Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 368 (2020) 113159
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Element boundary terms in reduced order models for flow problems: Domain decomposition and adaptive coarse mesh hyper-reduction

Ricardo Reyes[a], Ramon Codina[a,b,*]

[a] *Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona, Spain*
[b] *International Centre for Numerical Methods in Engineering (CIMNE), C/ Gran Capità S/N, 08034 Barcelona, Spain*

Received 1 December 2019; received in revised form 15 May 2020; accepted 18 May 2020
Available online xxxx

Abstract

In this paper we present a finite-element based reduced order model and, in particular, we consider two aspects related to the introduction of inter-element boundary terms in the formulation. The first is a domain decomposition strategy in which the transmission conditions involve boundary terms to account for non-matching meshes and discontinuous physical properties. The second is a coarse mesh hyper-reduction for which we propose an adaptive refinement driven by an *a posteriori* error estimator that contains element boundary terms. As the finite element full order model, the reduced order model is based on the Variational Multi-Scale framework, with sub-grid scales defined not only in the element interiors, but also on the inter-element boundaries. We present some examples of application using the incompressible Navier–Stokes equations and the Boussinesq approximation.

Keywords: Reduced order model (ROM); Variational multi-scale (VMS) method; Boundary subscales; Hyper-reduction; Adaptive mesh refinement (AMR); *a posteriori error estimates*

1. Introduction

In this work we extend the projection-based Variational Multi-Scale (VMS)-Reduced Order Model (ROM) formulation presented in [1], based on a VMS-Finite Element (FE) formulation as a Full Order Model (FOM), by including some inter-element boundary terms. These terms serve two purposes. Firstly, they may come from the approximation of the boundary subscales, or sub-grid scales (SGS), presented in [2], and in this sense they can be used in particular in *a posteriori* error estimators to be used in an Adaptive Mesh Refinement (AMR) procedure. In this work we apply this AMR to a coarse mesh that is employed in the hyper-reduction step. Secondly, we exploit a domain decomposition strategy for ROM discussed in [3], now considering a general case of non-matching meshes and discontinuous physical properties treated using a traditional Neumann–Dirichlet approach and the Discontinuous Galerkin (DG) approach proposed in [4].

* Corresponding author at: Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona, Spain.
E-mail addresses: ricardo.reyes@upc.edu (R. Reyes), ramon.codina@upc.edu (R. Codina).

https://doi.org/10.1016/j.cma.2020.113159
0045-7825/

Domain decomposition. In FE problems, when a domain decomposition strategy is used to couple two (or more) subdomains, either for homogeneous or heterogeneous problems, one needs to carefully design the transmission conditions. First, if the interpolation is assumed to be discontinuous across subdomains one has to introduce terms to account for this discontinuity, in the spirit of the DG method (see [5,6]). Furthermore, if an iteration-by-subdomain is employed, the iterative scheme may be made more robust by the introduction of boundary terms that may be understood as subscales on the element boundaries in the VMS context (see [7]). This fact, well understood in the FE context, is extended in this paper to the FE-based ROM approach presented in [1].

Domain decomposition schemes for ROM have been implemented in different areas, as part of hybrid ROM-FOM methods [3,8–10], as part of the physical model (e.g. fluid structure interaction cases [11] or coupled Stokes–Darcy flows [12]) and as part of local ROMs (using local bases computed separately for different regions of the domain [13–16] or as a partitioned global basis [3]).

By reviewing these techniques, we can argue that at least two sources of instability might appear when formulating a domain decomposition ROM: the discontinuities that appear naturally from the transmission conditions in the subdomain interfaces and the discontinuity in the local bases. In the same way as in [1], the instabilities that arise from the problem and space reduction are tackled with the VMS simultaneously; here we aim to solve both problems by adding the boundary SGS, i.e., the components of the solution that cannot be captured by the finite element mesh and that need to be approximated somehow.

In this work we follow a similar domain decomposition strategy as the one presented in [7] for FEs. We focus solely in a non-overlapping iterative method for homogeneous and heterogeneous boundaries using local bases.

Error estimators and mesh adaptivity techniques. Adaptive techniques for ROMs can be implemented in both the offline and online parts of the simulation. In the offline part the adaptive techniques have been used in the high fidelity data collection, where the use of AMR techniques increases the accuracy and quality of the input data [17–19] at a lower computational cost when the FOM is based on a mesh-based method.

Less straightforward is the application of adaptivity to the ROM. First, one could consider adaptivity based on the error associated to the basis computed in the offline stage. This adaptivity, in turn, can be of two types: the one consisting in an adaptive selection of the number of basis functions and, for mesh-based ROMs, the one associated with the spatial error of the basis functions. Examples of the former are the basis enrichment methods presented in [20–22], whereas h-adaptivity in the calculation of the basis is formulated in [23,24].

The second application of adaptivity in mesh-based ROMs is the one that we elaborate in this paper, that is related to the construction of a hyper-ROM. In nonlinear problems it is known that the construction of the ROM matrices is as expensive as that of the FOM ones, due to the dependence of these matrices on the unknowns. The idea of the hyper-ROM proposed in [1] is to evaluate nonlinear terms on meshes coarser than the FOM one, trying not to increase the error of the ROM. Mesh adaptivity can now be applied to the construction of this coarse mesh. While this can be done once in the offline stage, as proposed in [25], the method we propose in this paper is to carry out a genuine AMR in the ROM stage, and thus to compute the error of the (mesh-based) ROM solution and perform an AMR according to this error.

Besides the refinement algorithm itself, the second key ingredient of AMR techniques is the definition of the error estimators that allow one to decide over which domain region mesh refinement is required. The use of residual-based error estimators in FE problems —including interior and boundary subscales— is vast in the literature (see for example [26–29] and references within). This type of error estimators can become useful in the ROM context, since they are explicit and therefore do not require solving additional equations.

In this work we follow an approach similar to [30], where the error estimator is defined using dynamic Orthogonal Sub-Grid Scales (OSGS) and including the definition of boundary subscales presented in [2]. As a refinement algorithm we use solely the hierarchical h-AMR strategy presented in [31].

Paper organization. The paper is organized as follows. In Section 2 we describe a general (nonlinear) convection–diffusion–reaction problem. In Section 3 we describe the stabilized VMS method used to stabilize both the FOM and the ROM formulations, where the formulation includes SGS in the interior and in the boundaries of the elements. In Section 4 we describe the domain decomposition problem and how to include boundary terms to account for discontinuous interpolations between subdomains and boundary SGS to stabilize the transmission conditions at the interface. In Section 5 we describe the use of the SGS as an error indicator, which we use in an AMR that serves to design a mesh-based hyper-ROM formulation. In Section 6 we present numerical results that test the formulation,

including a flow past a cylinder and a backward facing step using the incompressible Navier–Stokes equation, and two differentially heated cavities using the Boussinesq approximation. And finally, in Section 7 we present some concluding remarks.

2. Continuous problem

Let us start by writing a general convection–diffusion–reaction problem posed in a spatial domain $\Omega \subset \mathbb{R}^d$ with a boundary Γ and a time interval from zero to a final time t_f, that consists in finding a vector function $\boldsymbol{y}(\boldsymbol{x}, t)$ of n components such that

$$\boldsymbol{\mathcal{M}}(\boldsymbol{y})\partial_t \boldsymbol{y} + \boldsymbol{\mathcal{L}}(\boldsymbol{y}; \boldsymbol{y}) = \boldsymbol{f}, \quad \text{in } \Omega,\ t \in]0, t_\mathrm{f}[, \tag{1}$$

where d is the number of space dimensions, $\boldsymbol{\mathcal{M}}(\boldsymbol{y})$ a mass matrix, $\boldsymbol{f}(\boldsymbol{x}, t)$ a forcing term, ∂_t the derivative over time, and $\boldsymbol{\mathcal{L}}$ a nonlinear differential operator in space of first or second order defined as $\boldsymbol{\mathcal{L}}(\boldsymbol{y}; \boldsymbol{z}) := \boldsymbol{A}_i^\mathrm{c}(\boldsymbol{y})\partial_i \boldsymbol{z} + \boldsymbol{A}_i^\mathrm{f}(\boldsymbol{y})\partial_i \boldsymbol{z} - \partial_i\big(\boldsymbol{K}_{ij}(\boldsymbol{y})\partial_j \boldsymbol{z}\big) + \boldsymbol{S}(\boldsymbol{y})\boldsymbol{z}$. Here, $\boldsymbol{A}_i^\mathrm{c}(\boldsymbol{y})$, $\boldsymbol{A}_i^\mathrm{f}(\boldsymbol{y})$, $\boldsymbol{K}_{ij}(\boldsymbol{y})$ and $\boldsymbol{S}(\boldsymbol{y})$ are $n \times n$ matrices and ∂_i denotes differentiation with respect to the ith Cartesian coordinate x_i. Indexes i, j run from 1 to d and repeated indexes imply summation. Additionally, if $\Gamma = \Gamma_D \cup \Gamma_N$, the initial and boundary conditions are set as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{y}_0 && \text{in } \Omega, && t = 0, \\ \boldsymbol{\mathcal{D}}(\boldsymbol{y}) &= \boldsymbol{\mathcal{D}}(\boldsymbol{y}_D) && \text{on } \Gamma_D, && t \in]0, t_\mathrm{f}[, \\ \boldsymbol{\mathcal{F}}(\boldsymbol{y}; \boldsymbol{y}) &= \boldsymbol{t}_N && \text{on } \Gamma_N, && t \in]0, t_\mathrm{f}[, \end{aligned}$$

where $\boldsymbol{y}_0$ is the prescribed initial condition, $\boldsymbol{y}_D$ the prescribed Dirichlet boundary condition, $\boldsymbol{\mathcal{D}}$ is the Dirichlet operator associated to $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{F}}(\boldsymbol{y}; \boldsymbol{z}) := n_i \boldsymbol{K}_{ij}(\boldsymbol{y})\partial_j \boldsymbol{z} - n_i \boldsymbol{A}_i^\mathrm{f}(\boldsymbol{y})\boldsymbol{z}$ is a flux operator, $\boldsymbol{n}$ the normal to Γ and $\boldsymbol{t}_N$ the prescribed Neumann boundary condition.

Defining $\mathscr{Y}$ as the space of functions in Ω where $\boldsymbol{y}$ is sought for each time t, satisfying $\boldsymbol{\mathcal{D}}(\boldsymbol{y}) = \boldsymbol{\mathcal{D}}(\boldsymbol{y}_D)$ on Γ_D, $\mathscr{Y}_0$ as its corresponding space of time independent test functions that satisfy $\boldsymbol{\mathcal{D}}\boldsymbol{v} = \boldsymbol{0}$ on Γ_D, $\langle \cdot, \cdot \rangle$ as an appropriate duality pairing, and $(\cdot, \cdot)$ as the L^2-inner product in Ω, we can define the variational form of the problem as finding $\boldsymbol{y} :]0, t_\mathrm{f}[\to \mathscr{Y}$ such that

$$\begin{aligned} (\boldsymbol{\mathcal{M}}(\boldsymbol{y})\partial_t \boldsymbol{y}, \boldsymbol{v}) + \langle \boldsymbol{\mathcal{L}}(\boldsymbol{y}; \boldsymbol{y}), \boldsymbol{v} \rangle &= \langle \boldsymbol{f}, \boldsymbol{v} \rangle, && \forall \boldsymbol{v} \in \mathscr{Y}_0, \quad t \in]0, t_\mathrm{f}[, \\ (\boldsymbol{y}, \boldsymbol{v}) &= (\boldsymbol{y}_0, \boldsymbol{v}), && \forall \boldsymbol{v} \in \mathscr{Y}_0 \quad t = 0. \end{aligned} \tag{2}$$

Denoting by $\langle \cdot, \cdot \rangle_{\Gamma_N}$ the integral of the product of two functions defined in Γ_N and defining the forms $B(\boldsymbol{y}; \boldsymbol{z}, \boldsymbol{v}) = \langle \boldsymbol{A}_i^\mathrm{c}(\boldsymbol{y})\partial_i \boldsymbol{z}, \boldsymbol{v} \rangle - \langle \boldsymbol{z}, \partial_i(\boldsymbol{A}_i^\mathrm{f}(\boldsymbol{y})^\top \boldsymbol{v}) \rangle + \langle \boldsymbol{S}(\boldsymbol{y})\boldsymbol{z}, \boldsymbol{v} \rangle + \langle \boldsymbol{K}_{ij}(\boldsymbol{y})\partial_j \boldsymbol{z}, \partial_i \boldsymbol{v} \rangle$, and $L(\boldsymbol{v}) = \langle \boldsymbol{f}, \boldsymbol{v} \rangle + \langle \boldsymbol{t}_N, \boldsymbol{v} \rangle_{\Gamma_N}$, we can write the equivalent problem: find a vector function $\boldsymbol{y} :]0, t_\mathrm{f}[\to \mathscr{Y}$ such that

$$\begin{aligned} (\boldsymbol{\mathcal{M}}(\boldsymbol{y})\partial_t \boldsymbol{y}, \boldsymbol{v}) + B(\boldsymbol{y}; \boldsymbol{y}, \boldsymbol{v}) &= L(\boldsymbol{v}), && \forall \boldsymbol{v} \in \mathscr{Y}_0, \quad t \in]0, t_\mathrm{f}[, \\ (\boldsymbol{y}, \boldsymbol{v}) &= (\boldsymbol{y}_0, \boldsymbol{v}), && \forall \boldsymbol{v} \in \mathscr{Y}_0 \quad t = 0. \end{aligned} \tag{3}$$

3. VMS formulation

3.1. Scale splitting

The VMS stabilization described here for both FOM and ROM cases follows the formulations presented in [32,33] for a FE problem and in [1] for a projection-based stabilized ROM. In this work we describe exclusively the formulation for the ROM using the subscript r for the resolved scales and the superscript $\check{}$ for the subscales. The equations for the FOM can be written in the same way and therefore we omit them. Furthermore, the model reduction approach and the basis construction using Proper Orthogonal Decomposition (POD) are the same as those used in [1,34].

To formulate the discrete ROM version of the variational problem, we denote $\mathscr{T}_h = \{K\}$ a FE partition of the domain Ω, with $h = \max\{h_K | K \in \mathscr{T}_h\}$ the diameter of the partition. If $\mathscr{Y}_h \subset \mathscr{Y}$ is the finite element space to construct the FOM, we denote by $\mathscr{Y}_r \subset \mathscr{Y}_h$ the ROM approximation space.

For the time discretization we consider a uniform partition of the time interval $]0, t_\mathrm{f}[$, with δt the time step size and j the time step counter. The temporal derivatives are approximated using finite differences as $\partial_t \boldsymbol{y} \,|_{tj} \approx \delta_t \boldsymbol{y}^j$,

where $\delta_t \boldsymbol{y}^j$ indicates a finite difference operator at t^j. If there is no possibility of confusion, the superscript j will be omitted.

We may write the time discrete version of the problem as: find $\{\boldsymbol{y}^j\} \subset \mathscr{Y}$, such that

$$(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}, \boldsymbol{v}) + B(\boldsymbol{y}; \boldsymbol{y}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in \mathscr{Y}, \text{ at } t^j, \ j = 1, 2, \ldots. \tag{4}$$

The VMS method consists in decomposing the space of the unknown into the finite-dimensional space $\mathscr{Y}_r$, and a continuous one, $\breve{\mathscr{Y}}$, so that $\mathscr{Y} = \mathscr{Y}_r \oplus \breve{\mathscr{Y}}$. The unknown and the test functions are accordingly split as $\boldsymbol{y} = \boldsymbol{y}_r + \breve{\boldsymbol{y}}$ and $\boldsymbol{v} = \boldsymbol{v}_r + \breve{\boldsymbol{v}}$, respectively. Then, the problem in Eq. (4) expands into: find $\{\boldsymbol{y}_r^j\} \subset \mathscr{Y}_r$ and $\{\breve{\boldsymbol{y}}^j\} \subset \breve{\mathscr{Y}}$, such that

$$\begin{aligned}
&(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r, \boldsymbol{v}_r) + (\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \breve{\boldsymbol{y}}, \boldsymbol{v}_r) + B\left(\boldsymbol{y}; \boldsymbol{y}_r, \boldsymbol{v}_r\right) \\
&\qquad + B(\boldsymbol{y}; \breve{\boldsymbol{y}}, \boldsymbol{v}_r) = L(\boldsymbol{v}_r), \quad \forall \boldsymbol{v}_r \in \mathscr{Y}_r, \text{ at } t^j, \ j = 1, 2, \ldots, \qquad (5a) \\
&(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r, \breve{\boldsymbol{v}}) + (\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \breve{\boldsymbol{y}}, \breve{\boldsymbol{v}}) + B\left(\boldsymbol{y}; \boldsymbol{y}_r, \breve{\boldsymbol{v}}\right) \\
&\qquad + B(\boldsymbol{y}; \breve{\boldsymbol{y}}, \breve{\boldsymbol{v}}) = L(\breve{\boldsymbol{v}}), \quad \forall \breve{\boldsymbol{v}} \in \breve{\mathscr{Y}}, \text{ at } t^j, \ j = 1, 2, \ldots. \qquad (5b)
\end{aligned}$$

Note that we have left $\boldsymbol{y}$ in the nonlinear dependencies; it could be replaced by $\boldsymbol{y}_r + \breve{\boldsymbol{y}}$ or simply approximated by $\boldsymbol{y}_r$.

Now, following the standard procedure in VMS, doing an integration by parts in the bilinear forms $B(\boldsymbol{y}; \breve{\boldsymbol{y}}, \boldsymbol{v}_r)$, $B\left(\boldsymbol{y}; \boldsymbol{y}_r, \breve{\boldsymbol{v}}\right)$ and $B(\boldsymbol{y}; \breve{\boldsymbol{y}}, \breve{\boldsymbol{v}})$ for $\boldsymbol{y}$ fixed, we may rewrite Eq. (5) as

$$\begin{aligned}
&(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r, \boldsymbol{v}_r) + (\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \breve{\boldsymbol{y}}, \boldsymbol{v}_r) + B\left(\boldsymbol{y}; \boldsymbol{y}_r, \boldsymbol{v}_r\right) + \sum_K \langle \breve{\boldsymbol{y}}, \boldsymbol{\mathcal{L}}^*(\boldsymbol{y}; \boldsymbol{v}_r)\rangle_K \\
&\qquad + \sum_K \langle \breve{\boldsymbol{y}}_E, \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}; \boldsymbol{v}_r)\rangle_{\partial K} = L(\boldsymbol{v}_r), \qquad \forall \boldsymbol{v}_r \in \mathscr{Y}_r, \text{ at } t^j, \ j = 1, 2, \ldots, \qquad (6a) \\
&\sum_K \langle \boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \breve{\boldsymbol{y}}, \breve{\boldsymbol{v}}\rangle_K + \sum_K \langle \boldsymbol{\mathcal{L}}(\boldsymbol{y}, \breve{\boldsymbol{y}}), \breve{\boldsymbol{v}}\rangle_K = \sum_K \langle \boldsymbol{r}(\boldsymbol{y}; \boldsymbol{y}_r), \breve{\boldsymbol{v}}\rangle_K, \qquad \forall \breve{\boldsymbol{v}} \in \breve{\mathscr{Y}}, \text{ at } t^j, \ j = 1, 2, \ldots. \qquad (6b)
\end{aligned}$$

where we have introduced the operators

$$\begin{aligned}
&\boldsymbol{\mathcal{L}}^*(\boldsymbol{y}; \boldsymbol{v}_r) := -\partial_i\left(\boldsymbol{K}_{ji}(\boldsymbol{y})^\top \partial_j \boldsymbol{v}_r\right) - \partial_i \boldsymbol{A}_i^{\mathrm{c}}(\boldsymbol{y})^\top \boldsymbol{v}_r - \partial_i \boldsymbol{A}_i^{\mathrm{f}}(\boldsymbol{y})^\top \boldsymbol{v}_r + \boldsymbol{S}(\boldsymbol{y})^\top \boldsymbol{v}_r, \\
&\boldsymbol{\mathcal{F}}^*(\boldsymbol{y}; \boldsymbol{v}_r) := n_i \boldsymbol{K}_{ji}(\boldsymbol{y})^\top \partial_j \boldsymbol{v}_r + n_i \boldsymbol{A}_i^{c\top}(\boldsymbol{y}) \boldsymbol{v}_r, \\
&\boldsymbol{r}(\boldsymbol{y}; \boldsymbol{y}_r) := \boldsymbol{f} - \boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r - \boldsymbol{\mathcal{L}}(\boldsymbol{y}; \boldsymbol{y}_r),
\end{aligned}$$

and the integral of the product of two functions over an element $K \in \mathscr{T}_h$ and its boundary have been written as $\langle \cdot, \cdot \rangle_K$ and $\langle \cdot, \cdot \rangle_{\partial K}$, respectively.

In Eq. (6a) we have introduced the value of the subscales on the element boundaries, $\breve{\boldsymbol{y}}_E$. We will assume that this function is uniquely defined on the element edges, i.e., it does not depend on the element that has a given edge, and it will be approximated independently from the subscales in the element interiors.

3.2. Subscales in the element interiors

As justified in [1] and references therein, to solve the subscale problem we use the approximation $\boldsymbol{\mathcal{L}}(\boldsymbol{y}; \breve{\boldsymbol{y}}) \approx \boldsymbol{\tau}_K^{-1}(\boldsymbol{y})\breve{\boldsymbol{y}}$ in each element K, where $\boldsymbol{\tau}_K$ is the matrix of stabilization parameters, assumed to be symmetric. This way we can write the equation for the subscales as

$$\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \breve{\boldsymbol{y}} + \boldsymbol{\tau}_K^{-1}(\boldsymbol{y})\breve{\boldsymbol{y}} = \breve{\boldsymbol{\Pi}}(\boldsymbol{r}(\boldsymbol{y}; \boldsymbol{y}_r)), \qquad \text{in } K \in \mathscr{T}_h, \quad \text{at } t^j, \ j = 1, 2, \ldots, \tag{7}$$

where $\breve{\boldsymbol{\Pi}} = \boldsymbol{I} - \breve{\boldsymbol{\Pi}}^\perp$, $\boldsymbol{I}$ is the identity matrix and $\breve{\boldsymbol{\Pi}}^\perp$ is the projection onto the space orthogonal to $\breve{\mathscr{Y}}$.

Suppose, to fix ideas, that the time integration scheme used for the sub-grid scales is the simplest backwards differentiation. Defining the effective stabilization parameter as $\boldsymbol{\tau}_{K,t}^{-1}(\boldsymbol{y}) := \delta t^{-1}\boldsymbol{\mathcal{M}} + \boldsymbol{\tau}_K^{-1}(\boldsymbol{y})$, we can write Eq. (6a) as

$$\begin{aligned}
&(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r, \boldsymbol{v}_r) + (\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \breve{\boldsymbol{y}}, \boldsymbol{v}_r) + B\left(\boldsymbol{y}; \boldsymbol{y}_r, \boldsymbol{v}_r\right) + \sum_K \langle \breve{\boldsymbol{y}}_E, \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}; \boldsymbol{v}_r)\rangle_{\partial K} \\
&\quad - \sum_K \langle \breve{\boldsymbol{\Pi}}(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r + \boldsymbol{\mathcal{L}}(\boldsymbol{y}; \boldsymbol{y}_r)), \boldsymbol{\tau}_{K,t}(\boldsymbol{y})\boldsymbol{\mathcal{L}}^*(\boldsymbol{y}, \boldsymbol{v}_r)\rangle_K = L(\boldsymbol{y}; \boldsymbol{v}_r) \\
&\quad - \sum_K \langle \breve{\boldsymbol{\Pi}}(\boldsymbol{f}) + \delta t^{-1}\boldsymbol{\mathcal{M}}(\boldsymbol{y})\breve{\boldsymbol{y}}^{j-1}, \boldsymbol{\tau}_{K,t}(\boldsymbol{y})\boldsymbol{\mathcal{L}}^*(\boldsymbol{y}, \boldsymbol{v}_r)\rangle_K, \quad \forall \boldsymbol{v}_r \in \mathscr{Y}_r, \text{ at } t^j, \ j = 1, 2, \ldots. \qquad (8)
\end{aligned}$$

As discussed in [1], we can define the projection $\breve{\boldsymbol{\Pi}}$ depending on the choice of the space $\breve{\mathscr{Y}}$. We can formulate two alternatives for the subscales: the Algebraic Sub-Grid Scale (ASGS) formulation consists in taking $\breve{\boldsymbol{\Pi}} = \boldsymbol{I}$, the identity on residuals $\boldsymbol{r}(\boldsymbol{y};\boldsymbol{y}_r)$, whereas the Orthogonal Sub-Grid Scale (OSGS) formulation consists in taking $\breve{\boldsymbol{\Pi}}$ as the L^2-orthogonal projection with respect to $\boldsymbol{\mathcal{M}}$, i.e. $\breve{\boldsymbol{\Pi}} := \boldsymbol{\Pi}_r^{\perp} = \boldsymbol{I} - \boldsymbol{\Pi}_r$, where $\boldsymbol{\Pi}_r$ is the L^2-projection onto $\mathscr{Y}_r$ (with respect to $\boldsymbol{\mathcal{M}}$). Note that in the OSGS case, $(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t\breve{\boldsymbol{y}}, \boldsymbol{v}_r) = 0$ and $\breve{\boldsymbol{\Pi}}(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t\boldsymbol{y}_r) = 0$ in Eq. (8) if $\boldsymbol{\mathcal{M}}$ is constant (otherwise it can be considered an approximation). In any case, we always consider $\delta_t\boldsymbol{y}_r \neq \boldsymbol{0}$, i.e., dynamic subscales [1,35].

3.3. Subscales on the element boundaries

To approximate the term in Eq. (8) concerning the subscales on the boundaries, $\sum_K \langle \breve{\boldsymbol{y}}_E, \boldsymbol{\mathcal{F}}^*(\boldsymbol{y};\boldsymbol{v}_r)\rangle_{\partial K}$, we follow a simplified version of the approach introduced for the FE problem in [2].

We start by writing the weak continuity of the total fluxes on the element boundaries, used to obtain Eq. (6b), as

$$\boldsymbol{0} = \sum_K \langle \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}), \breve{\boldsymbol{v}}\rangle_{\partial K} = \sum_K \langle \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r), \breve{\boldsymbol{v}}\rangle_{\partial K} + \sum_K \langle \boldsymbol{\mathcal{F}}(\boldsymbol{y};\breve{\boldsymbol{y}}), \breve{\boldsymbol{v}}\rangle_{\partial K}, \quad \forall \breve{\boldsymbol{v}} \in \breve{\mathscr{Y}}, \tag{9}$$

assuming for the moment that there is no Neumann boundary. This equation is a consequence of the fact that the flux operator $\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y})$ has to be continuous across the inter-element boundaries.

Let E be an edge of the finite element partition (face, in 3D), shared by two element domains K_1 and K_2. In the case E is an edge on the boundary, it is understood that K_2 is void. Let $\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)|_{\partial K_i \cap E}$ the flux computed with the normal from K_i, $i = 1, 2$, and define the jump of fluxes as

$$[\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{z})]\!]_E := \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{z})|_{\partial K_1 \cap E} + \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{z})|_{\partial K_2 \cap E}.$$

We could define this term as (twice) the average of fluxes, although we prefer to call it a jump, since the two terms are computed with normals of opposite sense.

The continuity of fluxes that implies Eq. (9) can be written as

$$\boldsymbol{0} = [\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y})]\!]_E = [\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)]\!]_E + [\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\breve{\boldsymbol{y}})]\!]_E, \quad \text{in } E \in \mathscr{T}_h. \tag{10}$$

Now we make the key approximation

$$[\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\breve{\boldsymbol{y}})]\!]_E = \boldsymbol{\tau}_E^{-1}\breve{\boldsymbol{y}}_E, \tag{11}$$

where $\boldsymbol{\tau}_E$ is a matrix of stabilization parameters to be determined and $\breve{\boldsymbol{y}}_E$ is the single valued subscale on E. This yields the model we shall use:

$$\breve{\boldsymbol{y}}_E = -\boldsymbol{\tau}_E[\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)]\!]_E. \tag{12}$$

In the case of edges belonging to Neumann boundaries, $E_N = \partial K \cap \Gamma_N$, Eq. (10) needs to be replaced by

$$\boldsymbol{t}_N = \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)|_{E_N} + \boldsymbol{\mathcal{F}}(\boldsymbol{y};\breve{\boldsymbol{y}})|_{E_N}, \tag{13}$$

and Eq. (12) needs to be replaced by

$$\breve{\boldsymbol{y}}_{E_N} = -\boldsymbol{\tau}_E(\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)|_{E_N} - \boldsymbol{t}_N). \tag{14}$$

Remark. The formulation arising from Eq. (12) is equivalent to the one presented in [2] with the element interior subscales evaluated at the edge E neglected.

3.4. Stabilized formulation

By using orthogonal subscales in the element interiors (OSGS formulation) and the subscales on the element boundaries and the Neumann boundaries given by Eqs. (12) and (14), respectively, we can write the stabilized

VMS-ROM approximation we propose as

$$
\begin{aligned}
&(\boldsymbol{\mathcal{M}}(\boldsymbol{y})\delta_t \boldsymbol{y}_r, \boldsymbol{v}_r) + B\left(\boldsymbol{y}; \boldsymbol{y}_r, \boldsymbol{v}_r\right) - \sum_K \langle \check{\boldsymbol{\Pi}}(\boldsymbol{\mathcal{L}}(\boldsymbol{y}; \boldsymbol{y}_r)), \boldsymbol{\tau}_{K,t}(\boldsymbol{y})\boldsymbol{\mathcal{L}}^*(\boldsymbol{y}, \boldsymbol{v}_r)\rangle_K \\
&\quad - \sum_E \langle [\![\boldsymbol{\mathcal{F}}(\boldsymbol{y}; \boldsymbol{y}_r)]\!], \boldsymbol{\tau}_E [\![\boldsymbol{\mathcal{F}}^*(\boldsymbol{y}; \boldsymbol{v}_r)]\!]\rangle_E - \sum_{E_N} \langle \boldsymbol{\mathcal{F}}(\boldsymbol{y}; \boldsymbol{y}_r), \boldsymbol{\tau}_E \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}; \boldsymbol{v}_r)\rangle_{E_N} \\
&\quad = L(\boldsymbol{y}; \boldsymbol{v}_r) - \sum_K \langle \check{\boldsymbol{\Pi}}(\boldsymbol{f}) + \delta t^{-1} \boldsymbol{\mathcal{M}}(\boldsymbol{y}) \check{\boldsymbol{y}}^{j-1}, \boldsymbol{\tau}_{K,t}(\boldsymbol{y})\boldsymbol{\mathcal{L}}^*(\boldsymbol{y}, \boldsymbol{v}_r)\rangle_K \\
&\quad - \sum_{E_N} \langle \boldsymbol{t}_N, \boldsymbol{\tau}_E \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}; \boldsymbol{v}_r)\rangle_{E_N}, \quad \forall \boldsymbol{v}_r \in \mathscr{V}_r, \quad \text{at } t^j, \ j = 1, 2, \ldots .
\end{aligned} \tag{15}
$$

3.5. Stabilization parameters for advection–diffusion–reaction equations

The formulation presented relies on the choice of the stabilization parameters in the element interiors $\boldsymbol{\tau}_K$ and on the element boundaries $\boldsymbol{\tau}_E$, which are of course problem dependent. Here we consider their expression for the scalar convection–diffusion equation (and thus $\boldsymbol{\tau}_K$ and $\boldsymbol{\tau}_E$ are scalars), in which

$$
\begin{aligned}
\mathcal{L}(y) &= -\nu \Delta y + \boldsymbol{a} \cdot \nabla y + \sigma y, \\
\mathcal{F}(y) &= \nu \boldsymbol{n} \cdot \nabla y.
\end{aligned}
$$

Note that these are now linear operators. In these expressions, $\nu > 0$ is the diffusion coefficient, $\boldsymbol{a}$ the advection velocity, considered constant for simplicity, and $\sigma \geq 0$ the reaction coefficient.

In [1] (see also references therein) it is shown that τ_K can be taken as

$$
\tau_K = \left(c_{1,K} \frac{\nu}{h^2} + c_{2,K} \frac{|\boldsymbol{a}|}{h} + c_{3,K} \sigma \right)^{-1}, \tag{16}
$$

where $c_{1,K}$, $c_{2,K}$ and $c_{3,K}$ are algorithmic constants, that may be taken as $c_{1,K} = 4$, $c_{2,K} = 2$ and $c_{3,K} = 1$ for linear elements. This expression of the stabilization parameter for the ROM is the same as that used for the FOM, in our case a FE method. It can be justified from a Fourier analysis that, when used to motivate Eq. (11), yields

$$
\tau_E = \left(c_E \frac{\nu}{h} \right)^{-1}, \tag{17}
$$

where c_E is another algorithmic constant.

Remark. The parameter τ_E is similar to the one obtained in [2] where it is defined using a direct approximation of the fluxes over the edge. There the constant c_E is associated with the fraction of h in which the subscale on the edge is taken into account inwards the element.

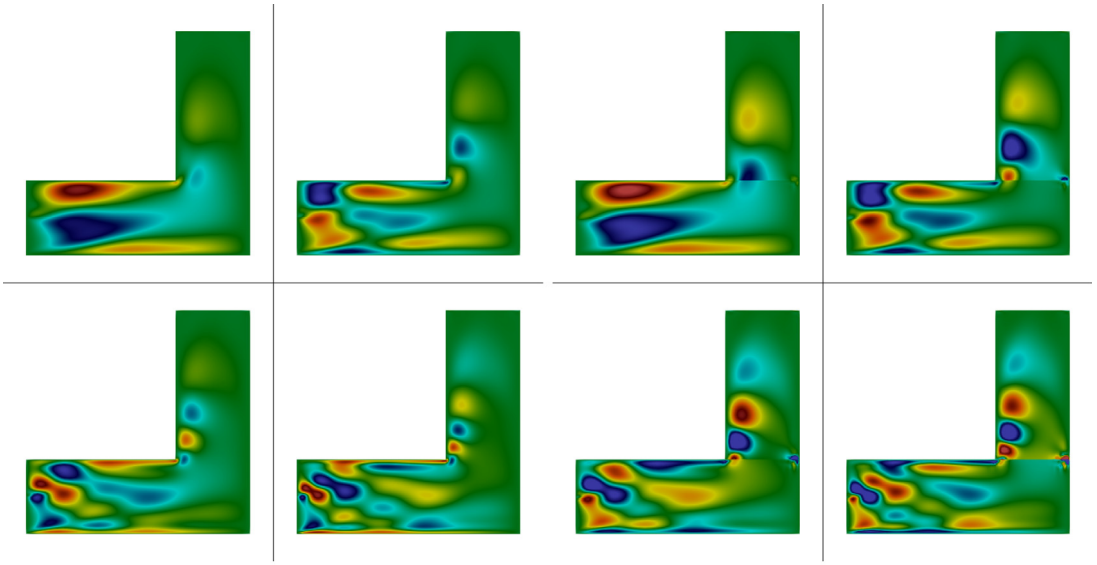
4. Domain decomposition

In this section we implement the stabilized VMS-ROM formulation presented in Section 3 to a non-overlapping domain decomposition problem as done for the FE counterpart in [7]. For simplicity we consider it without Neumann boundary conditions.

4.1. Problem statement

Suppose that the domain Ω is split as $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$, with $\Gamma = \bar{\Omega}_1 \cap \bar{\Omega}_2$. Considering the fluxes in Γ as unknowns and enforcing continuity weakly, we can write the problem as finding $\boldsymbol{y}_1$, $\boldsymbol{y}_2$ and $\boldsymbol{t}_\Gamma$ such that

$$
\begin{aligned}
\boldsymbol{\mathcal{M}}_1(\boldsymbol{y}_1)\partial_t \boldsymbol{y}_1 + \boldsymbol{\mathcal{L}}_1(\boldsymbol{y}_1; \boldsymbol{y}_1) &= \boldsymbol{f}, && \text{in } \Omega_1, && t \in]0, t_\mathrm{f}[, \\
\boldsymbol{\mathcal{D}}(\boldsymbol{y}_1) &= \boldsymbol{\mathcal{D}}(\boldsymbol{y}_{D_1}) && \text{on } \Gamma_{D_1}, && t \in]0, t_\mathrm{f}[, \\
\boldsymbol{\mathcal{M}}_2(\boldsymbol{y}_2)\partial_t \boldsymbol{y}_2 + \boldsymbol{\mathcal{L}}_2(\boldsymbol{y}_2; \boldsymbol{y}_2) &= \boldsymbol{f}, && \text{in } \Omega_2, && t \in]0, t_\mathrm{f}[, \\
\boldsymbol{\mathcal{D}}(\boldsymbol{y}_2) &= \boldsymbol{\mathcal{D}}(\boldsymbol{y}_{D_2}) && \text{on } \Gamma_{D_2}, && t \in]0, t_\mathrm{f}[,
\end{aligned}
$$



(a) x-Velocity global basis. (b) x-Velocity local bases.

Fig. 1. Modes 2 to 5 of the global and local bases obtained from an incompressible Navier–Stokes L-shaped flow problem.

$$\mathcal{D}(\boldsymbol{y}_1) = \mathcal{D}(\boldsymbol{y}_2) \qquad \text{on } \Gamma, \qquad t \in]0, t_f[, \tag{18a}$$

$$\mathcal{F}_1(\boldsymbol{y}_1; \boldsymbol{y}_1) = \boldsymbol{t}_\Gamma \qquad \text{on } \Gamma, \qquad t \in]0, t_f[, \tag{18b}$$

$$\mathcal{F}_2(\boldsymbol{y}_2; \boldsymbol{y}_2) = -\boldsymbol{t}_\Gamma \qquad \text{on } \Gamma, \qquad t \in]0, t_f[, \tag{18c}$$

where Γ_{D_i} is the boundary of Ω_i excluding Γ with Dirichlet data $\boldsymbol{y}_{D_i}$, $i = 1, 2$. Variables and operators with subscript i correspond to their global counterpart restricted to Ω_i. Obviously, what follows can be easily extended to several interacting subdomains.

Let us remark that the ROM basis can be computed globally or locally, independently for each subdomain (see [3]). Although it is evident that the local approach yields discontinuous bases, the interpolation of the global approach may also be discontinuous if different degrees of freedom are taken at each side of the interface.

To illustrate the discontinuity in the basis when it is calculated in a local way —one basis per subdomain, in Fig. 1 we show a comparison between the x-velocity bases obtained using a global domain and local subdomains in an incompressible Navier–Stokes L-shaped flow problem (Section 6.1.1).

4.2. Treatment of subdomain interface

The coupling between the two subdomains can be approached using different methods; here, we solve the domain decomposition problem using two techniques: the standard Neumann–Dirichlet coupling and a less common DG approach, as indicated earlier.

4.2.1. Standard Neumann-Dirichlet approach

As suggested by the transmission conditions in Eqs. (18a) to (18c), the Neumann–Dirichlet method consists in setting a Neumann boundary condition on one subdomain and a Dirichlet boundary condition on the other. Suppose that the Dirichlet boundary condition has to be applied to the ith subdomain. In this case, the space of trial functions in this domain will be made of functions satisfying the Dirichlet condition coming from the jth domain in a strong manner, and the space of test functions will satisfy the homogeneous counterpart of this condition. In turn, when solving in the jth subdomain, the term

$$-\langle \mathcal{F}(\boldsymbol{y}_i; \boldsymbol{y}_i), \mathcal{D}(\boldsymbol{v}_j) \rangle_\Gamma$$

will appear in the right-hand-side of the variational equation for this subdomain.

4.2.2. Discontinuous Galerkin approach

For the DG case, we consider the problem posed in the whole domain Ω (Eq. (3)) with a discontinuity in the interpolation of the unknowns across the interface Γ. To account for this discontinuity, we introduce the terms derived in [4] for DG methods (see also [5,6] for similar formulations) to our stabilized ROM formulation.

Let us introduce some notation. We define $\boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{z})$ as the array of components $\mathcal{H}_i(\boldsymbol{y};\boldsymbol{z}) := \boldsymbol{K}_{ij}(\boldsymbol{y})\partial_j \boldsymbol{z} - \boldsymbol{A}_i^{\mathrm{f}}(\boldsymbol{y})\boldsymbol{z}$, and thus $\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{z}) = n_i\mathcal{H}_i(\boldsymbol{y};\boldsymbol{z})$; the adjoint array $\boldsymbol{\mathcal{H}}^*(\boldsymbol{y};\boldsymbol{z})$ is defined analogously. We also introduce the array $\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})$ of components $\mathcal{D}_{n,i}(\boldsymbol{y}) = n_i\boldsymbol{\mathcal{D}}(\boldsymbol{y})$. On the interface Γ we define the jump $[\![\cdot]\!]$ and average $\{\cdot\}$ operators

$$[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})]\!] := \boldsymbol{\mathcal{D}}_n(\boldsymbol{y})\big|_{\partial\Omega_1\cap\Gamma} + \boldsymbol{\mathcal{D}}_n(\boldsymbol{y})\big|_{\partial\Omega_2\cap\Gamma},$$
$$\{\boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{z})\} := \frac{1}{2}\big(\boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{z})\big|_{\partial\Omega_1\cap\Gamma} + \boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{z})\big|_{\partial\Omega_2\cap\Gamma}\big).$$

Again, calling 'jump' the first term is just because it involves the normals, and $\boldsymbol{n}_1 = -\boldsymbol{n}_2$.

We start by imposing weakly the continuity condition (18a) *à la* Nitsche over the subdomain interface Γ; this way the problem in Eq. (3) can be written as

$$\sum_{i=1}^{2}\big[(\boldsymbol{\mathcal{M}}_i(\boldsymbol{y}_i)\partial_t\boldsymbol{y}_i,\boldsymbol{v}_i) + B_i\left(\boldsymbol{y}_i;\boldsymbol{y}_i,\boldsymbol{v}_i\right)\big] + \alpha\langle[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})]\!],[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{v})]\!]\rangle_\Gamma$$
$$- \langle[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})]\!],\{\boldsymbol{\mathcal{H}}^*(\boldsymbol{y};\boldsymbol{v})\}\rangle_\Gamma - \langle\{\boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{y})\},[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{v})]\!]\rangle_\Gamma = \sum_{i=1}^{2} L_i(\boldsymbol{v}_i), \tag{19}$$

which must hold for all test functions $\boldsymbol{v}_i$, $i = 1,2$, in the appropriate spaces. In this expression, α is a penalty-like parameter (that scales as $1/h$ in the discrete problem). The term $\langle\{\boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{y})\},[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{v})]\!]\rangle_\Gamma$ is obtained from integration by parts in each subdomain and the (consistent) term $\langle[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})]\!],\{\boldsymbol{\mathcal{H}}^*(\boldsymbol{y};\boldsymbol{v})\}\rangle_\Gamma$ is introduced to ensure adjoint consistency.

4.3. VMS-ROM approximation

We are now in a position to combine the treatment of the discontinuous interface with the stabilized formulation given by Eq. (15). Since the case of a standard Neumann–Dirichlet coupling described above does not offer any particular difficulty, here we present only the formulation when following the DG approach (Eq. (19)). To simplify the writing, we do not use the expanded form of the subscales in the element interiors and on the element boundaries, except for those on Γ. The problem to be solved consists in finding sequences $\{\boldsymbol{y}_{i,r}^j\}$, $i = 1,2$, in the appropriate ROM spaces, such that

$$\sum_{i=1}^{2}\Big[(\boldsymbol{\mathcal{M}}_i(\boldsymbol{y}_i)\delta_t\boldsymbol{y}_{i,r},\boldsymbol{v}_{i,r}) + B_i\left(\boldsymbol{y}_i;\boldsymbol{y}_{i,r},\boldsymbol{v}_{i,r}\right) + \sum_{K_i}\langle\check{\boldsymbol{y}}_i,\boldsymbol{\mathcal{L}}^*(\boldsymbol{y}_i;\boldsymbol{v}_{i,r})\rangle_{K_i}$$
$$+ \sum_{E_i}\langle\check{\boldsymbol{y}}_{i,E},[\![\boldsymbol{\mathcal{F}}^*(\boldsymbol{y};\boldsymbol{v}_r)]\!]\rangle_{E_i}\Big] - \langle[\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)]\!],\boldsymbol{\tau}_E[\![\boldsymbol{\mathcal{F}}^*(\boldsymbol{y};\boldsymbol{v}_r)]\!]\rangle_\Gamma$$
$$+ \alpha\langle[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})]\!],[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{v})]\!]\rangle_\Gamma - \langle[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{y})]\!],\{\boldsymbol{\mathcal{H}}^*(\boldsymbol{y};\boldsymbol{v})\}\rangle_\Gamma - \langle\{\boldsymbol{\mathcal{H}}(\boldsymbol{y};\boldsymbol{y})\},[\![\boldsymbol{\mathcal{D}}_n(\boldsymbol{v})]\!]\rangle_\Gamma$$
$$= \sum_{i=1}^{2} L_i(\boldsymbol{v}_{i,r}), \quad \text{for all test functions } \boldsymbol{v}_{i,r}, \text{ at } t^j,\ j = 1,2,\ldots, \tag{20}$$

where $\{K_i\}$ denotes the elements of the underlying FE partition in Ω_i and $\{E_i\}$ the *interior* edges.

A word of explanation is needed concerning the calculation of integrals over Γ when the FE meshes in Ω_1 and in Ω_2 do not coincide at this interface. Let us consider for example the term

$$\langle[\![\boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r)]\!],\boldsymbol{\tau}_E[\![\boldsymbol{\mathcal{F}}^*(\boldsymbol{y};\boldsymbol{v}_r)]\!]\rangle_\Gamma$$
$$= \langle\boldsymbol{\mathcal{F}}_1(\boldsymbol{y}_1;\boldsymbol{y}_{1,r}),\boldsymbol{\tau}_E\boldsymbol{\mathcal{F}}_1^*(\boldsymbol{y}_1;\boldsymbol{v}_{1,r})\rangle_\Gamma + \langle\boldsymbol{\mathcal{F}}_2(\boldsymbol{y}_2;\boldsymbol{y}_{2,r}),\boldsymbol{\tau}_E\boldsymbol{\mathcal{F}}_2^*(\boldsymbol{y}_2;\boldsymbol{v}_{2,r})\rangle_\Gamma$$
$$+ \langle\boldsymbol{\mathcal{F}}_1(\boldsymbol{y}_1;\boldsymbol{y}_{1,r}),\boldsymbol{\tau}_E\boldsymbol{\mathcal{F}}_2^*(\boldsymbol{y}_2;\boldsymbol{v}_{2,r})\rangle_\Gamma + \langle\boldsymbol{\mathcal{F}}_2(\boldsymbol{y}_2;\boldsymbol{y}_{2,r}),\boldsymbol{\tau}_E\boldsymbol{\mathcal{F}}_1^*(\boldsymbol{y}_1;\boldsymbol{v}_{1,r})\rangle_\Gamma. \tag{21}$$

First of all, a characteristic element length h needs to be defined to compute $\boldsymbol{\tau}_E$; in the case of different element sizes from both sides of Γ, we choose the largest h. The first two terms in the last expression offer no difficulty,

since Γ can be discretized using either the mesh of Ω_1 or of Ω_2 and numerical integration can be performed in the usual manner. The difficulty arises in the 'cross' terms. Consider for example $\langle \boldsymbol{\mathcal{F}}_1(\boldsymbol{y}_1; \boldsymbol{y}_{1,r}), \boldsymbol{\tau}_E \boldsymbol{\mathcal{F}}_2^*(\boldsymbol{y}_2; \boldsymbol{v}_{2,r})\rangle_\Gamma$. This term appears when $\boldsymbol{v}_{2,r} \neq \boldsymbol{0}$, i.e., when obtaining the discrete equations for the unknowns in Ω_2 and therefore the mesh of this domain needs to be used. Since $\boldsymbol{\mathcal{F}}_1(\boldsymbol{y}_1; \boldsymbol{y}_{1,r})$ can be computed in the mesh of Ω_1, some sort of interpolation of these boundary values to the edges of the mesh of Ω_2 on Γ will be required to evaluate the term we are considering. Analogous comments apply to the last term in Eq. (21), $\langle \boldsymbol{\mathcal{F}}_2(\boldsymbol{y}_2; \boldsymbol{y}_{2,r}), \boldsymbol{\tau}_E \boldsymbol{\mathcal{F}}_1^*(\boldsymbol{y}_1; \boldsymbol{v}_{1,r})\rangle_\Gamma$.

4.4. Iteration-by-subdomain strategy

In this work we approach the domain decomposition problem using an iteration-by-subdomain strategy, where we compute the unknowns in each subdomain assuming the data from the other known, and proceeding iteratively until convergence. This general idea encompasses in particular the classical Neumann–Dirichlet coupling, in which can be applied straightforwardly. In contrast, using the DG approach we can explore other coupling possibilities. Likewise, the iteration due to the nonlinearity of the problem can be dealt with at the same time as the domain decomposition coupling or through a nested strategy.

The equation to be solved for Ω_1 is obtained by taking $\boldsymbol{v}_{1,r} \neq \boldsymbol{0}$, $\boldsymbol{v}_{2,r} = \boldsymbol{0}$ in Eq. (20). Assuming the nonlinearity coupled with the iteration-by-subdomain and using a superscript in parenthesis to denote the iteration counter, k being the current iteration, this equation can be written as

$$
\begin{aligned}
&(\boldsymbol{\mathcal{M}}_1(\boldsymbol{y}_1^{(k-1)})\delta_t \boldsymbol{y}_{1,r}^{(k)}, \boldsymbol{v}_{1,r}) + B_1\left(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{y}_{1,r}^{(k)}, \boldsymbol{v}_{1,r}\right) \\
&\quad + \sum_{K_1} \langle \check{\boldsymbol{y}}_1^{(k)}, \boldsymbol{\mathcal{L}}^*(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{v}_{1,r})\rangle_{K_1} + \sum_{E_1} \langle \check{\boldsymbol{y}}_{1,E}^{(k)}, \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{v}_{1,r})\rangle_{E_1} \\
&\quad - \langle \boldsymbol{\mathcal{F}}(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{y}_{1,r}^{(l)}) + \boldsymbol{\mathcal{F}}(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{y}_{2,r}^{(k-1)}), \boldsymbol{\tau}_E \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{v}_{1,r})\rangle_\Gamma \\
&\quad + \alpha \langle \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{1,r}^{(m)}) + \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{2,r}^{(k-1)}), \boldsymbol{\mathcal{D}}_n(\boldsymbol{v}_{1,r})\rangle_\Gamma \\
&\quad - \frac{1}{2}\langle \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{1,r}^{(m)}) + \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{2,r}^{(k-1)}), \boldsymbol{\mathcal{H}}^*(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{v}_{1,r})\rangle_\Gamma \\
&\quad - \frac{1}{2}\langle \boldsymbol{\mathcal{H}}(\boldsymbol{y}_1^{(k-1)}; \boldsymbol{y}_{1,r}^{(l)}) + \boldsymbol{\mathcal{H}}(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{y}_{2,r}^{(k-1)}), \boldsymbol{\mathcal{D}}_n(\boldsymbol{v}_{1,r})\rangle_\Gamma = L_1(\boldsymbol{v}_{1,r}),
\end{aligned}
\tag{22}
$$

where the indexes l and m determine the type of coupling to be used. For example, for $l = k$, $m = k-1$ subdomain 1 would use the Neumann datum from subdomain 2, whereas for $l = k-1$, $m = k$ it would use the Dirichlet condition from subdomain 2. Nevertheless, a combined condition, with $l = m = k$, could also be used.

The equation for Ω_2, obtained by taking $\boldsymbol{v}_{2,r} \neq \boldsymbol{0}$, $\boldsymbol{v}_{1,r} = \boldsymbol{0}$ in Eq. (20), is

$$
\begin{aligned}
&(\boldsymbol{\mathcal{M}}_2(\boldsymbol{y}_2^{(k-1)})\delta_t \boldsymbol{y}_{2,r}^{(k)}, \boldsymbol{v}_{2,r}) + B_2\left(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{y}_{2,r}^{(k)}, \boldsymbol{v}_{2,r}\right) \\
&\quad + \sum_{K_2} \langle \check{\boldsymbol{y}}_2^{(k)}, \boldsymbol{\mathcal{L}}^*(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{v}_{2,r})\rangle_{K_2} + \sum_{E_2} \langle \check{\boldsymbol{y}}_{2,E}^{(k)}, \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{v}_{2,r})\rangle_{E_2} \\
&\quad - \langle \boldsymbol{\mathcal{F}}(\boldsymbol{y}_1^{(n)}; \boldsymbol{y}_{1,r}^{(n)}) + \boldsymbol{\mathcal{F}}(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{y}_{2,r}^{(m)}), \boldsymbol{\tau}_E \boldsymbol{\mathcal{F}}^*(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{v}_{2,r})\rangle_\Gamma \\
&\quad + \alpha \langle \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{1,r}^{(n)}) + \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{2,r}^{(l)}), \boldsymbol{\mathcal{D}}_n(\boldsymbol{v}_{2,r})\rangle_\Gamma \\
&\quad - \frac{1}{2}\langle \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{1,r}^{(n)}) + \boldsymbol{\mathcal{D}}_n(\boldsymbol{y}_{2,r}^{(l)}), \boldsymbol{\mathcal{H}}^*(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{v}_{2,r})\rangle_\Gamma \\
&\quad - \frac{1}{2}\langle \boldsymbol{\mathcal{H}}(\boldsymbol{y}_1^{(n)}; \boldsymbol{y}_{1,r}^{(n)}) + \boldsymbol{\mathcal{H}}(\boldsymbol{y}_2^{(k-1)}; \boldsymbol{y}_{2,r}^{(m)}), \boldsymbol{\mathcal{D}}_n(\boldsymbol{v}_{2,r})\rangle_\Gamma = L_2(\boldsymbol{v}_{2,r}).
\end{aligned}
\tag{23}
$$

Note that the appearance of indexes l and m in Eq. (23) has been swapped with respect to that of Eq. (22). Now, on top of the options $l, m = k-1, k$, there is also the possibility of using $n = k$ or $n = k-1$, i.e., to use the value just computed from Eq. (22) of the unknown in subdomain 1 or to use the value of the previous iteration. The first choice would be a Gauss–Seidel-type of coupling, whereas the second would be a Jacobi-type iterative scheme.

Along with the iterative strategy, to be able to converge to any solution regardless of using a Neumann–Dirichlet or a DG approach, we require an additional relaxation scheme for the transmitted quantities. Here, we have chosen

an Δ^2 Aitken relaxation scheme as the one used in [11], which reads

$$\boldsymbol{y}_i^{(k)} = \boldsymbol{y}_i^{(k)} + \omega_i^{(k)}(\boldsymbol{y}_i^{(k)} - \boldsymbol{y}_i^{(k-1)}), \quad i = 1, 2, \tag{24}$$

where $\omega_i^{(k)}$ is the relaxation parameter.

5. Adaptive coarse-mesh hyper-reduction

5.1. Mesh-based hyper-ROM

As it is well known, if the nonlinear terms are evaluated using the FE mesh of the FOM, the cost of assembling the matrices of the resulting algebraic system for the ROM is of the same order as that of the FOM. Some sort of approximation is thus required to evaluate these nonlinear contributions, which prevent from computing the system matrices in the offline stage. The method resulting from these approximations is called hyper-ROM.

Most hyper-ROM strategies are based on some sort of sampling to evaluate nonlinear terms. Here we will follow a completely different approach described in [1,34], where the mesh-based hyper-ROM proposed consists of the solution of the described ROM problem using a coarser mesh than the FOM one. In general, we want to use this coarser mesh without increasing the order of the error introduced by the ROM. To achieve this, the error measure provided by the subscales in Eq. (28) is crucial, as well as the AMR described later; the error measure we propose is given by Eq. (28) below.

Let $\mathscr{T}_H$ be a finite element partition of size $H > h$. Let us assume for both $\mathscr{T}_h$ and $\mathscr{T}_H$ a standard Lagrangian interpolation, with N_h nodes in $\mathscr{T}_h$ and N_H in $\mathscr{T}_H$ ($N_H < N_h$), $\{\varphi_h^a(\boldsymbol{x})\}$ being the basis for the FOM space ($a = 1, \dots, N_h$) and $\{\varphi_H^A(\boldsymbol{x})\}$ the basis for the FE space constructed from $\mathscr{T}_H$ ($A = 1, \dots, N_H$). Finally, let $\{\boldsymbol{\phi}_h^k(\boldsymbol{x})\}_{k=1}^r$ be the ROM basis obtained from the FOM solution. If $\boldsymbol{\phi}_h^{k,a}$, $a = 1, \dots, N_h$, are the nodal values of $\boldsymbol{\phi}_h^k(\boldsymbol{x})$ (those obtained from the POD), we may write the ROM solution at each time $t \in]0, t_f[$ as

$$\boldsymbol{y}_{r,h}(\boldsymbol{x}, t) = \sum_{k=1}^{r} \boldsymbol{\phi}_h^k(\boldsymbol{x}) Y^k(t) = \sum_{k=1}^{r} \left[\sum_{a=1}^{N_h} \varphi_h^a(\boldsymbol{x}) \boldsymbol{\phi}^{k,a} \right] Y^k(t), \tag{25}$$

where subscript h in $\boldsymbol{y}_{r,h}(\boldsymbol{x}, t)$ has been introduced to emphasize that it is computed from the FOM mesh and $\{Y^k(t)\}_{k=1}^r$ are the ROM degrees of freedom.

If Eq. (25) is used to evaluate the nonlinear terms, the cost will be $\mathscr{O}(N_h)$. The idea of the mesh-based hyper-ROM we propose is to project the ROM basis obtained from the FOM mesh to the new coarser mesh. Although other possibilities could be used, in this work we use a simple interpolation as projection. Thus, the nodal values of the ROM basis at the nodes of $\mathscr{T}_H$ can be obtained as

$$\boldsymbol{\phi}_h^k(\boldsymbol{x}_A) = \sum_{a=1}^{N_h} \varphi_h^a(\boldsymbol{x}_A) \boldsymbol{\phi}^{k,a}, \quad A = 1, \dots, N_H, \tag{26}$$

prior to any matrix evaluation, and then used to express the ROM solution as

$$\boldsymbol{y}_{r,H}(\boldsymbol{x}, t) = \sum_{k=1}^{r} \boldsymbol{\phi}_H^k(\boldsymbol{x}) Y^k(t) = \sum_{k=1}^{r} \left[\sum_{A=1}^{N_H} \varphi_H^A(\boldsymbol{x}) \boldsymbol{\phi}_h^k(\boldsymbol{x}_A) \right] Y^k(t). \tag{27}$$

The cost of using Eq. (27) to evaluate the nonlinear terms will be $\mathscr{O}(N_H)$, rather than the cost $\mathscr{O}(N_h)$ of using Eq. (25).

5.2. AMR approach to construct the hyper-ROM FE mesh

The success of the mesh-based hyper-ROM proposed relies in the refinement or coarsening of areas that are deemed of more or less importance, respectively. In this regard we could consider two options: a precomputed mesh based on some definition of the basis approximation error as in [25], or an AMR based on the real-time ROM error. In this work we propose a standard h-AMR approach using the algorithm described in [31]. Although the refinement and rebalance algorithms are easily included in the ROM, special attention should be given to the treatment of hanging nodes: since in this algorithm they are just an interpolation of the adjacent nodes and therefore do not provide any new information to the solution, we do not assemble them in the matrices or vectors.

In the AMR we propose, we know the FOM mesh $\mathscr{T}_h$ beforehand and we determine adaptively the coarse mesh $\mathscr{T}_H$ to be used in the hyper-ROM. These two meshes are in principle unrelated, but in the first iteration of the AMR procedure the latter can also be taken as a submesh the former, so that $\mathscr{T}_h$ is nested in $\mathscr{T}_H$.

The target mesh $\mathscr{T}_H$ of the AMR procedure should be such that the error of $\boldsymbol{y}_{r,H}$ given by Eq. (27) should be of the same order as the error of $\boldsymbol{y}_{r,h}$ given by Eq. (25) (after time discretization), that is to say,

$$\|\boldsymbol{y} - \boldsymbol{y}_{r,H}\| \approx \|\boldsymbol{y} - \boldsymbol{y}_{r,h}\|,$$

in a certain norm $\|\cdot\|$. However, as far as we are aware, *a priori* error estimates for the ROM are only available in terms of the eigenvalues discarded in the singular value decomposition associated to the POD, and not in terms of the mesh size h, even in the particular case of FE-based ROMs. Nevertheless, any *a posteriori* error estimator ξ, valid for a FE mesh, is in fact the only thing we need to drive the AMR. Indeed, if ξ_h is an error indicator for $\mathscr{T}_h$ and ξ_H the corresponding one for $\mathscr{T}_H$ (for example, the maximum of the element-wise error values), the condition to be fulfilled is

$$\xi_H \leq C\xi_h,$$

C being a constant close to 1. In practice, ξ_h is never computed, since $\boldsymbol{y}_{r,h}$ is too expensive to obtain. The AMR is thus stopped when ξ_H is below a given target error, which could be linked for example to the error of the FOM solution from which the ROM basis has been constructed.

5.3. Subscales as an error measure

In this work, we use the equations for subscales in the element interiors and in the boundaries to define an explicit error estimator that can be computed during the ROM solution. For that purpose we follow [29,30], where a scaled L^2-norm of the subscales is defined as $|\check{\boldsymbol{y}}|^2_{\boldsymbol{\tau}} := \check{\boldsymbol{y}}^\top \boldsymbol{\tau}^{-1} \check{\boldsymbol{y}}$, using the matrix of stabilization parameters $\boldsymbol{\tau}$ as scaling matrix. This way, we compute the error estimator at each element as

$$\xi_K^2 = \int_K \check{\boldsymbol{y}}^\top \boldsymbol{\tau}_K^{-1} \check{\boldsymbol{y}} + \sum_{E \subset \partial K} \int_E \check{\boldsymbol{y}}_E^\top \boldsymbol{\tau}_E^{-1} \check{\boldsymbol{y}}_E, \tag{28}$$

with $\check{\boldsymbol{y}}$ obtained by solving Eq. (7) at each time level t^j and $\check{\boldsymbol{y}}_E$ given by Eq. (12) (with the adequate modification for Neumann-type edges).

Since the subscales at the element interiors are part of the stabilization of Eq. (15), they are always computed —using Eq. (7)— at the integration points. However, when using these subscales as error estimation, they can be computed at any location in the interior of the element (e.g. at the center of the element or at the nodes).

Rather, the subscales on the element boundaries are defined by the inter-element jump operator, and therefore, the contribution of neighboring elements must be included. To allow computing the second term in Eq. (28) locally for each element, we follow the approach in [29,30], where the approximation described in [36,37] and the error estimator in [38] are used to express the term mentioned in terms of norms evaluated only in the element interior.

Note that the influence of the interior or boundary subscales in the error estimator ξ_K^2 depends on the definitions of the operators $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{F}}$, and the algorithmic constants used in the stabilization parameters $\boldsymbol{\tau}_K$ and $\boldsymbol{\tau}_E$. This way, we can argue that for problems that are more convection dominated, the interior subscales impact more in the error estimator, whereas for diffusion dominated problems the boundary subscales impact more.

6. Numerical examples

In this section we present numerical examples to test the behavior of the proposed ROM techniques. The first set of examples is used to illustrate the domain-decomposition stabilized formulation and the second set to test the AMR strategy. In all the examples we solve the FOM using a stabilized FE method with orthogonal subscales as the one described in [32] and we obtain the basis using an Singular Value Decomposition (SVD) from temporal snapshots as described in [1]. Furthermore, the domain discretizations for the ROM and the hyper-ROM are done using FEs, the time integration is carried out by an implicit Backward Differentiation Formula (BDF) of second order for the resolved scales and first order for the subscales, and the non-linearity is resolved using Picard's linearization scheme. For both sets of examples we use the incompressible Navier–Stokes equations and the Boussinesq approximation for thermally coupled incompressible flows.

For both physical models we can define the solution spaces $\mathscr{Y}$ for the trial solutions and $\mathscr{Y}_0$ for the test functions, and construct the FOM and ROM approximation spaces $\mathscr{Y}_h$ and $\mathscr{Y}_r$ as usual. For each ROM we set a particular amount of basis vectors r, which has an associated and retained energy η (quotient of the sum of the eigenvalues in the SVD decomposition of the retained modes over that of all the modes).

Incompressible Navier–Stokes equations

The incompressible Navier–Stokes problem consists of finding a velocity $\boldsymbol{u} : \Omega\times]0, t_\mathrm{f}[\to \mathbb{R}^d$ and a pressure $p : \Omega\times]0, t_\mathrm{f}[\to \mathbb{R}$ satisfying appropriate initial and boundary conditions and such that

$$\begin{aligned} \partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad & \text{in } \Omega, \quad t \in]0, t_\mathrm{f}[, \\ \nabla\cdot\boldsymbol{u} = 0 \qquad & \text{in } \Omega, \quad t \in]0, t_\mathrm{f}[, \end{aligned}$$

where ν denotes the kinematic viscosity and $\boldsymbol{f}$ a vector of body forces.

To write the stabilized problem, we define the terms that correspond to the abstract ones in Eq. (15) as

$$\begin{aligned} \boldsymbol{\mathcal{M}}(\boldsymbol{y}) &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0}^\top \\ \boldsymbol{0} & 0 \end{bmatrix}, \\ \boldsymbol{\mathcal{L}}(\boldsymbol{y};\boldsymbol{y}_r) &= \begin{bmatrix} \boldsymbol{u}\cdot\nabla\boldsymbol{u}_r - \nu\Delta\boldsymbol{u}_r + \nabla p_r \\ \nabla\cdot\boldsymbol{u}_r \end{bmatrix}, \quad \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r) = \begin{bmatrix} -\boldsymbol{n}\cdot p_r\boldsymbol{I} + \boldsymbol{n}\cdot\nu\nabla\boldsymbol{u}_r \\ 0 \end{bmatrix}, \\ \boldsymbol{\mathcal{L}}^*(\boldsymbol{y};\boldsymbol{v}_r) &= \begin{bmatrix} -\boldsymbol{u}\cdot\nabla\boldsymbol{v}_r - \nu\Delta\boldsymbol{v}_r - \nabla q_r \\ -\nabla\cdot\boldsymbol{v}_r \end{bmatrix}, \quad \boldsymbol{\mathcal{F}}^*(\boldsymbol{y};\boldsymbol{y}_r) = \begin{bmatrix} \boldsymbol{n}\cdot q_r\boldsymbol{I} + \boldsymbol{n}\cdot\nu\nabla\boldsymbol{v}_r \\ 0 \end{bmatrix}. \end{aligned}$$

The Galerkin form is

$$B(\boldsymbol{y};\boldsymbol{y}_r,\boldsymbol{v}_r) = \langle \boldsymbol{u}\cdot\nabla\boldsymbol{u}_r, \boldsymbol{v}_r\rangle + (\nu\nabla\boldsymbol{u}_r - p_r\boldsymbol{I}, \nabla\boldsymbol{v}_r) + (\nabla\cdot\boldsymbol{u}_r, q_r).$$

The stabilization parameter matrix is defined as

$$\boldsymbol{\tau}_K(\boldsymbol{y}) = \mathrm{diag}(\tau_1\boldsymbol{I}, \tau_2) = \begin{bmatrix} (c_1\frac{\nu}{h^2} + c_2\frac{|\boldsymbol{u}|_K}{h})^{-1}\boldsymbol{I} & \boldsymbol{0}^\top \\ \boldsymbol{0} & \nu + \frac{c_2}{c_1}|\boldsymbol{u}|h \end{bmatrix}, \quad \text{in } K \in \mathscr{T}_h,$$

with $c_1 = 4$ and $c_2 = 2$ for linear elements, and $\boldsymbol{\tau}_E = \mathrm{diag}(\tau_E\boldsymbol{I}, 0)$, $\tau_E^{-1} = c_E\frac{\nu}{h}$, $\boldsymbol{I}$ being the $d\times d$ identity.

Boussinesq equations

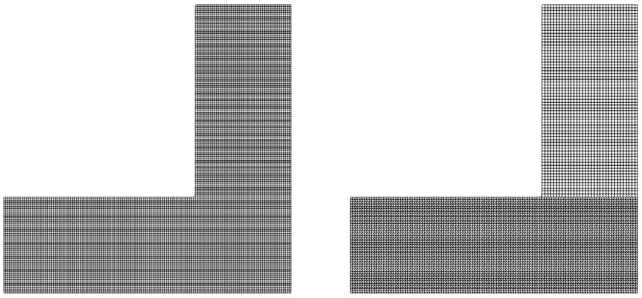
The Boussinesq approximation consists of finding a velocity $\boldsymbol{u} : \Omega\times]0, t_\mathrm{f}[\to \mathbb{R}^d$, a pressure $p : \Omega\times]0, t_\mathrm{f}[\to \mathbb{R}$ and a temperature $T : \Omega\times]0, t_\mathrm{f}[\to \mathbb{R}$ satisfying appropriate initial and boundary conditions and such that

$$\begin{aligned} \rho\partial_t\boldsymbol{u} + \rho\boldsymbol{u}\cdot\nabla\boldsymbol{u} - \mu\Delta\boldsymbol{u} + \nabla p = \rho\boldsymbol{f} - \rho\boldsymbol{g}\beta(T - T_0) \qquad & \text{in } \Omega, \quad t \in]0, t_\mathrm{f}[, \\ \nabla\cdot\boldsymbol{u} = 0 \qquad & \text{in } \Omega, \quad t \in]0, t_\mathrm{f}[, \\ \rho c_p\partial_t T + \rho c_p\boldsymbol{u}\cdot\nabla T - \lambda\Delta T = Q \qquad & \text{in } \Omega, \quad t \in]0, t_\mathrm{f}[, \end{aligned}$$

where, ρ denotes the density, μ the dynamic viscosity, $\boldsymbol{f}$ a vector of body forces, $\boldsymbol{g}$ the gravity acceleration vector, c_p the specific heat coefficient at constant pressure, λ the thermal conductivity, β the coefficient of thermal expansion, Q the heat source and T_0 a reference temperature.

In the same way as for the incompressible Navier–Stokes equations, we define the terms that correspond to the abstract ones in Eq. (15) as

$$\begin{aligned} \boldsymbol{\mathcal{M}}(\boldsymbol{y}) &= \begin{bmatrix} \rho\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 & 0 \\ \boldsymbol{0}^\top & 0 & \rho c_p \end{bmatrix}, \\ \boldsymbol{\mathcal{L}}(\boldsymbol{y};\boldsymbol{y}_r) &= \begin{bmatrix} \rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}_r - \mu\Delta\boldsymbol{u}_r + \nabla p_r \\ \nabla\cdot\boldsymbol{u}_r \\ \rho c_p\boldsymbol{u}\cdot\nabla T_r - \lambda\Delta T_r \end{bmatrix}, \quad \boldsymbol{\mathcal{F}}(\boldsymbol{y};\boldsymbol{y}_r) = \begin{bmatrix} -\boldsymbol{n}\cdot p_r\boldsymbol{I} + \boldsymbol{n}\cdot\mu\nabla\boldsymbol{u}_r \\ 0 \\ -\lambda\boldsymbol{n}\cdot\nabla T_r \end{bmatrix}, \end{aligned}$$



(a) Unified mesh. (b) Decomposed mesh.

Fig. 2. Comparison of meshes for unified and decomposed domains.

$$\boldsymbol{\mathcal{L}}^*(\boldsymbol{y};\boldsymbol{v}_r)=\begin{bmatrix}-\rho\boldsymbol{u}\cdot\nabla\boldsymbol{v}_r-\mu\Delta\boldsymbol{v}_r-\nabla q_r\\ -\nabla\cdot\boldsymbol{v}_r\\ -\rho c_p\boldsymbol{u}\cdot\nabla w_r-\lambda\Delta w_r\end{bmatrix},\quad \boldsymbol{\mathcal{F}}^*(\boldsymbol{y};\boldsymbol{y}_r)=\begin{bmatrix}\boldsymbol{n}\cdot q_r\boldsymbol{I}+\boldsymbol{n}\cdot\nu\nabla\boldsymbol{v}_r\\ 0\\ \lambda\boldsymbol{n}\cdot\nabla w_r\end{bmatrix},$$

The Galerkin form is:

$$\begin{aligned}B(\boldsymbol{y};\boldsymbol{y}_r,\boldsymbol{v}_r)=&\langle\rho\boldsymbol{u}\cdot\nabla\boldsymbol{u}_r,\boldsymbol{v}_r\rangle+(\mu\nabla\boldsymbol{u}_r-p_r\boldsymbol{I},\nabla\boldsymbol{v}_r)+(\nabla\cdot\boldsymbol{u}_r,q_r)\\&+\langle\rho c_p\boldsymbol{u}\cdot\nabla T_r,w_r\rangle+(\lambda\nabla T_r,\nabla w_r),\end{aligned}$$

The stabilization parameter matrix is defined as

$$\begin{aligned}\boldsymbol{\tau}_K(\boldsymbol{y})&=\operatorname{diag}(\tau_1\boldsymbol{I},\tau_2,\tau_3)\\&=\begin{bmatrix}(c_1\frac{\mu}{h^2}+c_2\frac{|\boldsymbol{u}|_K}{h})^{-1}\boldsymbol{I} & \boldsymbol{0} & 0\\ \boldsymbol{0}^\top & \mu+\frac{c_2}{c_1}|\boldsymbol{u}|h & 0\\ \boldsymbol{0}^\top & 0 & (c_1\frac{\lambda}{h^2}+c_2\frac{|\boldsymbol{u}|_K}{h})^{-1}\end{bmatrix},\quad \text{in } K\in\mathscr{T}_h,\end{aligned}$$

and now $\boldsymbol{\tau}_E=\operatorname{diag}(\tau_{E,u}\boldsymbol{I},0,\tau_{E,T})$, $\tau_{E,u}^{-1}=c_E\frac{\mu}{h}$, $\tau_{E,T}^{-1}=c_E\frac{\lambda}{h}$.

In our implementation, we have solved the problem in a standard iterative way, solving first for the velocity and the pressure and then for the temperature, and iterating until convergence.

6.1. Domain decomposition

6.1.1. Flow in an L-shaped domain

In this example we solve the Navier–Stokes equations for the L-shaped domain shown in Section 4. The computational domain consists of the union of two rectangles $[0,3]\times[0,1]$ and $[2,3]\times[1,3]$. The inlet is taken at $x=0$, with a discontinuous inflow velocity $\boldsymbol{u}=(1,0)$ for $0\le y\le\frac{1}{2}$ and $\boldsymbol{u}=(0,0)$ for $\frac{1}{2}<y\le 1$; the outflow (where both the x and y velocity components are left free) is set at $y=3$, whereas the rest of the boundaries have a prescribed null velocity. The viscosity is prescribed as $\nu=0.005$.

To evaluate the performance of the domain decomposition algorithm in the ROM formulation, we solve the described problem using different approaches: a unified domain, a Neumann–Dirichlet approach and a DG approach — using the two rectangular subdomains described before for the last two cases. For the joined domain we solve two cases, one using a fine mesh of 50 000 quadrilateral linear elements with an element size $h=0.01$ as a reference solution and one using a mesh of 12 500 triangular linear elements with an element size $h=0.02$. For the decomposed domain we use a mesh of 9600 triangular linear elements with an element size $h=0.025$ for the lower rectangle and a mesh of 2211 quadrilateral linear elements with an element size $h=0.03$ for the upper one. Fig. 2 shows a comparison of the meshes for the unified and decomposed domains.

For all the cases we use a time step of $\delta t=0.0025$ and we construct the reduced basis from the 1200 initial time steps of simulation. Likewise, the ROM simulations and the comparison with the FOM are done over the initial transient period $t=[0,3]$. For the domain decomposition FOM we use a Neumann–Dirichlet approach.

Table 1
Labels for FOM and ROM cases.

FOM_{ref}	Fine mesh unified domain
FOM	Unified domain
FOM-DD	Decomposed domain
ROM	Unified domain
ROM-DD_{A}	Neumann–Dirichlet with subscales in the interface
ROM-DD_{B}	Neumann–Dirichlet without subscales in the interface
ROM-DD_{C}	DG with subscales in the interface
ROM-DD_{D}	DG without subscales in the interface

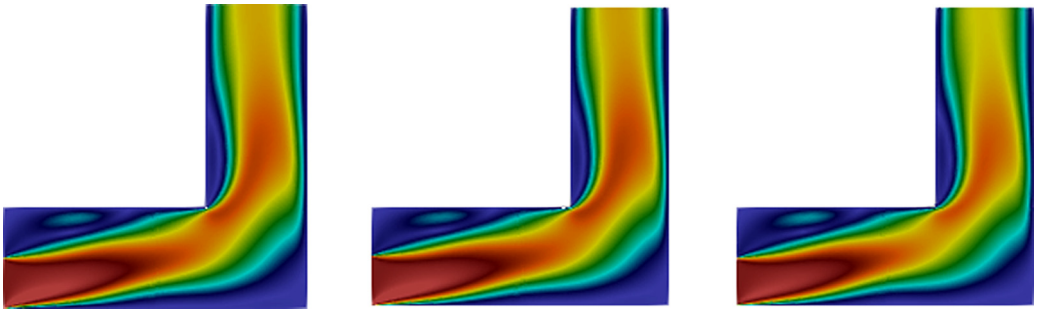


Fig. 3. Contour plots of velocity magnitude for the FOMs at $t = 3$. From left to right: FOM_{ref}, FOM and FOM-DD.

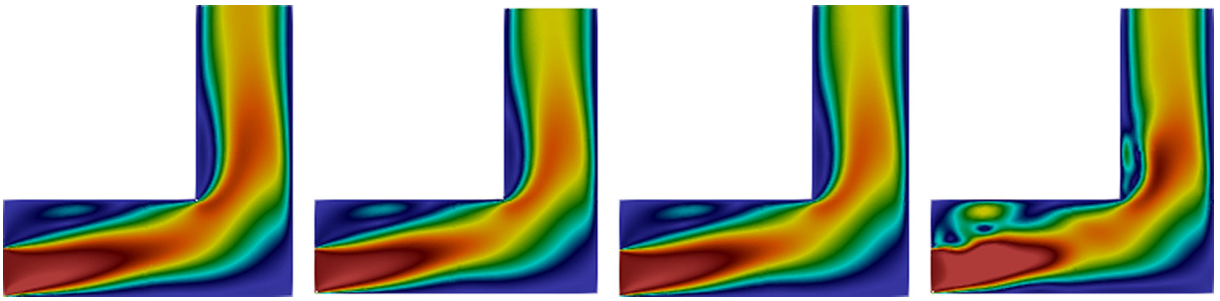


Fig. 4. Velocity contour plots for the ROMs at $t = 3$. From left to right: ROM, ROM-DD_{A}, ROM-DD_{B}, ROM-DD_{C}.

In the domain decomposition cases, we construct local bases for each subdomain, this means that the bases for both cases —unified or decomposed— have different relation between the retained energy η and the amount of basis vectors r. We choose to solve both cases using the same amount of retained energy $\eta = 0.99$ regardless of the amount of basis vectors, yielding $r = 8$ for the unified mesh, $r = 5$ for the lower subdomain and $r = 6$ for the upper subdomain.

Using both domain decomposition approaches —Neumann–Dirichlet and DG— we solve two cases for the ROM, with and without the subscales on the subdomain interface (Eq. (21)), where we set $c_E = 0.001$. The DG cases are solved using a Gauss–Seidel coupling, and setting $\alpha = 10.0/h$ (recall that it should scale as $1/h$ upon changes of mesh size).

In Table 1 we label all the FOM and ROM results that are presented in the following figures. Figs. 3 and 4 show contour plots of velocity at $t = 3$ for all the cases of FOM and ROM, respectively. We obtained from all the ROM cases meaningful solutions except for the DG approach without the subscales on the boundary interface —ROM-DD_{D}, the solution of which diverged. We remark that for the Neumann–Dirichlet approach the use or not of the subscales in the subdomain interface shows no visible difference. For the DG approach velocity results could have been improved changing α, but we have not tried to determine the optimal value of this parameter.

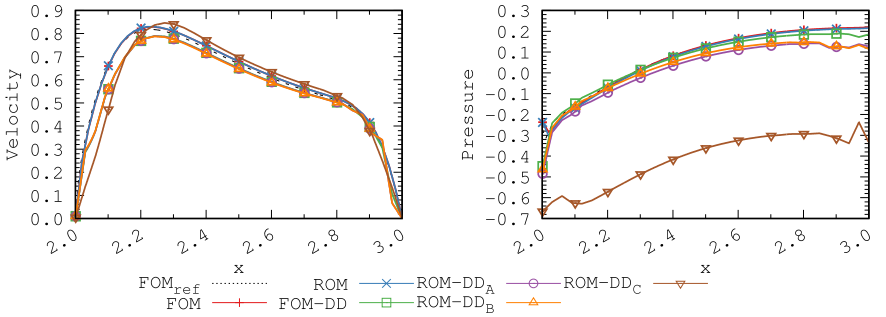


Fig. 5. Flow in an L-shaped domain: velocity and pressure at $y = 1$ and $t = 3$.

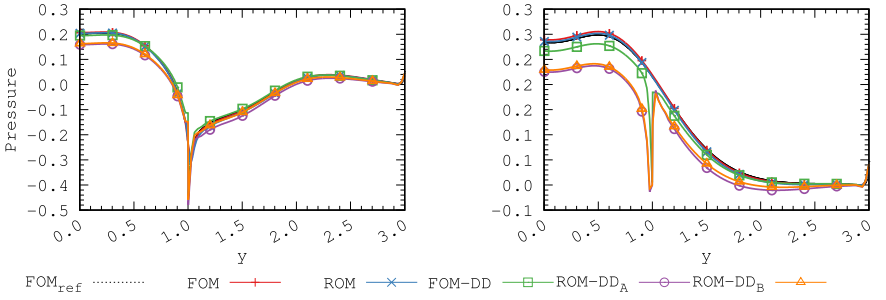


Fig. 6. Flow in an L-shaped domain: pressure at $x = 2$ (left) and $x = 3$ (right) for $t = 3$.

Fig. 5 compares the velocity and pressure over the cut between subdomains for all the cases at the last time step of the simulation, $t = 3$. Although the velocity solutions behave in a similar way for all the cases, we can notice that the DG approach has a worse approximation in the pressure field. This is observed from Fig. 6, where we show the pressure at $x = 2$ and $x = 3$ for all the cases at $t = 3$; we see how the pressure is singular in the non-convex corner of the L, and singular in the other corner of the interface for the domain decomposition cases.

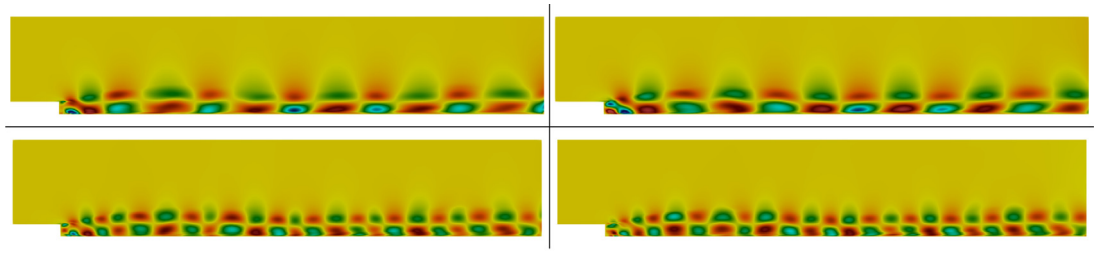
6.2. AMR hyper-ROM

6.2.1. Backward-facing step

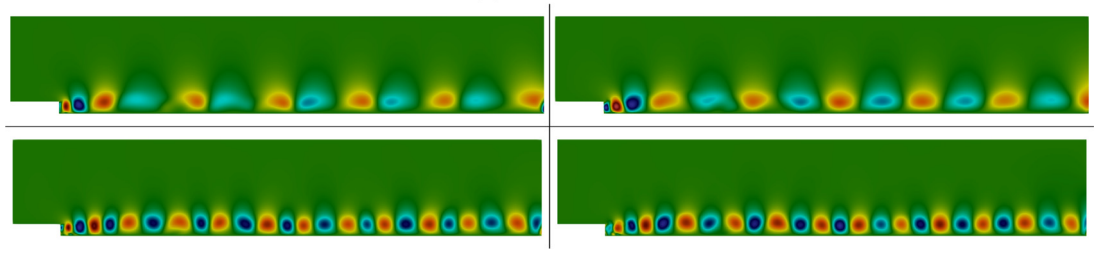
As the first numerical example to test the behavior of the AMR hyper-ROM formulation, we solve a two dimensional flow over a backward facing step in the same way as the one presented in [1,39]. The computational domain is set as $[0, 44]\times[0, 9]\setminus[0, 4]\times[0, 1]$, with inflow and outflow boundary conditions set as (1.0, 0.0) and free, respectively. At the lower and upper boundaries we set a von Kármán law of the wall with a boundary layer width $\delta = 0.001$. The viscosity is prescribed to $\nu = 0.00005$, resulting in a Reynolds number of 20 000. No turbulence model is used, letting the stabilization act as an Implicit Large Eddy Simulation model [33,40] if required.

We solve the FOM using a discretized domain of 61 520 quadrilateral bilinear elements and 62 214 nodes and a time step of $\delta t = 0.05$ for the time discretization; we construct a ROM basis over 1000 time steps after an initial 100 steps of simulation. Fig. 7 shows the four first modes of the reduced basis for velocity and pressure.

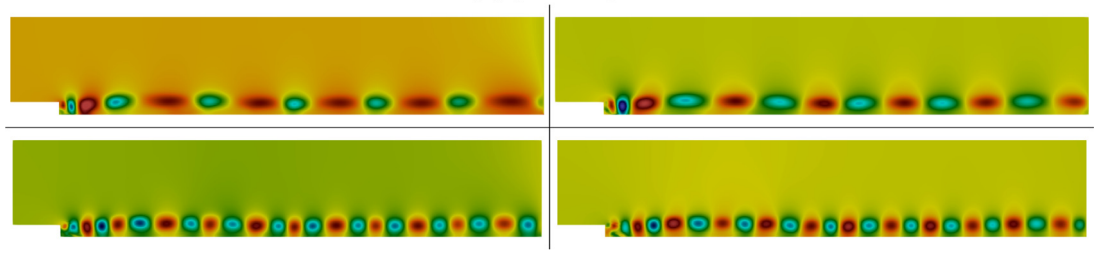
In this example we solve the ROM and hyper-ROMs using a basis of $r = 21$ modes with a retained energy of $\eta = 0.95$. To apply the AMR hyper-ROM we start from a mesh of 1497 quadrilateral bilinear elements and 1392 nodes, from which we follow a three level refinement with a refinement tolerance set as $10^{-5} < \xi_K^2 < 10^{-4}$, ξ_K being given by Eq. (28). Fig. 8 shows a comparison of the calculated ξ_K^2 at the time $t = 50$ for the four cases. Figs. 8c and 8d show the meshes for the uniform and the AMR hyper-ROM, with 5568 and 4977 elements, respectively. Comparing Figs. 7 and 8d, we can identify the area with the finer mesh with the part where the basis contains more information. In Fig. 9 we display contour plots of the velocity magnitude and the pressure for the AMR hyper-ROM for the last time step of the simulation, $t = 50$.



(a) x-velocity.

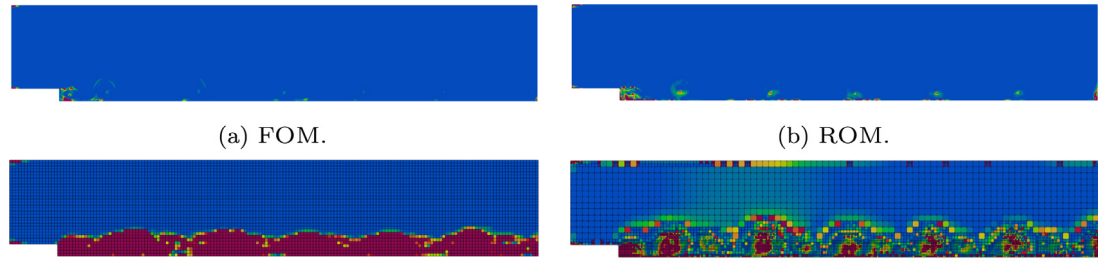


(b) y-velocity.



(c) Pressure.

Fig. 7. Backward facing step: first four modes of the reduced basis.



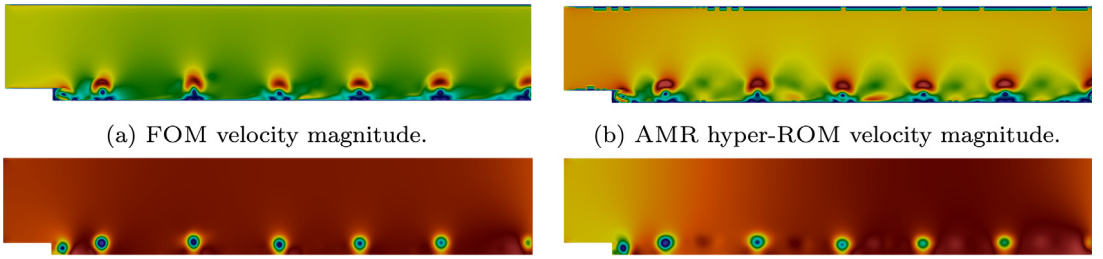
(a) FOM. (b) ROM.

(c) Hyper-ROM. (d) AMR hyper-ROM.

Fig. 8. Backward facing step: contour plot of $\xi_K^2 \in [10^{-5}, 10^{-4}]$ for FOM, ROM, hyper-ROM and AMR hyper-ROM cases at $t = 50$.

To compare in a more quantitative way the solutions, we present in Fig. 10 the velocity and the pressure and in Fig. 11 a comparison of their spectra at the control point (22, 1), for the FOM, ROM, hyper-ROM and AMR hyper-ROM (labeled HROM_a) with an average of 4723 elements.

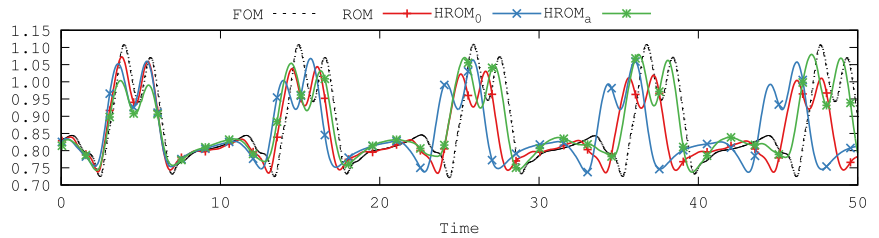
In both figures the use of the AMR hyper-ROM improves the behavior of the solution compared to the uniform mesh hyper-ROM while maintaining a similar computational cost. Moreover, this method indicates improvement over the ROM, correcting the amplitude and frequency for velocity and pressure even further. We can argue that this outperforming of the AMR hyper-ROM over the ROM can be seen as a result of comparing a single point instead of using a more representative value of the overall solutions. In Section 6.2.2 we do a comparison using the force exerted over a surface as a more accurate measure of the error.



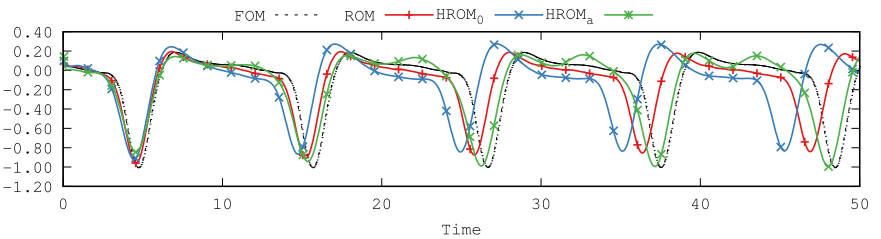
(a) FOM velocity magnitude. (b) AMR hyper-ROM velocity magnitude.

(c) FOM pressure. (d) AMR hyper-ROM pressure.

Fig. 9. Backward facing step: contour plots for FOM and AMR hyper-ROM at $t = 50$ with $\eta = 0.95$ and $r = 21$.



(a) Velocity magnitude.



(b) Pressure.

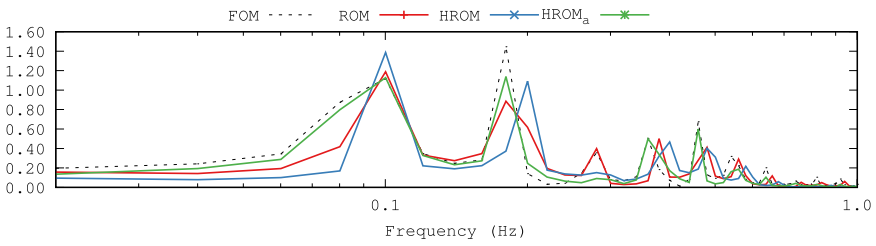
Fig. 10. Backward facing step: comparison at the control point (22, 1) for FOM, ROM, hyper-ROM and adaptive hyper-ROM with $\eta = 0.95$ and $r = 21$.

6.2.2. *Flow over a cylinder*

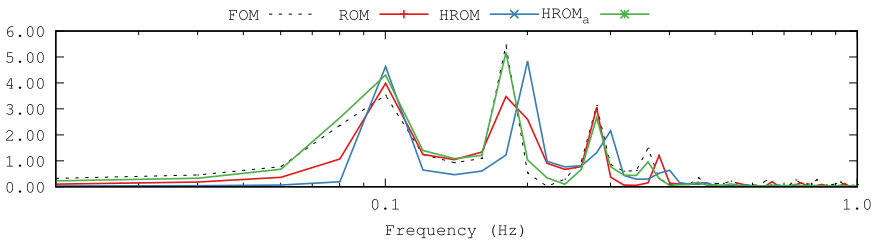
The second numerical example for the AMR hyper reduction consists in a two dimensional flow over a cylinder similar to the one presented in [1]. The computational domain is $\Omega = [0, 16] \times [0, 8]$, with the cylinder D of diameter 1 and centered at $(4, 4)$. The velocity at $x = 0$ is prescribed to $(1, 0)$, whereas at $y = 0$ and $y = 8$ the y-velocity component is prescribed to 0 and the x-velocity component is left free. The outflow (where both the x and y velocity components are left free) is $x = 16$. The viscosity is prescribed to $\nu = 0.001$, resulting in a Reynolds number of 1000. In the same way as for the previous example we test the formulation using a FOM and a ROM with the same mesh, and a hyper-ROM and an AMR hyper-ROM with a substantially lower amount of elements.

For the FOM the domain is discretized using a symmetric mesh of 92 320 bilinear elements and a time step $\delta t = 0.05$. We gather 1000 snapshots after an initial transient of 100 steps of simulation to compute the ROM basis. Fig. 12 shows the four first modes of the reduced basis for velocity and pressure.

In this example we solve the ROM and both hyper-ROMs using a basis of $r = 6$ modes with a retained energy of $\eta = 0.75$. The hyper-ROM domain is discretized using a similarly distributed mesh consisting of 18 360 bilinear elements, whilst for the AMR hyper-ROM we start with a discretized domain of 1840 bilinear elements reaching on average 19 392 elements using a three level refinement with the refinement tolerance set as $10^{-6} < \xi_K^2 < 10^{-5}$. Fig. 13 shows the fine, coarse and refined meshes.

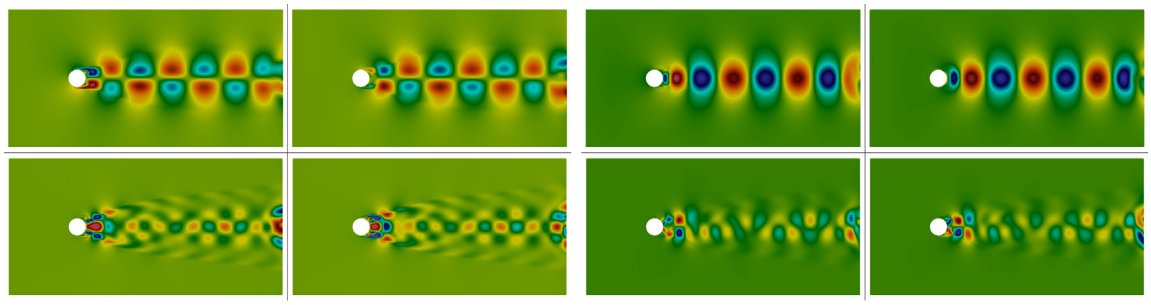


(a) Velocity magnitude.

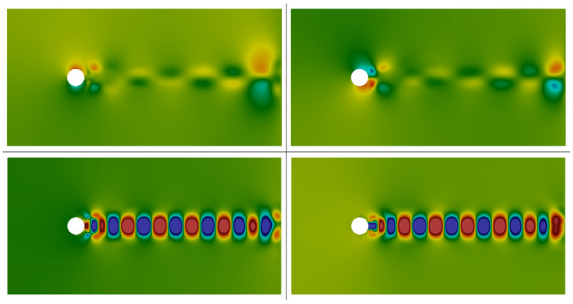


(b) Pressure.

Fig. 11. Backward facing step: Fourier transform at the control point $(22, 1)$ for FOM, ROM, hyper-ROM and adaptive hyper-ROM with $\eta = 0.95$ and $r = 21$.



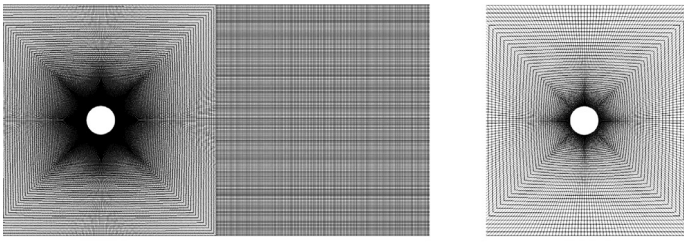
(a) x-velocity. (b) y-velocity.



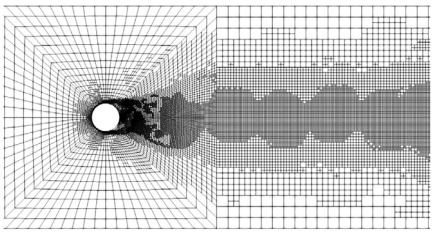
(c) Pressure.

Fig. 12. Flow over a cylinder: first four modes of the reduced basis.

Fig. 14 shows the calculated ξ_K^2 for the four cases at $t = 50$. In the same way as in the previous example, when comparing Figs. 12 and 14d we observe that the refinement occurs in the same areas where the basis contains more information.

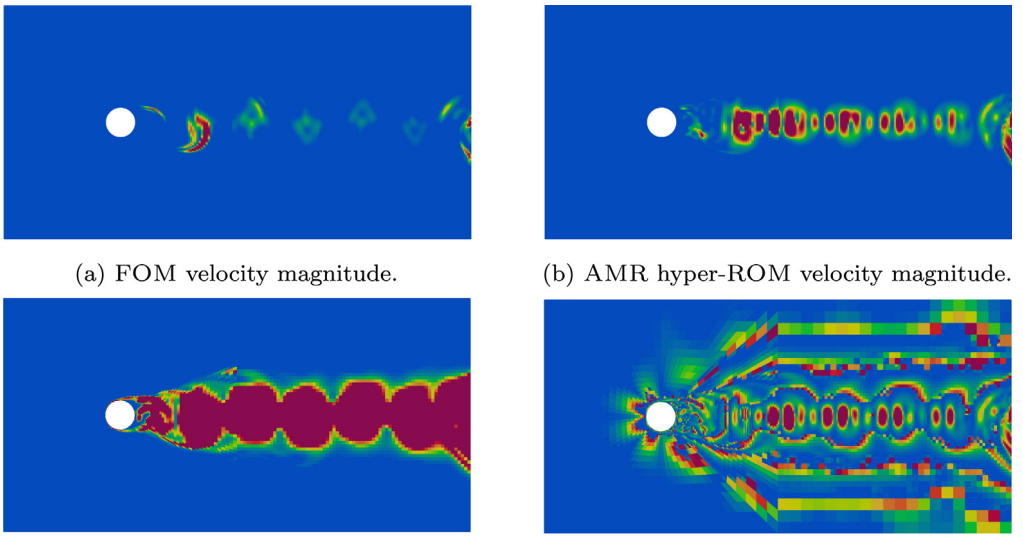


(a) FOM mesh. (b) Hyper-ROM mesh.



(c) AMR hyper-ROM mesh at $t = 50$.

Fig. 13. Flow over a cylinder: mesh for FOM, hyper-ROM and AMR hyper-ROM.



(a) FOM velocity magnitude. (b) AMR hyper-ROM velocity magnitude.

(c) FOM pressure. (d) AMR hyper-ROM pressure.

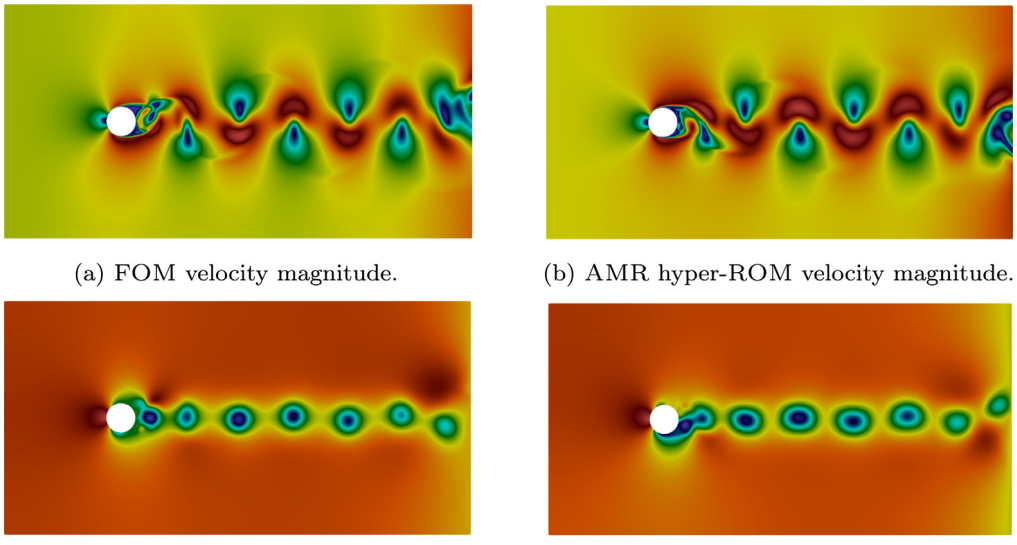
Fig. 14. Flow over a cylinder: contour plot of $\xi_K^2 \in [10^{-6}, 10^{-5}]$ for FOM, ROM, hyper-ROM and AMR hyper-ROM at $t = 50$.

Fig. 15 shows a comparison of contour plots of velocity magnitude and pressure for FOM and AMR hyper-ROM, where the quality of the approximation is evident, being the results obtained with both approaches almost indistinguishable.

To compare in a more quantitative way the results, we use the norm of the total force exerted over the cylinder boundary $\Gamma_\circ$, defined as

$$F^\circ(t) = \left| \int_{\Gamma_\circ} \boldsymbol{\mathcal{F}}(\boldsymbol{u}(\boldsymbol{x}, t)) \, \mathrm{d}\Gamma \right| .$$

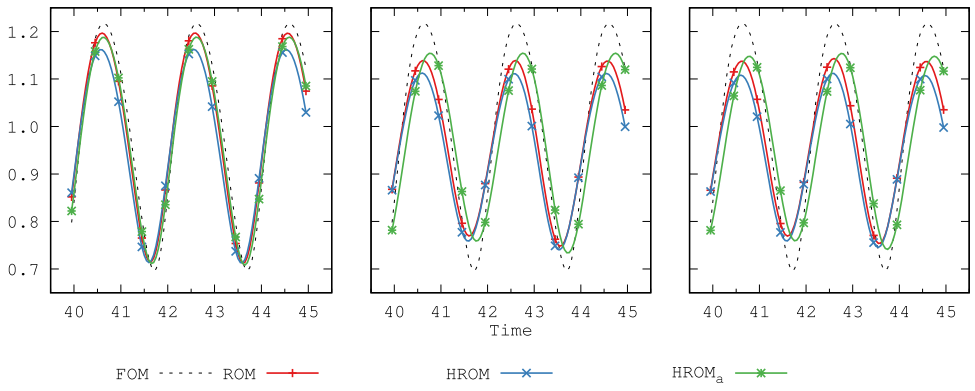
Fig. 16 shows the comparison of the total force in the time interval $[40, 45]$ for all the cases, using three different sizes of the basis. Fig. 17 shows a comparison of the spectra of the force for the same cases.



(a) FOM velocity magnitude. (b) AMR hyper-ROM velocity magnitude.

(c) FOM pressure. (d) AMR hyper-ROM pressure.

Fig. 15. Flow over a cylinder: contour plots for FOM and AMR hyper-ROM at $t = 50$ with $\eta = 0.75$, $r = 6$.



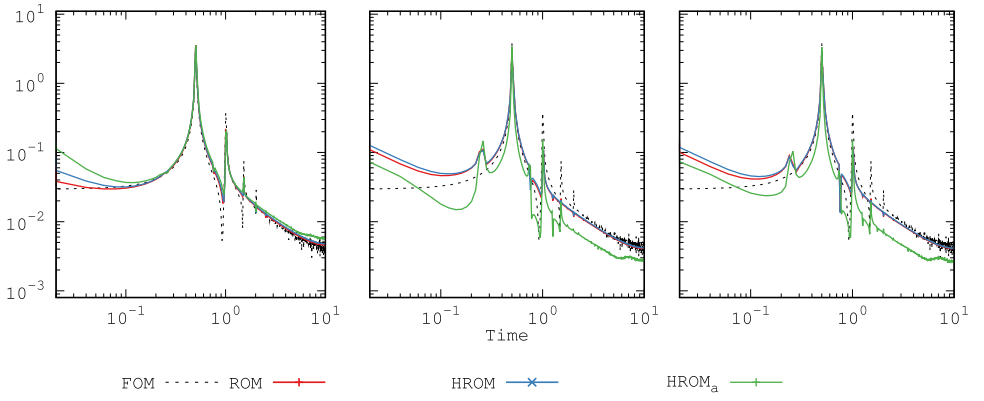
(a) $\eta = 0.75$, $r = 6$. (b) $\eta = 0.67$, $r = 5$. (c) $\eta = 0.6$, $r = 4$.

Fig. 16. Flow over a cylinder: comparison of F° between FOM, ROM, hyper-ROM and AMR hyper-ROM.

To evaluate the behavior of the formulation for dynamic physical phenomena we choose a Root-Mean-Square Deviation (RMSD) of the ROM solution with respect to the FOM. With $F^\circ_{j,\mathrm{FOM}}$ defined as the total force obtained with the FOM at time t^j, $j = 1, \ldots, S$, and $F^\circ_{j,\mathrm{ROM}}$ the one obtained with any of the ROMs, we set

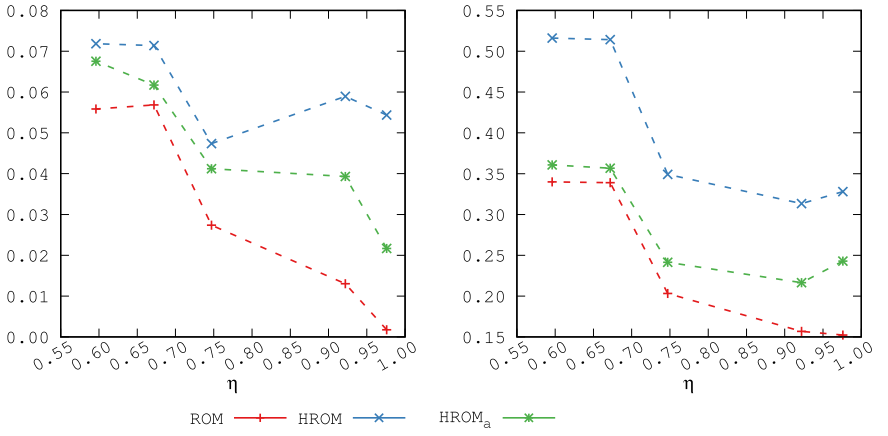
$$F^\circ_{\mathrm{RMSD}} = \sqrt{\frac{1}{S}\sum_{j=1}^{S}\left(F^\circ_{j,\mathrm{ROM}} - F^\circ_{j,\mathrm{FOM}}\right)^2}.$$

Fig. 18 shows the RMSD of the total force F°_{RMSD} and the time average of the SGS norm $\overline{\xi^2} = \frac{1}{S}\sum_{j=1}^{S}\sum_K \xi_K^2|_{t^j}$, for 5 cases with $r = [4, 5, 6, 20, 43]$ modes and a retained energy of $\eta = [0.6, 0.67, 0.75, 0.92, 0.98]$. Although the qualitative comparison of the total force F° in Fig. 16 and the spectrum in Fig. 17 displays a better performance of the AMR hyper-ROM over the ROM, the quantitative comparison using the RMSD reveals that, albeit being better than its uniform mesh counterpart, the hyper-ROM approaches the ROM solution as expected.



(a) $\eta = 0.75$, $r = 6$. (b) $\eta = 0.67$, $r = 5$. (c) $\eta = 0.6$, $r = 4$.

Fig. 17. Flow over a cylinder: Fourier transform of F° for FOM, ROM, hyper-ROM and AMR hyper-ROM.



(a) F°_{RMSD}. (b) Time average of ξ^2.

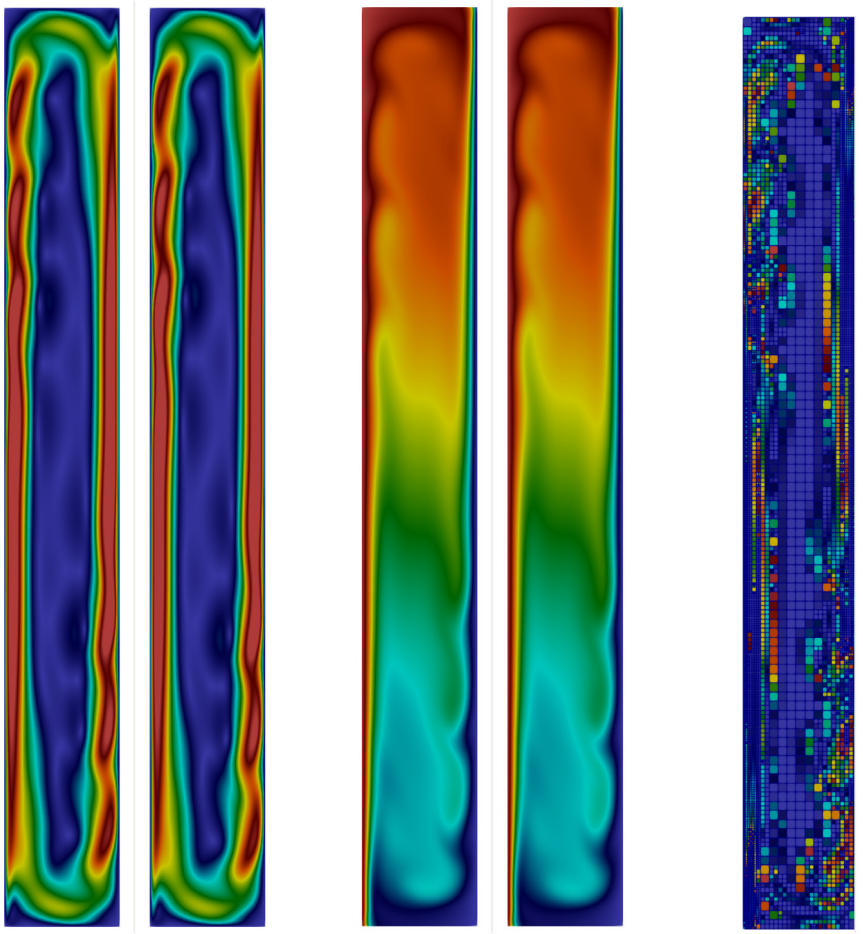
Fig. 18. Flow over a cylinder: F°_{RMSD} and ξ^2 for ROM, hyper-ROM and AMR hyper-ROM.

6.2.3. Flow in a differentially heated cavity with aspect ratio 8

In this example we model the flow in a differentially heated cavity with aspect ratio 8 as the one presented in [41,42] with a Rayleigh number (defined below) $\text{Ra} = 3.45 \cdot 10^5$, where it is known that a Hopf bifurcations occur and the flow is oscillatory.

The computational domain is defined as $\Omega = [0, W] \times [0, H]$, with $H/W = 8$. The temperatures on the walls perpendicular to the x-coordinate (horizontal) are fixed to $T_h = 600$ (left) and $T_c = 300$ (right), while adiabatic boundary conditions are prescribed on the remaining walls. No slip conditions are set for the velocity over all walls. A gravity acceleration $g = 6.599$ is imposed in the negative y-coordinate (vertical). The initial temperature and density are set to $T_0 = 450$ and $\rho = 1$, respectively, and the dimensionless Prandtl and Rayleigh numbers are set to $\text{Pr} = \frac{c_p \mu}{\lambda} = 0.71$ and $\text{Ra} = \frac{\beta |\boldsymbol{g}| \rho^2 c_p}{\lambda \mu}(T_h - T_c) = 5 \cdot 10^5$. The flow is considered an ideal gas with physical properties (in SI units) $R = 287.0$ (universal gas constant), $c_p = 1020$, $\mu = 0.001$, $\lambda = 3.59$ and $\beta = 0.00222$.

We solve the FOM and the ROM using a discretized domain of 80 000 quadrilateral bilinear elements and a time step of $\delta t = 0.01$; we construct a ROM basis over 1000 time steps. The AMR hyper-ROM starts from a mesh of 1300 quadrilateral bilinear elements, where following a three level refinement with a refinement tolerance set as $10^{-5} < \xi_K^2 < 10^{-4}$ we reach an average of 10 220 elements. The hyper-ROM case in which we do not refine the mesh is discretized using a mesh of 10 476 quadrilateral bilinear elements.



(a) Velocity contours. (b) Temperature contours. (c) AMR hyper-ROM ξ_K^2.

Fig. 19. Comparison of contour plots for FOM (left) and AMR hyper-ROM (right) at $t = 10$. AMR hyper-ROM ξ_K^2 and refined mesh at $t = 10$.

Fig. 19 shows a comparison of velocity and temperature contour plots between the FOM and the AMR hyper-ROM, and the refined mesh and the calculated ξ_K^2 at the time $t = 10$.

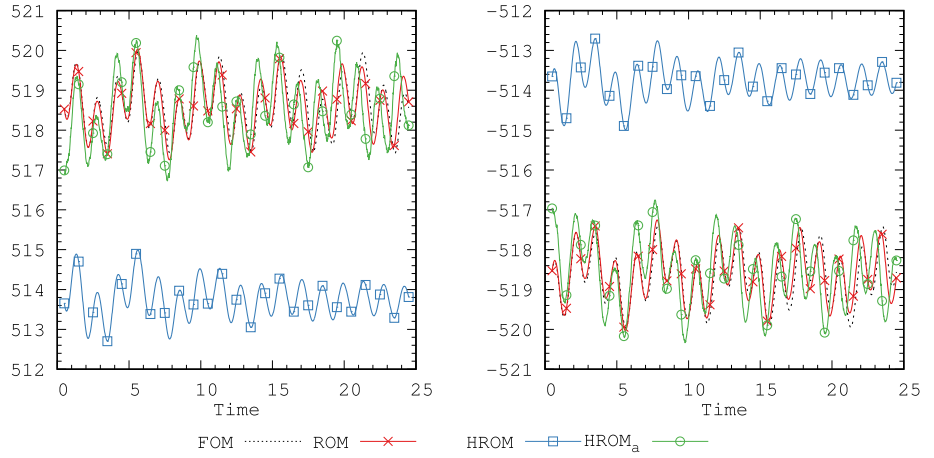
To compare the results and the performance of the AMR, we use the Nusselt number over the hot and cold walls, defined as

$$\mathrm{Nu}(t) = \left| \frac{1}{T_h - T_c} \int_0^H \frac{\partial T}{\partial x} \, \mathrm{d}y \right|, \tag{29}$$

with $\frac{\partial T}{\partial x}$ evaluated at either $x = 0$ (hot wall) or $x = W$ (cold wall). Fig. 20 compares the Nusselt number for FOM, ROM, hyper-ROM and AMR hyper-ROM (labeled $\mathrm{HROM_a}$), and Fig. 21 shows the Fourier transform of the Nusselt number for the same cases.

To measure the behavior of the formulation over the dynamic Nussel number, we also choose a RMSD of the ROM solution with respect to the FOM. With $\mathrm{Nu}_{j,\mathrm{FOM}}$ defined as the Nusselt number obtained with the FOM at time t^j, $j = 1, \ldots, S$, and $\mathrm{Nu}_{j,\mathrm{ROM}}$ the one obtained with any of the ROMs, we define the RMSD of the Nusselt number as

$$\mathrm{Nu}_{\mathrm{RMSD}} = \sqrt{\frac{1}{S} \sum_{j=1}^{S} \left(\mathrm{Nu}_{j,\mathrm{ROM}} - \mathrm{Nu}_{j,\mathrm{FOM}} \right)^2}.$$



(a) Hot wall. (b) Cold wall.

Fig. 20. Comparison of the Nusselt number between FOM, ROM, hyper-ROM and AMR hyper-ROM at $t = 10$.

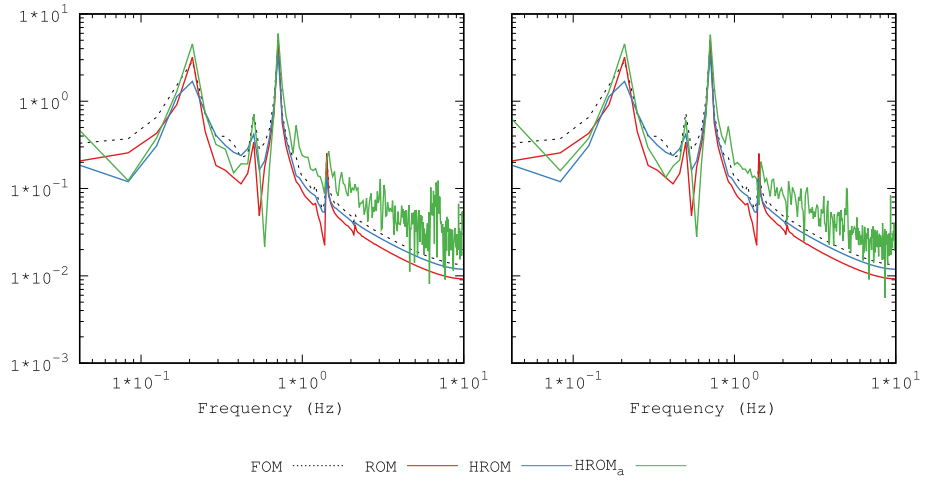


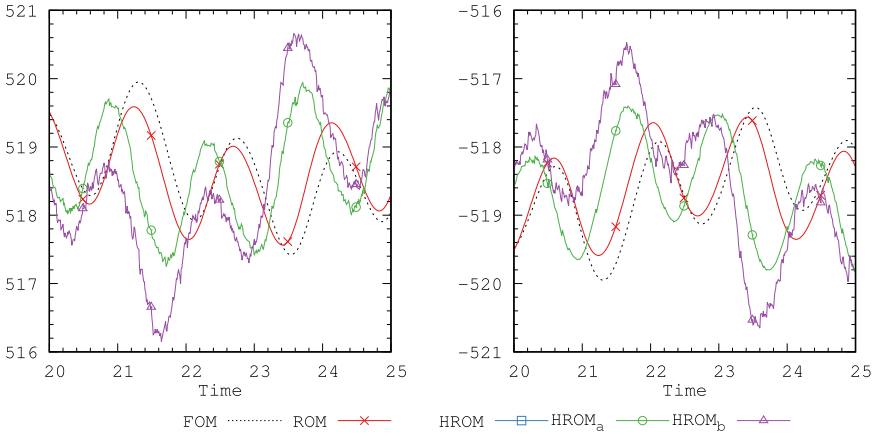
Fig. 21. Fourier transform of the Nusselt number for FOM, ROM, hyper-ROM and AMR hyper-ROM. Hot (left) and cold (right) walls.

Table 2
$\mathrm{Nu}_{\mathrm{RMSD}}$, ξ^2 and $\overline{\mathrm{Nu}}$ for ROM, hyper-ROM and AMR hyper-ROM.

	ROM		H-ROM		AMR H-ROM	
	Hot	Cold	Hot	Cold	Hot	Cold
$\mathrm{Nu}_{\mathrm{RMSD}}$	0.1292	0.1292	3.0228	3.0228	0.5452	0.5329
$\overline{\mathrm{Nu}}$	321.91	−321.91	318.91	−318.91	321.84	−321.86
$\overline{\xi^2}$	0.3786		0.5956		0.4895	

Table 2 shows the RMSD of the Nusselt number $\mathrm{Nu}_{\mathrm{RMSD}}$, the time average of the SGS L^2-norm $\overline{\xi^2}$ and the discrete time average of the Nusselt number $\overline{\mathrm{Nu}}$ for ROM, hyper-ROM and AMR hyper-ROM.

To illustrate the importance of the boundary subscales while obtaining the SGS error estimator, in Fig. 22 we compare the Nusselt number for FOM, ROM, hyper-ROM, AMR hyper-ROM (labeled HROM_a), and an AMR hyper-ROM case where the error estimator is computed without the boundary subscales (labeled HROM_b). Here we see how the use of the boundary subscales is needed to stabilize the solution.



(a) Hot wall. (b) Cold wall.

Fig. 22. Comparison of the Nusselt number between FOM, ROM, hyper-ROM and AMR hyper-ROM at $t = 10$.

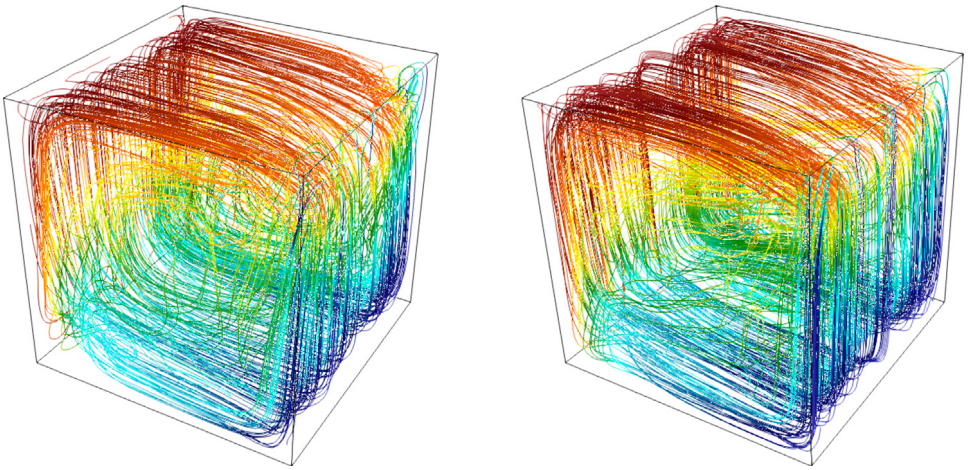


Fig. 23. Comparison of streamlines for FOM (left) and AMR hyper-ROM (right) at $t = 10$, colored depending on the temperature.

6.2.4. 3D differentially heated cavity

The last example consists in a 3D differentially heated cavity of aspect ratio 1 similar to the one solved in [34,43,44]. The computational domain is defined as $\Omega = [0, L] \times [0, L] \times [0, L]$ for the 3D problem, with $L = 1$. The temperatures on the walls perpendicular to the x-coordinate (horizontal) are fixed to $T_h = 960$ and $T_c = 240$, while adiabatic boundary conditions are prescribed in the remaining walls. No slip and impermeable conditions are set over all the walls, together with a homogeneous gravity acceleration $g = 12.035$ prescribed in the negative y-coordinate (vertical). The initial temperature is set to $T_0 = 600$, the density to $\rho_0 = 0.58841$, the dimensionless Prandtl number to $\text{Pr} = \frac{c_p \mu}{\lambda} = 0.71$ and the Rayleigh number to $\text{Ra} = \frac{\beta g \rho^2 c_p}{\lambda \mu}(T_h - T_c) = 3.55 \cdot 10^6$. As in the previous example the flow is considered an ideal gas with physical properties $R = 287.0$, $c_p = 1004.5$, $\mu = 0.0001$, $k = 1.4148$ and $\beta = 0.0001666$ (SI units).

For the FOM and ROM we use a uniform structured mesh composed of 64 000 regular hexahedral elements of size $h = 0.025$, from which we collect 500 snapshots for velocity, pressure and temperature at every 2 time steps in a 10 seconds interval. We solve the ROM, hyper-ROM and AMR hyper-ROM cases with a basis size of $r = 10$ and a retained energy $\eta = 0.99$. Fig. 23 shows a comparison of streamlines and temperature plots between the FOM and the AMR hyper-ROM.

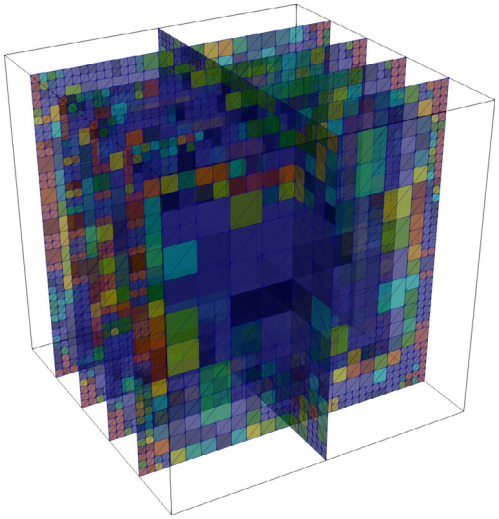


Fig. 24. AMR hyper-ROM ξ_K^2 at $t = 10$ with $5 \cdot 10^{-5} < \xi_K^2 < 5 \cdot 10^{-4}$.

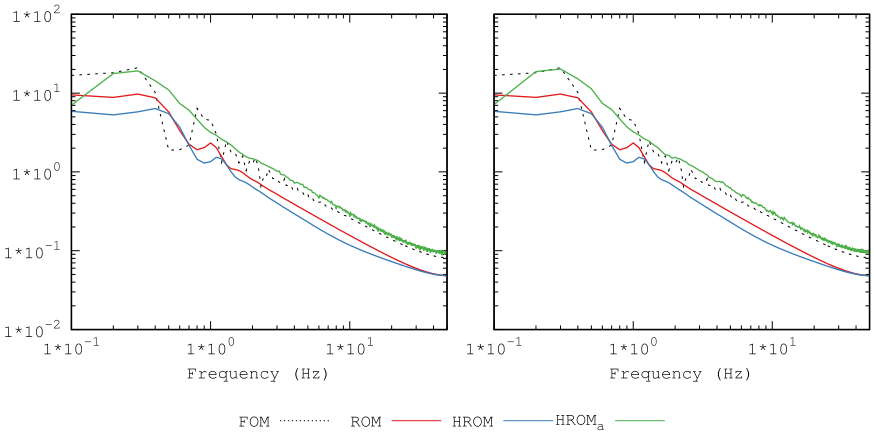


Fig. 25. Fourier transform of the Nusselt number for FOM, ROM, hyper-ROM and AMR hyper-ROM. Hot (left) and cold (right) walls.

We test the AMR hyper-ROM starting from a mesh of 125 regular hexahedral elements, where following a 3 level refinement with a refinement tolerance set as $5 \cdot 10^{-5} < \xi_K^2 < 5 \cdot 10^{-4}$ we reach an average of 16 393 elements. As in the previous examples we compare the AMR hyper-ROM with a fixed mesh hyper-ROM with 15 625 elements. Fig. 24 shows the refined mesh and the calculated ξ_K^2 at the time $t = 10$ for the AMR hyper-ROM.

In the same way as for the previous example, in Fig. 25 we show a comparison of the Fourier transform of the Nusselt number for the hot and cold walls. As expected the ROM and both hyper-ROM cases approximate adequately the FOM solution, with a better performance in the amplitude for the AMR hyper-reduction case.

7. Conclusions

In this paper we have developed a formulation for the element boundary subscales in VMS-ROMs. As done in [1] for the subscales in the interior of the elements, this formulation borrows the core ideas from the FE counterpart presented in [2]. We present two main techniques where we apply the boundary subscales formulation:

- Domain decomposition, where the boundary subscales are used as part of the transmission conditions between the subdomains, implemented using a DG approach and an iteration-by-subdomain scheme.
- Adaptive coarse mesh hyper-reduction, where the subscales —in the interior and in the boundaries of the elements— are used as error indicators for the AMR algorithm.

From the domain decomposition method we found that the ROM formulation behaves adequately regardless of the use of the subscales in the subdomain interface for a Neumann–Dirichlet approach, whereas for the DG approach the use of these subscales are essential to be able to obtain a solution. Further work on the iterative scheme and the subdomain interface formulation (stabilization and penalization constants in Eq. (20)) of the DG approach are needed to obtain a more adequate solution.

The AMR hyper-reduction behaves as expected, improving the quality of the solution when compared to a uniform mesh-based hyper-ROM, while retaining a similar computational cost. It is important to notice that at this point the efficiency of the model becomes heavily dependent of the efficiency of the refinement and interpolation or projection techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

R. Reyes acknowledges the scholarship received from COLCIENCIAS, from the Colombian Government. R. Codina acknowledges the support received from the ICREA, Spain Acadèmia Research Program, from the Catalan Government.

References

[1] R. Reyes, R. Codina, Projection-based reduced order models for flow problems: a variational multiscale approach, Comput. Methods Appl. Mech. Engrg. 363 (2020) 112844, http://dx.doi.org/10.1016/j.cma.2020.112844.

[2] R. Codina, J. Principe, J. Baiges, Subscales on the element boundaries in the variational two-scale finite element method, Comput. Methods Appl. Mech. Engrg. 198 (5–8) (2009) 838–852, http://dx.doi.org/10.1016/j.cma.2008.10.020.

[3] J. Baiges, R. Codina, S. Idelsohn, A domain decomposition strategy for reduced order models. application to the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 267 (2013) 23–42, http://dx.doi.org/10.1016/j.cma.2013.08.001.

[4] R. Codina, S. Badia, On the design of discontinuous Galerkin methods for elliptic problems based on hybrid formulations, Comput. Methods Appl. Mech. Engrg. 263 (2013) 158–168, http://dx.doi.org/10.1016/j.cma.2013.05.004.

[5] A. Hansbo, P. Hansbo, An unfitted finite element method, based on nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Engrg. 191 (47–48) (2002) 5537–5552, http://dx.doi.org/10.1016/S0045-7825(02)00524-8.

[6] F. Brezzi, B. Cockburn, L.D. Marini, E. Süli, Stabilization mechanisms in discontinuous Galerkin finite element methods, Comput. Methods Appl. Mech. Engrg. 195 (25–28) (2006) 3293–3310, http://dx.doi.org/10.1016/j.cma.2005.06.015.

[7] R. Codina, J. Baiges, Finite element approximation of transmission conditions in fluids and solids introducing boundary subgrid scales, Internat. J. Numer. Methods Engrg. 87 (1–5) (2011) 386–411, http://dx.doi.org/10.1002/nme.3111.

[8] D.J. Lucia, P.S. Beran, Projection methods for reduced order models of compressible flows, J. Comput. Phys. 188 (1) (2003) 252–280, http://dx.doi.org/10.1016/S0021-9991(03)00166-9.

[9] M. Buffoni, H. Telib, A. Iollo, Iterative methods for model reduction by domain decomposition, Comput. & Fluids 38 (6) (2009) 1160–1167, http://dx.doi.org/10.1016/j.compfluid.2008.11.008.

[10] A. Corigliano, F. Confalonieri, M. Dossi, S. Mariani, Coupled domain decomposition–proper orthogonal decomposition methods for the simulation of quasi-brittle fracture processes, Adv. Model. Simul. Eng. Sci. 3 (1) (2016) http://dx.doi.org/10.1186/s40323-016-0081-9.

[11] A. Tello, R. Codina, J. Baiges, Fluid structure interaction by means of variational multiscale reduced order models, Internat. J. Numer. Methods Engrg. (2020) http://dx.doi.org/10.1002/nme.6321.

[12] I. Martini, G. Rozza, B. Haasdonk, Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-darcy system, Adv. Comput. Math. 41 (5) (2015) 1131–1157, http://dx.doi.org/10.1007/s10444-014-9396-6.

[13] D. Amsallem, M.J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, Internat. J. Numer. Methods Engrg. 92 (10) (2012) 891–916, http://dx.doi.org/10.1002/nme.4371.

[14] L. Iapichino, A. Quarteroni, G. Rozza, A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks, Comput. Methods Appl. Mech. Engrg. 221–222 (2012) 63–82, http://dx.doi.org/10.1016/j.cma.2012.02.005.

[15] M. Ohlberger, F. Schindler, Error control for the localized reduced basis multiscale method with adaptive on-line enrichment, SIAM J. Sci. Comput. 37 (6) (2015) A2865–A2895, http://dx.doi.org/10.1137/151003660.

[16] L. Iapichino, A. Quarteroni, G. Rozza, Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries, Comput. Math. Appl. 71 (1) (2016) 408–430, http://dx.doi.org/10.1016/j.camwa.2015.12.001.

[17] J.L. Eftang, A.T. Patera, E.M. Rønquist, An "hp" certified reduced basis method for parametrized elliptic partial differential equations, SIAM J. Sci. Comput. 32 (6) (2010) 3170–3200, http://dx.doi.org/10.1137/090780122.

[18] B. Haasdonk, M. Dihlmann, M. Ohlberger, A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space, Math. Comput. Model. Dyn. Syst. 17 (4) (2011) 423–442, http://dx.doi.org/10.1080/13873954.2011.547674.

[19] S. Ullmann, M. Rotkvic, J. Lang, POD-Galerkin Reduced-order modeling with adaptive finite element snapshots, J. Comput. Phys. 325 (2016) 244–258, http://dx.doi.org/10.1016/j.jcp.2016.08.018.
[20] B. Peherstorfer, K. Willcox, Online adaptive model reduction for nonlinear systems via low-rank updates, SIAM J. Sci. Comput. 37 (4) (2015) A2123–A2150, http://dx.doi.org/10.1137/140989169.
[21] B. Peherstorfer, K. Willcox, Dynamic data-driven model reduction: adapting reduced models from incomplete data, Adv. Model. Simul. Eng. Sci. 3 (1) (2016) 11, http://dx.doi.org/10.1186/s40323-016-0064-x.
[22] A. Thari, V. Pasquariello, N. Aage, S. Hickel, Adaptive reduced-order modeling for non-linear fluid-structure interaction, arXiv e-print (2017) arXiv:1702.04332.
[23] K. Carlberg, Adaptive h -refinement for reduced-order models, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 1192–1210, http://dx.doi.org/10.1002/nme.4800.
[24] P.A. Etter, K.T. Carlberg, Online adaptive basis refinement and compression for reduced-order models via vector-space sieving, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112931, http://dx.doi.org/10.1016/j.cma.2020.112931.
[25] J.S. Hale, E. Schenone, D. Baroli, L. Beex, S. Bordas, A hyper-reduction method using adaptivity to cut the assembly costs of reduced order models, 2019, e-print.
[26] G. Hauke, M.H. Doweidar, M. Miana, The multiscale approach to error estimation and adaptivity, Comput. Methods Appl. Mech. Engrg. 195 (13–16) (2006) 1573–1593, http://dx.doi.org/10.1016/j.cma.2005.05.029.
[27] G. Hauke, D. Fuster, M.H. Doweidar, Variational multiscale a-posteriori error estimation for multi-dimensional transport problems, Comput. Methods Appl. Mech. Engrg. 197 (33–40) (2008) 2701–2718, http://dx.doi.org/10.1016/j.cma.2007.12.022.
[28] G. Hauke, D. Fuster, F. Lizarraga, Variational multiscale a posteriori error estimation for systems: The euler and Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 283 (2015) 1493–1524, http://dx.doi.org/10.1016/j.cma.2014.10.032.
[29] J. Baiges, R. Codina, Variational multiscale error estimators for solid mechanics adaptive simulations: An orthogonal subgrid scale approach, Comput. Methods Appl. Mech. Engrg. 325 (2017) 37–55, http://dx.doi.org/10.1016/j.cma.2017.07.008.
[30] C. Bayona, R. Codina, J. Baiges, Variational multiscale error estimators for the adaptive mesh refinement of compressible flow simulations, Comput. Methods Appl. Mech. Engrg. 337 (2018) 501–526, http://dx.doi.org/10.1016/j.cma.2018.03.010.
[31] J. Baiges, C. Bayona, Refficientlib: An efficient load-rebalanced adaptive mesh refinement algorithm for high-performance computational physics meshes, SIAM J. Sci. Comput. 39 (2) (2017) C65–C95, http://dx.doi.org/10.1137/15M105330X.
[32] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Engrg. 191 (39–40) (2002) 4295–4321, http://dx.doi.org/10.1016/S0045-7825(02)00337-7.
[33] R. Codina, S. Badia, J. Baiges, J. Principe, Variational multiscale methods in computational fluid dynamics, in: Encyclopedia of Computational Mechanics Second Edition, John Wiley & Sons, Ltd, Chichester, UK, 2018, pp. 1–28, http://dx.doi.org/10.1002/9781119176817.ecm2117.
[34] R. Reyes, R. Codina, J. Baiges, S. Idelsohn, Reduced order models for thermally coupled low mach flows, Adv. Model. Simul. Eng. Sci. 5 (1) (2018) 28, http://dx.doi.org/10.1186/s40323-018-0122-7.
[35] R. Codina, J. Principe, Dynamic subscales in the finite element approximation of thermally coupled incompressible flows, Internat. J. Numer. Methods Fluids 54 (6–8) (2007) 707–730, http://dx.doi.org/10.1002/fld.1481.
[36] L. El Alaoui, A. Ern, Residual and hierarchical a posteriori error estimates for nonconforming mixed finite element methods, ESAIM Math. Model. Numer. Anal. 38 (6) (2004) 903–929, http://dx.doi.org/10.1051/m2an:2004044.
[37] E. Burman, M.A. Fernández, P. Hansbo, Continuous interior penalty finite element method for Oseen's Equations, SIAM J. Numer. Anal. 44 (3) (2006) 1248–1274, http://dx.doi.org/10.1137/040617686.
[38] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity, Internat. J. Numer. Methods Engrg. 33 (7) (1992) 1365–1382, http://dx.doi.org/10.1002/nme.1620330703.
[39] J. Baiges, R. Codina, S. Idelsohn, Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier-Stokes equations, Internat. J. Numer. Methods Fluids 72 (12) (2013) 1219–1243, http://dx.doi.org/10.1002/fld.3777.
[40] O. Guasch, R. Codina, Statistical behavior of the orthogonal subgrid scale stabilization terms in the finite element large eddy simulation of turbulent flows, Comput. Methods Appl. Mech. Engrg. 261–262 (2013) 154–166, http://dx.doi.org/10.1016/j.cma.2013.04.006.
[41] M.a. Christon, P.M. Gresho, S.B. Sutton, Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution), Internat. J. Numer. Methods Fluids 40 (8) (2002) 953–980, http://dx.doi.org/10.1002/fld.395.
[42] R. Codina, J. Principe, M. Avila, Finite element approximation of turbulent thermally coupled incompressible flows with numerical sub-grid scale modelling, Int. J. Numer. Methods Heat Fluid Flow 20 (2010) 492–516, http://dx.doi.org/10.1108/09615531011048213.
[43] P. Le Quéré, C. Weisman, H. Paillère, J. Vierendeels, E. Dick, R. Becker, M. Braack, J. Locke, Modelling of natural convection flows with large temperature differences: A benchmark problem for low mach number solvers. part 1. reference solutions, ESAIM Math. Model. Numer. Anal. 39 (3) (2005) 609–616, http://dx.doi.org/10.1051/m2an:2005027.
[44] M. Avila, J. Principe, R. Codina, A finite element dynamical nonlinear subscale approximation for the low mach number flow equations, J. Comput. Phys. 230 (22) (2011) 7988–8009, http://dx.doi.org/10.1016/j.jcp.2011.06.032.