Navier-Stokes Equations

When solving a non-linear problem by means of a POD based ROM, it is necessary to project the full-order system of equations to the reduced-order space at each iteration of the non-linear problem. For non-linear problems, this is troublesome because the expected orders of magnitude reduction in the computational cost of solving the reduced-order system is not observed in practice: the computational time of the reduced-order model is governed by the need of rebuilding the full-order system at each iteration, and then projecting it to the reduced-order subspace.

This issue has motivated a lot of research recently, leading to several strategies which reduce the cost of computing projected reduced-order system of equations

[4, 8, 15, 22, 31, 35–38]. These approaches are known as hyper-reduced models. In these methods, the non-linear 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. This allows to effectively reduce the amount of computations required to build the reduced order system, and results in a reduced-order model whose computational cost is directly proportional to its number of degrees of freedom.

We have been working in a hyper-reduced approach for the incompressible Navier-Stokes equations. The particularity of the strategy we propose is that the equations for the reduced-order model are treated in an explicit way. This allows to send all the non-linear terms to the right hand-side of the reduced-order system, leaving in the left-hand side only the mass matrix due to the temporal derivatives. The main advantage is, of course, that the mass matrix is linear, and the hyper-reduced approaches need only to be applied to right-hand side vector. This effectively reduces the overall cost of the reduced-order model.

Let us start by introducing some notation for the POD approximation of a general problem. Let $\mathbf{U} \in \mathbb{R}^M$ be the global unknown vector associated to a non-linear variational problem. Suppose that after linearizing and fully discretizing in time and space the given problem, the following matrix form is obtained which allows to obtain the vector of nodal unknowns \mathbf{U} at a given iteration of the non-linear procedure, for a certain time step:

$$\mathbf{A}\mathbf{U} = \mathbf{F}, \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{M \times M}$ is the matrix of the system whose solution is \mathbf{U} , and $\mathbf{F} \in \mathbb{R}^M$ the RHS vector. The POD approximation of the previous system is obtained by projecting it onto a low dimensional subspace $\mathcal{U} \subset \mathbb{R}^N$. Vectors \mathbf{U} are now approximated by:

$$\mathbf{U} \approx \Phi\boldsymbol{\alpha}, \quad (2)$$

where $\Phi \in \mathbb{R}^{M \times N}$ 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 Φ . The reduced-order basis Φ is obtained by means of the POD method [14, 23, 26], that is by doing the singular value decomposition of a set of solution snapshots, which in our case are taken from the results of a full-order simulation. After projecting the full-order system to this reduced-order subspace and applying a least squares approach, the final reduced-order system is:

$$\Phi^T \mathbf{A} \Phi \boldsymbol{\alpha} = \Phi^T \mathbf{F}. \quad (3)$$

2.1 Stabilized Finite Element Approximation of the Incompressible Navier-Stokes Equations

In this section we summarize the finite element stabilized formulation for the incompressible Navier-Stokes equations used in the rest of the chapter. Let us consider the transient incompressible Navier-Stokes equations, which consist of finding $\mathbf{u} : \Omega \times (0, T) \rightarrow \mathbb{R}^d$ and $p : \Omega \times (0, T) \rightarrow \mathbb{R}$ such that:

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

for $t > 0$, where $\partial_t \mathbf{u}$ 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 \mathbf{f} the given source term. Appropriate initial conditions have to be appended to this problem.

Let now $V = H^1(\Omega)^d$, and $V_0 = \{\mathbf{v} \in V \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$. 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 $[\mathbf{u}, p] \in L^2(0, T; V) \times \mathcal{D}'(0, T; Q)$ such that:

$$(\mathbf{v}, \partial_t \mathbf{u}) + B([\mathbf{v}, q], [\mathbf{u}, p]) = \langle \mathbf{v}, \mathbf{f} \rangle \quad \forall [\mathbf{v}, q] \in V \times Q, \quad (4)$$

with

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

where

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

Here, (\cdot, \cdot) stands for the $L^2(\Omega)$ inner product and $\langle \cdot, \cdot \rangle$ for the integral of the product of two functions, not necessarily in $L^2(\Omega)$. Let $\{K\}$ be a finite element partition of Ω , from which we construct the finite element spaces $V_h \subset V$, $V_{h0} \subset V_0$, $Q_h \subset Q$. The semilinear form B suffers from the well-known stability issues due to the convective nature of the flow, but also requires a compatibility between the velocity and pressure approximation spaces due to the classical LBB inf-sup condition. In order to deal with these stability issues, we use a stabilized finite element formulation [16], which is as follows: for each t , find $\mathbf{u}_h(t) \in V_h$, $p_h(t) \in Q_h$ such that:

$$\begin{aligned} (\mathbf{v}_h, \partial_t \mathbf{u}_h) + B([\mathbf{v}_h, q_h], [\mathbf{u}_h, p_h]) + \sum_K \tau_K (\mathbf{u}_h \cdot \nabla \mathbf{v}_h + \nu \Delta \mathbf{v}_h \\ + \nabla q_h, \mathbf{r}([\mathbf{u}_h, p_h]))_K = \langle \mathbf{v}_h, \mathbf{f} \rangle, \quad (5) \end{aligned}$$

for all $\mathbf{v}_h \in V_{h,0}$, $q_h \in Q_h$. Initial conditions need to be appended to this problem. In (5):

$$\mathbf{r}([\mathbf{u}_h, p_h]) = \partial_t \mathbf{u}_h - \nu \Delta \mathbf{u}_h + \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla p_h - \mathbf{f}, \quad (6)$$

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:

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

where $|\mathbf{u}_h|_K$ is the mean velocity modulus in element K , h is the element size and c_1 and c_2 are stabilization constants.

Regarding the discretization in time, we consider implicit integration schemes. For the full-order system, only implicit time integration schemes can be used, because no time derivatives of the pressure appear in the equations. Taking this into account, we can do the following: supposing that the velocity and pressure at time step n $[\mathbf{u}_h^n, p_h^n]$ are known, we may solve (5) for example with $\partial_t \mathbf{u}_h$ being discretized using a backward differences in time scheme:

$$\begin{aligned} \partial_t \mathbf{u}_h &\approx \delta_t \mathbf{u}_h^{n+1}, \\ \delta_t \mathbf{u}_h^{n+1} &:= \begin{cases} \frac{1}{\delta t} (\mathbf{u}_h^{n+1} - \mathbf{u}_h^n) & \text{1st order scheme} \\ \frac{1}{\delta t} (\frac{3}{2} \mathbf{u}_h^{n+1} - 2\mathbf{u}_h^n + \frac{1}{2} \mathbf{u}_h^{n-1}) & \text{2nd order scheme} \end{cases} \end{aligned} \quad (7)$$

where δt is the time step size.

2.2 Explicit Reduced-Order Model

As explained in the previous sections, it is convenient to treat the reduced-order model by using an explicit time integration scheme, because this leads to important computational gains when using hyper-reduced order reconstruction methods. However, we have also explained the need of using an implicit time integration scheme for the full-order model of the incompressible Navier-Stokes equations, due to the presence of the pressure field. Here we summarize the strategy we use for building an explicit reduced-order model which is suitable for the incompressible Navier-Stokes equations [7].

Let us start by introducing the velocity and pressure reduced-order subspaces. $\hat{Q} \subset Q_h$ is the pressure subspace defined by the pressure part of the POD basis functions Φ , $\hat{p} \in \hat{Q}$ is the reduced-order pressure field. $\hat{V} \subset V_h$ is the velocity subspace defined by the velocity part of the POD basis functions Φ . For each time t , $\hat{\mathbf{u}}(t) \in \hat{V}$ is the reduced-order velocity. In order to develop the explicit reduced-order model where the pressure is treated in an explicit way, we take into account that:

- All reduced basis functions do already fulfill the stabilized continuity equation. Since the reduced-order basis is built from weakly incompressible solution snapshots and the incompressibility constraint is linear, the reduced basis functions (and their linear combinations) do also fulfill it.
- If basis functions are taken to be joint velocity-pressure basis functions (that is Φ contains the coefficients of functions in $\hat{V} \times \hat{Q}$), then the pressure at time step $n + 1$ is automatically recovered from coefficients α_{n+1} and the reduced order basis Φ even if all the terms involving the pressure are treated in an explicit way in the reduced order formulation.

The variational formulation for the first order in time reduced-order model that we propose is:

$$\begin{aligned}
 & (\hat{\mathbf{v}}, \delta_t \hat{\mathbf{u}}^{n+1}) + (\hat{\mathbf{v}}, \hat{\mathbf{u}}^{n*} \cdot \nabla \hat{\mathbf{u}}^{n*}) + \nu (\nabla \hat{\mathbf{v}}, \nabla \hat{\mathbf{u}}^{n*}) - (\hat{p}^{n*}, \nabla \cdot \hat{\mathbf{v}}) \\
 & + \sum_K \tau_K (\hat{\mathbf{u}}^n \cdot \nabla \hat{\mathbf{v}} + \nu \Delta \hat{\mathbf{v}}, \delta_t \hat{\mathbf{u}}^n - \nu \Delta \hat{\mathbf{u}}^n + \hat{\mathbf{u}}^n \cdot \nabla \hat{\mathbf{u}}^n + \nabla \hat{p}^n - \mathbf{f}^n)_K = (\hat{\mathbf{v}}, \mathbf{f}^n).
 \end{aligned} \tag{8}$$

where the terms $\hat{\mathbf{u}}^{n*}$ 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:

$$\begin{aligned}
 \hat{\mathbf{u}}^{n*} &= 2\hat{\mathbf{u}}^n - \hat{\mathbf{u}}^{n-1}, \\
 \hat{p}^{n*} &= 2\hat{p}^n - \hat{p}^{n-1}.
 \end{aligned} \tag{9}$$

In the case of the second order in time reduced-order model, we use the same variational formulation (8), but the terms $\hat{\mathbf{u}}^{n*}$ and \hat{p}^{n*} are now a third order approximation of the state at $n + 1$ given by:

$$\begin{aligned}
 \hat{\mathbf{u}}^{n*} &= \frac{12}{5}\hat{\mathbf{u}}^n - \frac{9}{5}\hat{\mathbf{u}}^{n-1} + \frac{2}{5}\hat{\mathbf{u}}^{n-2}, \\
 \hat{p}^{n*} &= \frac{12}{5}\hat{p}^n - \frac{9}{5}\hat{p}^{n-1} + \frac{2}{5}\hat{p}^{n-2}.
 \end{aligned} \tag{10}$$

Note that for the first order explicit scheme we propose to use the second order extrapolation (9), and for the second order scheme the third order extrapolation (10).

The key point of this formulation is that only the temporal derivative terms involve values of the reduced-order velocity or pressures at the new time step. This ensures that the resulting reduced-order matrix is linear. However, the reduced-order right-hand-side still needs to be approximated. After solving the reduced-order system, the velocity and pressure fields at $n + 1$ can be recovered by multiplying the reduced-order basis Φ by the obtained reduced-order components α_{n+1} .

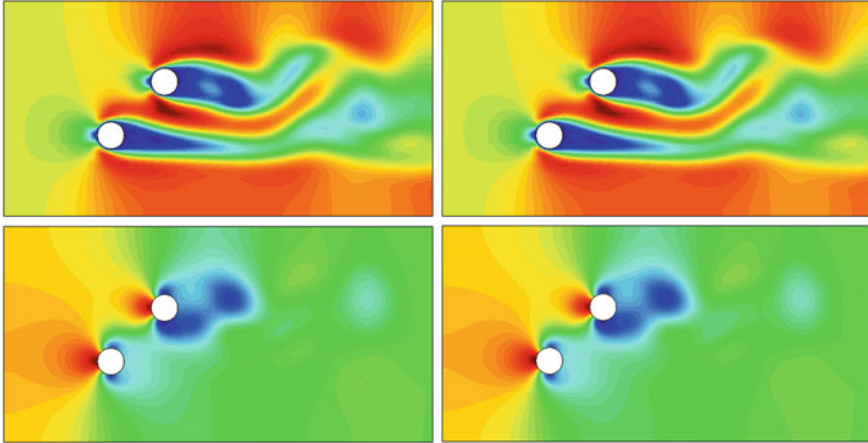


Fig. 1 Comparison of the FOM (*left*) and ROM (*right*) velocities (*top*) and pressures (*bottom*) after 400 time steps of simulation

2.3 Numerical Example. Bidimensional Flow Past Two Cylinders

The first numerical example consists in the bidimensional flow past two cylinders. The computational domain is a 16×8 rectangle. The cylinders are centered at coordinates $(3, 3)$ and $(6, 5)$, and both of them are of diameter 1. The inflow velocity is 1, which together with the density $\rho = 1$ and the viscosity $\mu = 0.01$ results in a Reynolds number $Re = 100$. The time step is set to $\delta t = 0.1$. The mesh is composed of 7310 linear triangular elements. After running the full-order simulation and taking the corresponding snapshots, the explicit reduced-order model is run. The number of degrees of freedom for the ROM is only 10.

Figure 1 shows a comparison of the velocity and pressure fields for the full-order and the explicit reduced-order model after 400 time steps of simulation. The high-fidelity and the reduced-order fields are very similar. In Fig. 2 we compare the time history and Fourier transform of the vertical velocity and the pressure at coordinates $(8.5, 4)$. We observe that the time history and Fourier transform of both vertical velocity and pressure are accurate for the reduced-order model. The cpu-time for running the full-order model is 53.24s, the time for running the explicit reduced-order model is 19.78s, a 63 % reduction in computational time.

3 Hyper-Reduction Approach

At this point, we already have an explicit reduced-order model in which all the non-linear terms are in the right-hand-side vector and the reduced system matrix is linear and does not change between time steps. However, computing the right-hand-side

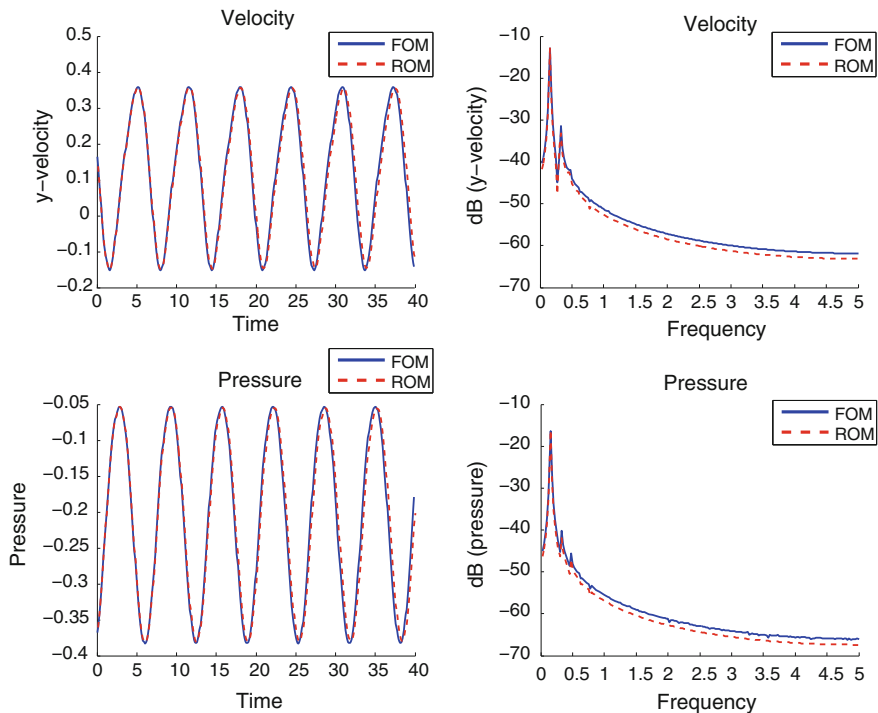


Fig. 2 Comparison of the FOM and ROM velocity and pressure time history at the point (8.5, 4) (left) and their Fourier transform (right)

vector at each time step is still expensive (number of operations of $\mathcal{O}(M)$), because we need to recompute \mathbf{F}^{n+1} and then project it to the reduced-order subspace by calculating $\Phi^T \mathbf{F}^{n+1}$. The approach we follow for reducing this computational cost is to reconstruct the non-linear vector \mathbf{F}^{n+1} by sampling only some of the entries of this vector and applying a least-squares minimization strategy. The method we follow was first presented in [18], and a similar approach has been recently used in [13] applied to an implicit reduced-order method for the incompressible Navier-Stokes equations.

Let us consider a reduced order basis for the right-hand-side vectors \mathbf{F} , Φ_F , obtained by means of a proper orthogonal decomposition of a set of snapshots for \mathbf{F} . Φ_F defines a low-dimensional subspace $\mathcal{F} \subset \mathbb{R}^M$, so that any right-hand-side vector \mathbf{F} can be approximated as:

$$\mathbf{F} \simeq \Phi_F \mathbf{F}_\phi,$$

where now $\mathbf{F}_\phi \in \mathbb{R}^N$ are the reduced-order coefficients for the reconstruction. Let us also consider that we only know the nodal values for \mathbf{F} at some sampling components $F_{i(k)}$, $1 \leq k \leq n_s$, where n_s is the number of sampling components of the vector,

$i(k)$ denotes the k th sampling component. We now want to recover the reduced order basis coefficients \mathbf{F}_Φ of the reduced order basis Φ_F for vector \mathbf{F} .

In order to recover \mathbf{F}_Φ we can solve the least-squares minimization problem:

$$\mathbf{F}_\Phi = \arg \min_{\mathbf{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^N (\Phi_{F,i(k)j} a_j - F_{i(k)})^2, \quad (11)$$

where $\Phi_{F,i(k)j}$ denotes the basis vector j evaluated at the k th sampling component, $i(k)$.

The previous procedure provides the tools required to extrapolate the right-hand-side vector 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. If n_s is $\mathcal{O}(N)$, then the computational cost of rebuilding \mathbf{F}^{n+1} for solving each time step is reduced to $\mathcal{O}(N)$, and the overall cost of the reduced-order model is $\mathcal{O}(N)$.

3.1 A Discrete Version of the Best Points Interpolation Method (DBPM)

When using hyper-reduced order models 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 [5, 8, 17]. Amongst the most extensively used are the *Discrete Empirical Interpolation Method (DEIM)* [15], where the sampling components are selected iteratively by imposing that the error growth at each iteration is limited and the *Best Points Interpolation Method (BPIM)* approach presented in [31], 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.

The strategy we use, presented in [7], is a hybrid between the BPIM and the DEIM. We call it a *Discrete version of the Best Point Interpolation Method (DBPIM)*. 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, due to the lack of smoothness of the vectors which are being approximated 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 neighbour points in the finite element mesh and replaces a sampling node with its neighbour 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).

The first step of the DBPIM algorithm consists of finding the projection $\Pi_{\mathcal{F}}$ of the snapshots onto the reduced order subspace defined by the reduced order basis, \mathcal{F} . For each snapshot, this yields the coefficients $\mathbf{F}_{\Phi}^{\alpha}$. In the second step we choose an initial set of sampling nodes, which can be done 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 $\mathbf{F}_{\Phi}^{\alpha, \text{approx}}$ by means of the previously described least-squares strategy. The error associated to a set of sampling components $\mathbf{i} \in \mathbb{N}^{n_s}$, whose k -th component is indicated as $i(k) \in 1, \dots, M$, is obtained by computing the difference between the exact and the approximated $\mathbf{F}_{\Phi}^{\alpha}$ coefficients:

$$e(\mathbf{i}) = \sum_{\alpha=1}^{N_{\text{snapshots}}} \|\mathbf{F}_{\Phi}^{\alpha, \text{approx}}(\mathbf{i}) - \mathbf{F}_{\Phi}^{\alpha}\| \quad (12)$$

Algorithm 1 Discrete best points interpolation method

Compute the optimal basis coefficients for the snapshot set:

$\Phi_F \mathbf{F}_{\Phi}^{\alpha} = \Pi_{\mathcal{F}}(\mathbf{F}^{\alpha})$, $\alpha = 1, N_{\text{snapshots}}$

Choose an initial set of sampling components $\mathbf{i} \in \mathbb{N}^{n_s} \mid 1 \leq i(k) \leq M$, $k = 1, \dots, n_s$.

Solve: $\mathbf{F}_{\Phi}^{\alpha, \text{approx}}(\mathbf{i}) = \arg \min_{\mathbf{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^N (\Phi_{F, i(k)j} a_j - F_{i(k)}^{\alpha})^2$, $\alpha = 1, N_{\text{snapshots}}$

$e(\mathbf{i}) = \sum_{j=1}^{N_{\text{snapshots}}} \|\mathbf{F}_{\Phi}^{\alpha, \text{approx}}(\mathbf{i}) - \mathbf{F}_{\Phi}^{\alpha}\|^2$

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 l th neighbour of $i(m)$ has not been previously tested **then**

 Temporarily replace sampling node $i(m)$ by its l th neighbour

 Solve: $\mathbf{F}_{\Phi}^{\alpha, \text{approx}}(\mathbf{i}) = \arg \min_{\mathbf{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^N (\Phi_{F, i(k)j} a_j - F_{i(k)}^{\alpha})^2$, $\alpha = 1, N_{\text{snapshots}}$

$e_{\text{temp}}(\mathbf{i}) = \sum_{\alpha=1}^{N_{\text{snapshots}}} \|\mathbf{F}_{\Phi}^{\alpha, \text{approx}}(\mathbf{i}) - \mathbf{F}_{\Phi}^{\alpha}\|$

if $e_{\text{temp}} < e$ **then**

$e = e_{\text{temp}}$

 Permanently replace sampling node $i(m)$ by its l th neighbour

 Restart l loop

end if

end if

end for

end while

where

$$\mathbf{F}_\Phi^{\alpha, \text{aprox}}(\mathbf{i}) = \arg \min_{\mathbf{a} \in \mathbb{R}^N} \sum_{k=1}^{n_s} \sum_{j=1}^N (\Phi_{F, i(k)j} a_j - F_{i(k)}^\alpha)^2, \alpha = 1, N_{\text{snapshots}} \quad (13)$$

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

$$\mathbf{b} = \arg \min_{\mathbf{i} \in \mathbb{N}^{n_s} \mid 1 \leq i(k) \leq M, k=1, \dots, n_s} e(\mathbf{i}) \quad (14)$$

where $e(\mathbf{i})$ is given in (12).

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 neighbours 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 neighbour. This procedure is repeated while the set of sampling points changes due to the algorithm (**while** loop in Algorithm 1).

3.2 Numerical Example. Two-dimensional Low Reynolds Flow Past a NACA Airfoil

In this section we simulate the incompressible flow around a NACA 0012 airfoil profile [24]. The computational domain is a 32×16 rectangle, with the trailing edge of the 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 upper and lower 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 $Re = 1000$ 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 29945 linear element mesh has been used. The mesh is refined around the airfoil surface in order to be able to better capture the solution in the region surrounding the boundary layer. 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.

100 velocity-pressure snapshots have been taken and the 10 first reduced basis functions have been kept for the reduced-order model. For the hyper-reduced order model, 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.

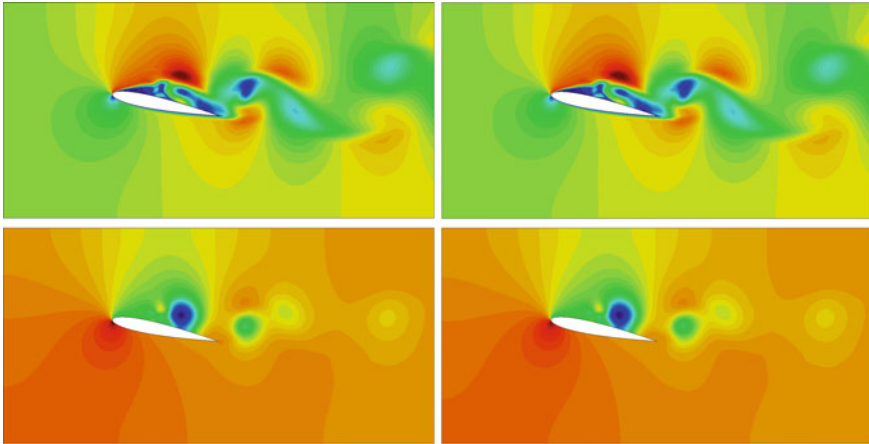


Fig. 3 Velocity (*top*) and pressure (*bottom*) contours at $Re = 1000$, $\alpha = 0.2$ after 200 time steps. Full-order (*left*) and Hyper-Reduced Order Model (*right*)

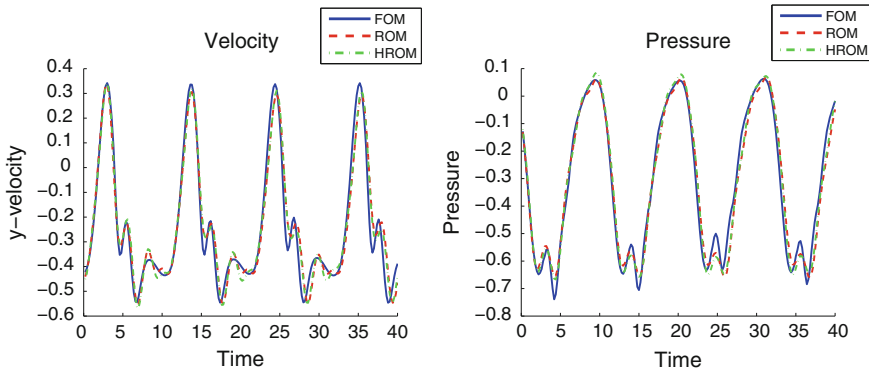


Fig. 4 Velocity (*left*) and pressure (*right*) time history at a control point at the wake of the airfoil, $Re = 1000$, $\alpha = 0.2$, second order time integration

Figure 3 compares the velocity and pressure fields after 200 time steps for the full-order and the hyper-reduced model. The reduced-order model almost exactly matches the results from the full order model.

Regarding the computational cost, the full order model takes 148.9 s to run, the reduced-order model takes 49.6 s (33 %). Finally, reduced-order model 2, in which the computational cost depends only on the size of the reduced-order model, takes only 0.71 s (0.45 %) to run.

Figures 4 and 5 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 reduced-order models manage to correctly capture the main modes amplitudes and frequencies.

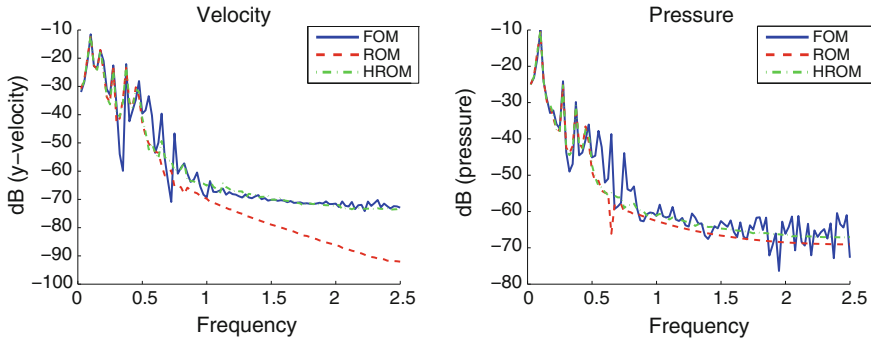


Fig. 5 Velocity (*left*) and pressure (*right*) spectra at a control point at the wake of airfoil, $Re = 1000$, $\alpha = 0.2$, second order time integration

4 A Domain Decomposition Approach for POD Reduced-Order Models

Despite the important reduction in computational cost provided by reduced-order models, one of their major drawbacks is the lack of robustness 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 [3], requiring the snapshot collection and the reduced model to be updated when, for instance, an optimization process leads to a parameter configuration which becomes too separated from the starting parameter set.

In this section we present 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 [6]. It is based on introducing a domain decomposition approach to the model reduction, partitioning the computational domain into several regions, each of which is dealt with localized POD bases. This gives us the possibility of treating each of the subdomains with a different degree of approximation, or even, as we will see, solving the full-order equations in some of the subdomains.

4.1 Domain Decomposition POD Model

Let us consider the splitting of the computational domain Ω into two subdomains Ω_k , $k = 1, 2$, and the associated local unknowns $\mathbf{U}^k \in \mathbb{R}^{M_k}$, $M = M_1 + M_2$. 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 ϕ^k consisting of the reduced basis functions $\phi_i^k \in \mathbb{R}^{M_k}$, $i = 1, \dots, N_k$, to approximate U^k in each subdomain. Note that the number of basis functions in each subdomain is not necessarily the same, although we have considered it to be equal from now on for simplicity. The possible ways to construct this basis are discussed later. This local basis can be extended to the global domain by defining $\Phi_i^k \in \mathbb{R}^M$:

$$\Phi_i^1 := \begin{bmatrix} \phi_i^1 \\ \mathbf{0} \end{bmatrix}, \quad \Phi_i^2 := \begin{bmatrix} \mathbf{0} \\ \phi_i^2 \end{bmatrix}, \quad (15)$$

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

$$U \approx \sum_{i=1}^N (\Phi_i^1 \alpha_i^1 + \Phi_i^2 \alpha_i^2) = (\Phi^1 \alpha^1 + \Phi^2 \alpha^2), \quad \Phi^k \in \mathbb{R}^{M \times N}, \quad \alpha^k \in \mathbb{R}^N \quad k = 1, 2, \quad (16)$$

where α^k are the solution coefficients for subdomain k .

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

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

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:

$$\begin{aligned} (\Phi^1)^T A (\Phi^1 \alpha^1 + \Phi^2 \alpha^2) &= (\Phi^1)^T F \\ (\Phi^2)^T A (\Phi^1 \alpha^1 + \Phi^2 \alpha^2) &= (\Phi^2)^T F. \end{aligned} \quad (17)$$

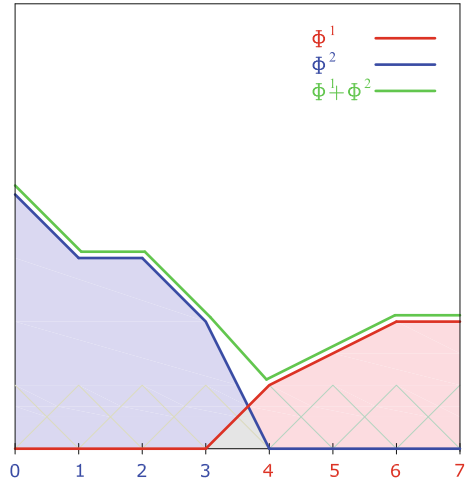
Defining $a_{kl} = (\Phi^k)^T A \Phi^l \in \mathbb{R}^{N \times N}$ and $f_k = (\Phi^k)^T F \in \mathbb{R}^N$, we may write this system as

$$a_{11} \alpha^1 + a_{12} \alpha^2 = f_1, \quad (18)$$

$$a_{21} \alpha^1 + a_{22} \alpha^2 = f_2. \quad (19)$$

If we also consider the decomposition of A and F the final reduced order system can be written in terms of the local bases ϕ^k :

Fig. 6 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*)



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

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 $2N$).

An important point is that each algebraic local basis function Φ_i^k arises from a function defined in space. This spatial function is a linear combination of the finite element shape functions of the nodes of subdomain Ω_k . As a consequence, each of the components in Φ_i^k corresponds to a nodal value of the spatial field to be represented on the finite element mesh. This is illustrated in Fig. 6, 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 (15). This will also hold for the combination of local basis functions, even if these belong to different subdomains. In Fig. 6 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.

4.2 Local POD (L-POD)

The strategy for building the local POD basis consists in performing a POD for the part of the snapshots corresponding to each of the subdomains. The snapshots are first partitioned according to the domain decomposition strategy and the local basis ϕ^k is obtained from these partitioned snapshots. The global basis is again defined as $\Phi = [\Phi^1, \Phi^2]$. Note that the number of local basis functions in each subdomain does not necessarily coincide, $N_1 \neq N_2$, $N = N_1 + N_2$.

The main features of these domain-decomposition local POD bases are the following:

- Each 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.
- 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.

4.3 Stabilization Through Overlapping and Penalty Terms

The previous domain decomposition strategy for reduced-order models, despite its simplicity, suffers from unstable behavior when it is used in a straightforward manner in the explicit reduced-order model for the stabilized finite element approximation of the incompressible Navier-Stokes equations described in the previous sections. These instabilities can be easily explained taking into account that an explicit time marching scheme is equivalent in this case to an explicit iteration-by-subdomain strategy, which is known to have convergence and stability issues. This is the reason

why we propose a domain interface stabilization term, which is obtained by allowing some overlapping between subdomains and enforcing the equality of the unknown values at this overlapping region.

As in classical iteration by subdomain strategies, the overlapping region Ω_\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 $\Phi^{01} \in \mathbb{R}^{M \times N_1}$ and $\Phi^{02} \in \mathbb{R}^{M \times N_2}$ are:

$$\Phi^{01} = \begin{bmatrix} \phi^1 \\ \phi^{1\cap} \\ \mathbf{0} \end{bmatrix}, \quad \Phi^{02} = \begin{bmatrix} \mathbf{0} \\ \phi^{2\cap} \\ \phi^2 \end{bmatrix}, \quad (20)$$

where now

$$\phi^k \in \mathbb{R}^{M_k \times N_k}, \quad (21)$$

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

$$\phi^{k\cap} \in \mathbb{R}^{M_{k\cap} \times N_k}, \quad (22)$$

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

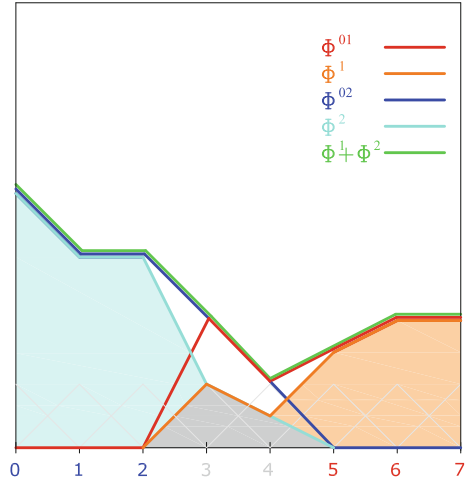
In this case the corrected bases are:

$$\Phi^1 = \begin{bmatrix} \phi^1 \\ \beta\phi^{1\cap} \\ \mathbf{0} \end{bmatrix}, \quad \Phi^2 = \begin{bmatrix} \mathbf{0} \\ (1-\beta)\phi^{2\cap} \\ \phi^2 \end{bmatrix}, \quad (23)$$

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. 7, where some illustrative basis functions for a one-dimensional problem are depicted.

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

Fig. 7 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



$$\phi^{1\cap} \alpha^1 = \mathbf{U}^{1\cap} = \mathbf{U}^{2\cap} = \phi^{2\cap} \alpha^2 \in \mathbb{R}^{M\cap}, \quad (24)$$

where now we take $\alpha^k \in \mathbb{R}^{N_k}$ the ROM degrees of freedom for each subdomain.

This condition can be equivalently written as:

$$\begin{aligned} (\Phi^{1\cap})^T \Phi^{1\cap} \alpha^1 - (\Phi^{1\cap})^T \Phi^{2\cap} \alpha^2 &= \mathbf{0}, \\ (\Phi^{2\cap})^T \Phi^{1\cap} \alpha^1 - (\Phi^{2\cap})^T \Phi^{2\cap} \alpha^2 &= \mathbf{0}, \end{aligned} \quad (25)$$

where

$$\Phi^{k\cap} = \begin{bmatrix} \mathbf{0} \\ \phi^{k\cap} \\ \mathbf{0} \end{bmatrix}. \quad (26)$$

Introducing (25) as a penalized constraint in the ROM system we get:

$$\mathbf{a}_{11} \alpha^1 + \mathbf{a}_{12} \alpha^2 + \frac{1}{\varepsilon} (\mathbf{M}_{11} \alpha^1 - \mathbf{M}_{12} \alpha^2) = \mathbf{f}_1, \quad (27)$$

$$\mathbf{a}_{21} \alpha^1 + \mathbf{a}_{22} \alpha^2 + \frac{1}{\varepsilon} (\mathbf{M}_{21} \alpha^1 - \mathbf{M}_{22} \alpha^2) = \mathbf{f}_2, \quad (28)$$

where

$$\mathbf{M}_{kl} = (\Phi^{k\cap})^T \Phi^{l\cap} \in \mathbb{R}^{N_k \times N_l}. \quad (29)$$

and the definition of Φ^k for building the \mathbf{a} matrices is taken as in (23).

An important property of the block diagonal penalty matrices \mathbf{M}_{kk} is that they can only be guaranteed to be full-rank matrices if $\Omega_\cap = \Omega$. However, this stabilization strategy shows good results in the numerical examples even if $\Omega_\cap \neq \Omega$. The introduction of the \mathbf{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 $\frac{1}{\varepsilon}\mathbf{M}$ is proportional to the norm of \mathbf{A} . In this way we ensure that the resulting system does not become ill-conditioned.

4.4 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 these 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. The hybrid FOM-ROM system without overlapping is:

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

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 the explicit reduced order model described in the previous sections is used for the incompressible Navier-Stokes equations, the \mathbf{A} matrix and the \mathbf{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:

$$\begin{bmatrix} \mathbf{A}|_{FF} & \mathbf{A}|_{FR}\phi^R \\ (\phi^R)^T \mathbf{A}^{\text{exp}}|_{RF} & (\phi^R)^T \mathbf{A}^{\text{exp}}|_{RR}\phi^R \end{bmatrix} \begin{bmatrix} \mathbf{U}^F \\ \boldsymbol{\alpha}^R \end{bmatrix} = \begin{bmatrix} \mathbf{F}|_F \\ (\phi^R)^T \mathbf{F}^{\text{exp}}|_R \end{bmatrix}. \quad (31)$$

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:

$$\begin{bmatrix} \mathbf{A}|_{FF} & \mathbf{A}|_{FR}\phi^R \\ \mathbf{A}_{RF}^{PG} & \mathbf{A}_{RR}^{PG}\phi^R \end{bmatrix} \begin{bmatrix} \mathbf{U}^F \\ \boldsymbol{\alpha}^R \end{bmatrix} = \begin{bmatrix} \mathbf{F}|_F \\ \mathbf{F}_R^{PG} \end{bmatrix}, \quad (32)$$

where

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

Also, 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.

4.5 Particularities of the Application to the Incompressible Navier-Stokes Equations

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 neighbour 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.

4.6 Numerical Example. 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 $Re = 100$. The computational domain consists of a 24×12 rectangle 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

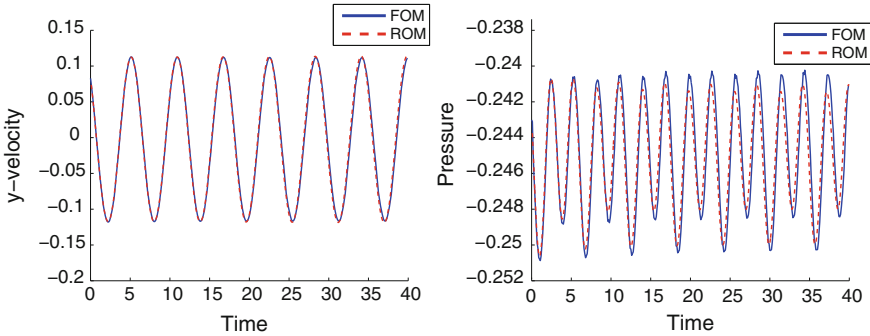


Fig. 8 Comparison of the FOM and ROM velocities at (5,4) for the initial configuration

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 $Re = 100$ based on the dimension of the cylinder and the inflow velocity. A second order backward difference 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 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. 8. 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. 8, 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. 9, 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. Figure 10 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 basis 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 corre-

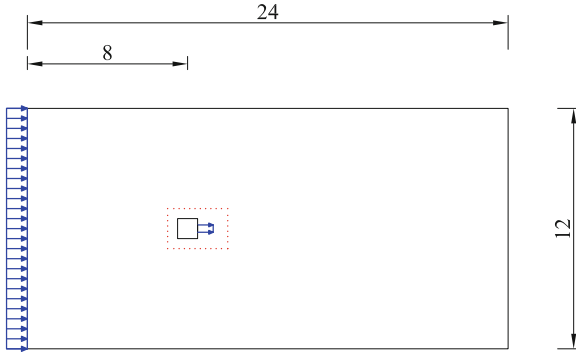


Fig. 9 Flow injection configuration. The red dotted line denotes the FOM domain for the FOM-ROM model

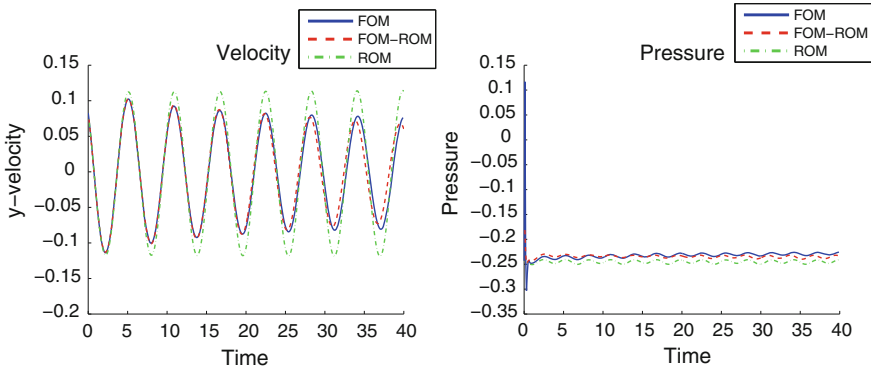


Fig. 10 Comparison of the vertical velocity (*left*) and pressure (*right*) at (5,4) for the FOM, FOM-ROM and ROM models for the injection case

sponds 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 vortices 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. As it will be shown, the combination of both strategies (FOM and ROM) allows us to recover a solution which is close to the full FOM solution, but at a much lower computational cost.

Figure 10 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,

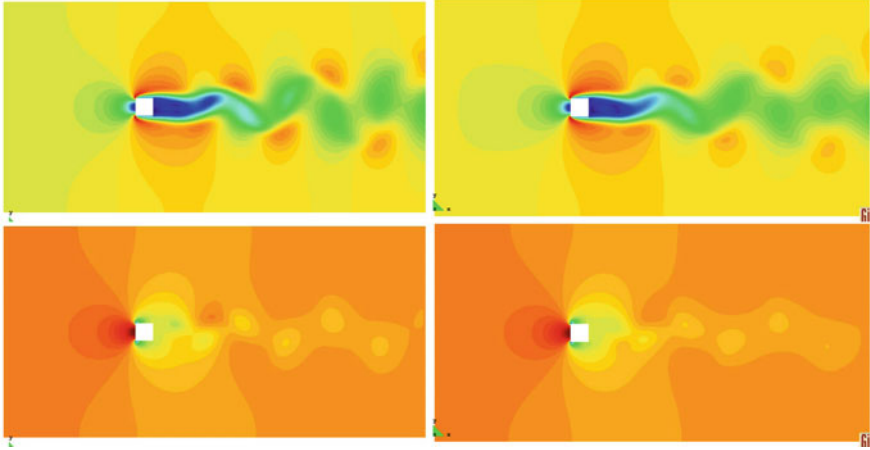


Fig. 11 Comparison of the velocity (*top*) and pressure (*bottom*) fields after 400 steps. *Left* FOM. *Right* FOM-ROM

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). Figure 11 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 vortices 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 of optimality of the snapshot set, the FOM-ROM model is capable of properly representing the solution in the FOM region. Figure 12 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.

5 Conclusions

In this chapter we have discussed several strategies for dealing with the reduced-order approximation of the incompressible Navier-Stokes equations. We have departed from a stabilized finite element full-order approximation and we have approached the order reduction by using a Proper Orthogonal Decomposition (POD) method.

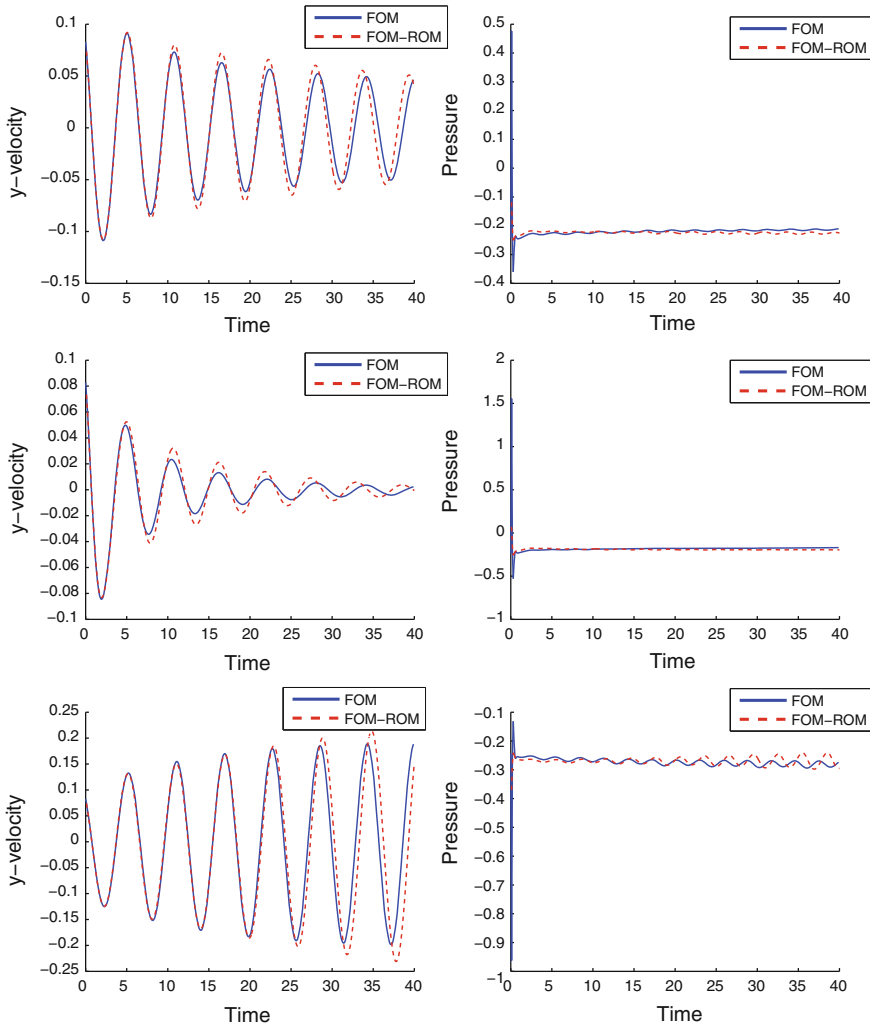


Fig. 12 Comparison of the vertical velocity (*left*) and pressure (*right*) at (5,4) for the FOM, FOM-ROM and ROM models for the injection case. Injection velocities, from *top* to *bottom* 0.2, 0.5, -0.2

In the first part of the chapter, we have focused in the construction of an explicit reduced-order model for the incompressible Navier-Stokes equations, and the application of hyper-reduction techniques to it. The basic idea is to treat all the terms except the mass matrix in the temporal derivative in an explicit way. This includes the non-linear convective term, but also the stabilization terms which can be highly non-linear through the stabilization parameter τ . In order to do so, we take advantage of the fact that the snapshots used for building the reduced-order basis through a singular value decomposition in the POD procedure do already fulfill the stabilized continuity equation. Secondly, we also acknowledge the fact that, if the velocity and

pressure are treated jointly, then the pressure can be recovered from the reduced-order basis and the solution coefficients at the end of each time step.

The proposed explicit reduced-order model performs well in practical cases, as illustrated in the numerical examples section. Despite the time-stepping scheme being explicit, the Courant-Friedrichs-Levy condition can be violated, which can be explained because the reduced basis functions expand over the whole computational domain. On the other hand, the reduced model is sensitive to the inclusion of noisy basis functions, which can cause unstable solutions to appear. The sensitivity of the explicit reduced-order model to this issue can be improved by reducing the time step and refining the finite element mesh.

A hyper-reduction strategy for the explicit reduced-order model has also been presented, which is based on the reconstruction of the right-hand-side vector through a gappy-pod procedure. For the selection of the indices of the gappy reconstruction, we use a discrete version of the Best Points Interpolation Method (DBPIM), which uses only values at the nodes of the finite element mesh, with the advantage that the selected points can be guaranteed to be at least locally optimal.

In the second part of the chapter, we have presented a domain decomposition strategy for non-linear hyper-reduced-order models. The method consists of restricting the reduced-order basis functions to the nodes of each subdomain. This definition of the partitioned problem directly ensures the continuity of the recovered solution. The local POD bases are obtained by computing a local POD decomposition for the partitioned snapshots. When applied to the explicit reduced-order model for the incompressible Navier-Stokes equations a stabilizing penalization term is required. This penalty term is defined so that it weakly enforces the equality of the unknown between subdomains in an overlapping region.

The domain decomposition reduced-order model can be extended to a particular case, in which one of the subdomains is solved by using the full-order finite element equations while the other ones are solved using the reduced-order model. This diminishes the computational cost in the low-resolution subdomains, while keeping the high fidelity solution in the domain regions which are subject to more complex physical phenomena.

Numerical examples illustrate the accuracy of the proposed methods for the solution of incompressible flow problems at a low computational cost: the reduced order-model allows us to save up to 65 % of the computational cost, while in the case of the hyper-reduced order models the computational saving is larger than 99 % the original computational cost.

References

1. Akhtar I, Borggaard J, Hay A (2010) Shape sensitivity analysis in flow models using a finite-difference approach. *Math Probl Eng* 1–23:2010
2. Antil H, Heinkenschloss M, Hoppe RHW, Sorensen DC (2010) Domain decomposition and model reduction for the numerical solution of pde constrained optimization problems with

- localized optimization variables. *Comput Vis Sci* 13(6):249–264
3. Arian E, Fahl M, Sachs EW (2000). Trust-Region proper orthogonal decomposition for flow control. Institute for computers, pp 2000–2101
 4. Astrid P (2004). Reduction of process simulation models: a proper orthogonal decomposition approach. PhD thesis, Department of Electrical Engineering, Eindhoven University of Technology
 5. Astrid P, Weiland S, Willcox K, Backx T (2008) Missing point estimation in models described by proper orthogonal decomposition. *IEEE Trans Autom Control* 53:2237–2251
 6. Baiges J, Codina R, Idelsohn S (2013) A domain decomposition strategy for reduced order models. Application to the incompressible Navier-Stokes equations. *Comput Meth Appl Mech Eng* 267:23–42
 7. Baiges J, Codina R, Idelsohn S (2013) Explicit Reduced Order Models for the stabilized finite element approximation of the incompressible Navier-Stokes equations. *Int J Numer Meth Fluids* 72:1219–1243
 8. Barrault M, Maday Y, Nguyen NC, Patera AT (2004) An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* 339(9):667–672
 9. Bergmann M, Cordier L, Brancher JP (2007) Drag minimization of the cylinder wake by trust-region proper orthogonal decomposition. *Notes on numerical fluid mechanics and multi-disciplinary design* 95:19
 10. Buffoni M, Telib H, Iollo A (2009) Iterative methods for model reduction by domain decomposition. *Comput Fluids* 38(6):1160–1167
 11. Bui-Thanh T, Willcox K, Ghattas O (2008) Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM J Sci Comput* 30(6):3270
 12. Burkardt J, Gunzburger M, Lee H (2006) POD and CVT-based reduced-order modeling of Navier-Stokes flows. *Comput Meth Appl Mech Eng* 196(1–3):337–355
 13. Carlberg K, Bou-Mosleh C, Farhat C (2011) Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *Int J Numer Meth Eng* 86(2):155–181
 14. Chatterjee A (2000) An introduction to the proper orthogonal decomposition. *Curr Sci* 78(7):808–817
 15. Chaturantabut S, Sorensen DC (2009). Discrete empirical interpolation for nonlinear model reduction. Technical Report TR09-05, Rice University, Houston, Texas
 16. Codina R (2001) A stabilized finite element method for generalized stationary incompressible flows. *Comput Meth Appl Mech Eng* 190:2681–2706
 17. Drohmann M, Haasdonk B, Ohlberger M (2012) Reduced basis approximation for nonlinear parameterized evolution equations based on empirical operator interpolation. *SIAM J Sci Comput* 34:937–962
 18. Everson R, Sirovich L (1995) Karhunen-Loève procedure for gappy data. *J Opt Soc Am A* 12:1657–1664
 19. Galletti B, Bruneau CH, Zannetti L, Iollo A (2004) Low-order modelling of laminar flow regimes past a confined square cylinder. *J Fluid Mech* 503:161–170
 20. Glaz B, Liu L, Friedmann PP (2010) Reduced-Order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework. *AIAA J* 48(10):2418–2429
 21. Graham WR, Peraire J, Tang KY (1999) Optimal control of vortex shedding using low-order models. Part i-open-loop model development. *Int J Numer Meth Eng* 44(7):945–972
 22. Grepl MA, Maday Y, Nguyen NC, Patera AT (2007) Efficient Reduced-Basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM. Math Model Numer Anal* 41(03):575–605
 23. Holmes P, Lumley JL, Berkooz G (1998) Turbulence, coherent structures. Dynamical systems and symmetry. Cambridge University Press, New York
 24. Jacobs EN, Ward KE, Pinkerton RM (1933). The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. NACA report, 460

25. Kalashnikova I, Barone MF (2011). Stable and efficient galerkin reduced order models for non-linear fluid flow. In: AIAA-2011-3110, 6th AIAA theoretical fluid mechanics conference, Honolulu
26. Kosambi DD (1943) Statistics in function space. *J Indian Math Soc* 7:76–88
27. Lassila T, Rozza G (2010) Parametric free-form shape design with PDE models and reduced basis method. *Comput Meth Appl Mech Eng* 199(23–24):1583–1592
28. LeGresley PA (2005). Application of proper orthogonal decomposition to design decomposition methods. PhD thesis, Department of Aeronautics and Astronautics, Stanford University
29. Lucia DJ, Beran PS (2003) Projection methods for reduced order models of compressible flows. *J Comput Phys* 188(1):252–280
30. Lucia DJ, King PI, Beran PS (2003) Reduced order modeling of a two-dimensional flow with moving shocks. *Comput Fluids* 32(7):917–938
31. Nguyen NC, Peraire J (2008) An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *Int J Numer Meth Eng* 76(1):27–55
32. Noack BR, Morzynski M, Tadmor G (2011) *Reduced-Order modelling for flow control*. Springer, Berlin
33. Rabczuk T, Bordas SPA, Kerfriden P, Gouy O (2012). A partitioned model order reduction approach to rationalise computational expenses in multiscale fracture mechanics
34. Rozza G, Lassila T, Manzoni A (2011) Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. In: Hesthaven JS, Ranquist EM (eds) *Spectral and high order methods for partial differential equations.*, vol 76Springer, Berlin, pp 307–315
35. Ryckelynck D (2005) A priori hyperreduction method: an adaptive approach. *J Comput Phys* 202(1):346–366
36. Ryckelynck D (2009) Hyper-reduction of mechanical models involving internal variables. *Int J Numer Meth Eng* 77(1):75–89
37. Verhoeven A, Voss T, Astrid P, ter Maten EJW, Bechtold T (2007) Model order reduction for nonlinear problems in circuit simulation. *PAMM* 7(1):1021603–1021604
38. Verhoeven A, Maten J, Striebel M, Mattheij R (2009) Model order reduction for nonlinear ic models. In: Korytowski A, Malanowski K, Mitkowski W, Szymkat M (eds) *System modeling and optimization*, vol 312, IFIP advances in information and communication technology, Springer, Berlin, pp 476–491
39. Veroy K, Patera AT (2005). Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Int J Numer Meth Fluids*, 47(8–9):773–788
40. Wang Z, Akhtar I, Borggaard J, Iliescu T (2011). Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. [arXiv:1106.3585](https://arxiv.org/abs/1106.3585)
41. Wicke M, Stanton M, Treuille A. Modular bases for fluid dynamics. *ACM Trans Graph*, 28(3):39:1–39:8