Adv Comput Math (2023) 49:21
https://doi.org/10.1007/s10444-022-10008-w

Efficient and certified solution of parametrized one-way coupled problems through DEIM-based data projection across non-conforming interfaces

Elena Zappon[1] · Andrea Manzoni[1] · Alfio Quarteroni[1,2]

Received: 26 April 2022 / Accepted: 15 December 2022 / Published online: 16 March 2023

Abstract

One of the major challenges of coupled problems is to manage nonconforming meshes at the interface between two models and/or domains, due to different numerical schemes or domain discretizations employed. Moreover, very often complex submodels depend on (e.g., physical or geometrical) parameters, thus making the repeated solutions of the coupled problem through high-fidelity, full-order models extremely expensive, if not unaffordable. In this paper, we propose a reduced order modeling (ROM) strategy to tackle parametrized one-way coupled problems made by a first, *master* model and a second, *slave* model; this latter depends on the former through Dirichlet interface conditions. We combine a reduced basis method, applied to each subproblem, with the discrete empirical interpolation method to efficiently interpolate or project Dirichlet data across either conforming or non-conforming meshes at the domains interface, building a low-dimensional representation of the overall coupled problem. The proposed technique is numerically verified by considering a series of test cases involving both steady and unsteady problems, after deriving a posteriori error estimates on the solution of the coupled problem in both cases. This work arises from the need to solve staggered cardiac electrophysiological models and represents the first step towards the setting of ROM techniques for the more general two-way Dirichlet-Neumann coupled problems solved with domain decomposition sub-structuring methods, when interface non-conformity is involved.

Keywords Coupled problems · Reduced order models · Proper orthogonal decomposition · Discrete empirical interpolation · Interface non-conformity · A posteriori error estimates

Communicated by: Olga Mula

Elena Zappon
elena.zappon@polimi.it

Extended author information available on the last page of the article.

Mathematics Subject Classification (2010) 65M60 · 65N99 · 68U99

1 Introduction

Fast simulation techniques for multi-scale and multi-physics problems are nowadays of key relevance in many applications from engineering including, among others, fluid-structure interaction (FSI) and electro-mechanical (EM) couplings [1–7]. Often, these problems are characterized by two (or more) nonlinear partial differential equations (PDEs) representing detailed parametric physical systems interacting through their boundaries. When a full-order model (FOM) based, e.g., on the finite element (FE) method, is used, solving accurately such coupled systems becomes computationally demanding, especially when strong constraints on spatial mesh sizes and/or time steps must be imposed to deal with steep fronts solutions or fast dynamics.

In several cases, such coupled problems describe complex phenomena involving a set of physical and/or geometrical input parameters, so that a detailed characterization of input-output maps requires the repeated solution of the problem at hand [8–14]. In these cases, numerical simulations carried out by high-fidelity FOMs may easily become out of reach. In addition, when dealing with nonconforming meshes at the interface, special techniques, e.g., MORTAR method and INTERNODES [15–20], need to be employed, making the underlying FOM even more computationally demanding.

In these contexts, efficient reduced order models (ROMs) can be successfully applied to decrease the overall computational costs. Preliminary studies on reduced coupled problems, especially in a FSI context, were carried out in [21–25], where ROMs were applied to one or both subproblems, considering splitting schemes to handle the model coupling, and in [26], where POD was used to reduce an FSI problem solved with a monolithic FE scheme. In all these cases, geometrical and numerical interface conformity between subdomains was necessary to set up the numerical schemes. Domain decomposition (DD) techniques coupled with reduced basis (RB) methods [27–29], instead, have been explored in several works, e.g., [30–33]. Their goal is to solve expensive models set on involved domains by splitting the considered geometries in a set of building blocks, often exploiting topological similarity, and applying the reduction locally, on each building block. The final global approximation is then computed gluing together the local solutions through different techniques, e.g., Lagrangian multipliers or by Fourier basis functions. Moreover, RB-DD methods have been considered to construct efficient preconditioners [34] or to perform static condensation [35, 36].

In this paper, we propose a RB method to solve efficiently one-way coupled problems. Precisely, we consider parametric (either elliptic or parabolic) problems defined on two domains with a common interface. The first *master* model is solved as an independent model, imposing homogeneous Neumann boundary conditions at the interface; the second *slave* model is dependent from the master model through Dirichlet interface conditions. Note that Dirichlet interface conditions naturally inherit the parameter dependency from the master model. As high-fidelity FOM, we consider the finite element (FE) method involving either conforming or non-conforming discretizations at the interface across the domains. In the former case, the slave solution at the interface corresponds to the master one; in the latter, the slave interface solution must be computed with an interpolation method, e.g., relying on radial basis functions.

Then, we implement RB methods to reduce completely the parametric one-way coupled problem, including the interface conditions. To this end, we consider a modular approach: the two sub-problems are reduced independently through a POD-Galerkin approach, while interface data are handled by setting a further efficient interpolation — or projection — stage relying on the discrete empirical interpolation method (DEIM) [37–41]. In particular, we show how to use this reduction paradigm to transfer Dirichlet data across both conforming and non-conforming domain interfaces, effectively replacing any high fidelity interpolation technique in the reduced problem with the DEIM.

Differently from previous works, this new approach is able to decrease the overall computational costs of solving parametric one-way coupled problems through RB methods, including the possibility to consider interface grid non-conformity. The interface Dirichlet data can be easily transferred between the two problems without implementing other techniques that could increase the algebraic system dimensions, as, e.g., in the case of the Lagrange multipliers. Furthermore, several test cases and the derived a posteriori error estimate show that this modular approach allows to have full control of the solution accuracy at each reduction step, and to tailor the corresponding ROM on every sub-problems. The proposed technique can be seen as a reduced form of the DD method applied to one-way coupled problems, in which the two submodels are handled independently and solved sequentially.

The present work arises from the need of solving staggered cardiac electrophysiological models, i.e., bidomain-torso models [42], and is a preliminary investigation toward the solution of a more general (and challenging) parametrized two-way coupled problem obtained through Dirichlet-Neumann interface conditions. Therefore, the described modular ROM can be extended to the case in which DD sub-structuring methods [20, 43] are applied to steady or unsteady two-way coupled problems, in case of both conforming or non-conforming interface grids; this will be the focus of a forthcoming manuscript.

The structure of the paper is as follows: after formulating parametrized one-way coupled problems in Sections 2, 3 and 4 are devoted to their high fidelity discretization and their reduced order modeling, respectively. The treatment of the interface data is postponed to Section 5. Section 6 presents a detailed derivation of a posteriori error estimates for the proposed technique. FOM and ROM solutions are then compared on a series of numerical benchmarks in Section 7; conclusions and perspectives then follow in Section 8.

2 Problem formulation

In this section, we introduce an algebraic formulation of the parametrized one-way coupled problems, assuming to deal with the finite element method as high-fidelity

FOM, despite the proposed strategy is independent of the employed FOM. Let us consider a d-dimensional domain ($d = 2,3$) partitioned into two non-overlapping subdomains Ω_1 and Ω_2, sharing the interface $\Gamma := \overline{\Omega}_1 \cap \overline{\Omega}_2$, and denote by $\mathbf{n}_i$, $i = 1, 2$ the outer unit normal directions of the two domains with respect to Γ. Hereon, to simplify the notation, we call *master model* the problem set in Ω_1, and *slave model* the one set in Ω_2; correspondingly, we refer to their solutions as to the *master solution* and the *slave solution*, respectively.

2.1 Steady case

We first consider a steady coupled problem; the unsteady counterpart will be described in the following subsection. Given two input parameter vectors $\boldsymbol{\mu}_1 \in \mathscr{P}^{d_1}$ and $\boldsymbol{\mu}_2 \in \mathscr{P}^{d_2}$, $d_1, d_2 \geq 1$, and two functions $f_1(\boldsymbol{\mu}_1)$ and $f_2(\boldsymbol{\mu}_2)$ defined on Ω_i, $i = 1, 2$ respectively, we look for u_1 and u_2 such that

$$\begin{cases} \mathcal{L}_1(\boldsymbol{\mu}_1)u_1(\boldsymbol{\mu}_1) = f_1 & \text{in } \Omega_1 \\ \mathrm{BCs}(\boldsymbol{\mu}_1) & \text{on } \partial\Omega_1 \setminus \Gamma \\ \frac{\partial u_1(\boldsymbol{\mu}_1)}{\partial n} = 0 & \text{on } \Gamma, \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{L}_2(\boldsymbol{\mu}_2)u_2(\boldsymbol{\mu}_2) = f_2 & \text{in } \Omega_2 \\ \mathrm{BCs}(\boldsymbol{\mu}_2) & \text{on } \partial\Omega_2 \setminus \Gamma \\ u_2(\boldsymbol{\mu}_2) = u_1(\boldsymbol{\mu}_1) & \text{on } \Gamma. \end{cases} \tag{1}$$

The two problems in (1) can represent either two different physical systems, or the multi-domain form of the same system, where $\mathcal{L}_1$ and $\mathcal{L}_2$ denote two parametrized second order elliptic operators. The master model is made independent from the slave one imposing homogeneous Neumann boundary conditions on Γ, while Dirichlet boundary conditions

$$u_2(\boldsymbol{\mu}_2) = u_1(\boldsymbol{\mu}_1) \quad \text{on } \Gamma, \tag{2}$$

are applied at the interface to ensure the continuity of the solution from the master to the slave model. One-way coupled problems such as this can arise, e.g., from a partitioned scheme's splitting operation on a two-way coupled problem featured by Dirichlet-Neumann interface conditions.

The high-fidelity FOM would then require to *(i)* solve the master model with FE method, then *(ii)* extract the master solution at the domains' interface and *(iii)* use the data obtained in *(ii)* as Dirichlet boundary conditions to solve the slave model with FE method. This procedure is affordable if the grids used are conforming at the interface of the two domains; however, it usually involves the application of methods such as the MORTAR [15, 19, 20, 44] or the INTERNODES [17, 18, 45, 46] methods in the non-conforming case, which might become very expensive especially for three-dimensional domains.

Given the parameterized nature of the problem, the solutions u_i can be seen as two maps $u_1 : \mathscr{P}^{d_1} \to V_1$ and $u_2 : \mathscr{P}^{d_2} \to V_2$ that to any $\boldsymbol{\mu}_1 \in \mathscr{P}^{d_1}$ and $\boldsymbol{\mu}_2 \in \mathscr{P}^{d_2}$ associate the solutions $u_1(\boldsymbol{\mu}_1)$ and $u_2(\boldsymbol{\mu}_2)$ in V_1 and V_2, where

$$V_i = H^1_{\partial\Omega_{i,D}}(\Omega_i) := \{v \in H^1(\Omega_i) : \ v_{|\partial\Omega_{i,D}} = 0\}, \ i = 1, 2; \tag{3}$$

here $\partial\Omega_{i,D}$ are suitable disjoint subsets of $\partial\Omega_i \setminus \Gamma$ for Dirichlet boundary conditions of problems (1), respectively. Hereon, we will consider only homogeneous Neumann boundary conditions on the $\partial\Omega_i \setminus \Gamma$ while, in presence of Dirichlet boundary conditions, a lifting technique can be applied. We will assume that the solution of (1) exists and is unique for each $\boldsymbol{\mu}_i \in \mathscr{P}^{d_i}$.

2.2 Unsteady case

With the same notation of (1), we consider the following time-dependent problem: assuming that $\boldsymbol{\mu}_1 \in \mathscr{P}^{d_1}$ and $\boldsymbol{\mu}_2 \in \mathscr{P}^{d_2}$, $d_1, d_2 \geq 1$ are two set of time-independent parameters, $f_1(t; \boldsymbol{\mu}_1)$, $f_2(t; \boldsymbol{\mu}_2)$ are time-dependent functions defined on $\Omega_i \times (0, T)$, $i = 1, 2$ respectively, and $\mathcal{L}_1$ and $\mathcal{L}_2$ are second-order elliptic operators, we look for $u_1(t; \boldsymbol{\mu}_1)$ and $u_2(t; \boldsymbol{\mu}_2)$ such that

$$\begin{cases} \dfrac{\partial u_1(t; \boldsymbol{\mu}_1)}{\partial t} + \mathcal{L}_1(\boldsymbol{\mu}_1) u_1(t; \boldsymbol{\mu}_1) = f_1(t; \boldsymbol{\mu}_1) & \text{in } \Omega_1 \times (0, T) \\ \mathrm{BCs}(t; \boldsymbol{\mu}_1) & \text{on } \partial\Omega_1 \setminus \Gamma \times (0, T) \\ u_1(0; \boldsymbol{\mu}_1) = u_{1,0}(\boldsymbol{\mu}_1) & \text{on } \Omega_1 \times \{0\}, \end{cases} \tag{4}$$

$$\begin{cases} \dfrac{\partial u_2(t; \boldsymbol{\mu}_2)}{\partial t} + \mathcal{L}_2(\boldsymbol{\mu}_2) u_2(t; \boldsymbol{\mu}_2) = f_2(t; \boldsymbol{\mu}_2) & \text{in } \Omega_2 \times (0, T) \\ \mathrm{BCs}(t; \boldsymbol{\mu}_2) & \text{on } \partial\Omega_2 \setminus \Gamma \times (0, T) \\ \mathbf{u}_2(0; \boldsymbol{\mu}_2) = u_{2,0}(\boldsymbol{\mu}_2) & \text{on } \Omega_2 \times \{0\}, \end{cases} \tag{5}$$

where $T > 0$ represent the final time. As in Section 2.1, we consider homogeneous Neumann interface conditions for the master models and Dirichlet interface conditions as coupling conditions on the slave model:

$$u_2(t; \boldsymbol{\mu}_2) = u_1(t; \boldsymbol{\mu}_1) \qquad \text{on } \Gamma.$$

Note that the time variable t has been added to account for the time dependency of the solution, while the spatial variable $\mathbf{x}$ is implicitly considered.

Remark 1 The differential operators $\mathcal{L}_1$ and $\mathcal{L}_2$ here considered are time independent; however, the method presented in this work can be easily adapted to the case of time-dependent partial differential operators, too. Note that an unsteady coupled problem can also be obtained coupling a time-dependent problem in Ω_1 and a time-independent problem in Ω_2, or viceversa. For example, if a time-dependent model is one-way coupled with a time-independent model, the slave model becomes time-dependent through the Dirichlet boundary conditions received from the master model (see the test case *ii* of Section 7.2).

Similar considerations on the resolution affordability can be stated as for the steady case. Time-dependent models such as (4)– (5) could be solved computing, for each time instant, *(i)* the master solution, *(ii)* extracting the Dirichlet interface conditions from the master model, and *(iii)* using them to solve the slave model. Also in this case, we set ROM strategies for unsteady models starting from their discretized form with FE methods. As before, Neumann terms can appear in the formulation according to the problem considered, while in presence of Dirichlet boundary conditions, the lifting method can be implemented. Hereon, we assume that the solution of the proposed problems exists and is unique for each parameter instance and each time $t > 0$.

3 High fidelity discretization

The RB method proposed in this work aims at reducing the computational costs of solving a parametrized one-way coupled problem, for instance, when multiple queries with different parameters values are required; in principle, it can be applied in case of both conforming or non-conforming grids at the domains interface. For the sake of generality, here we consider interface non-conformity through two a priori independent discretizations on the two domains, with two families of triangulations $\mathcal{T}_{h_1} = \cup_m T_{1,m}$ in Ω_1 and $\mathcal{T}_{h_2} = \cup_m T_{2,m}$ in Ω_2, respectively. For instance, one of the two meshes can be made by simplices (triangles or tetrahedra) and the other by quads (quadrilaterals of hexahedra), or both can be made by the same kind of elements, however featuring different mesh sizes h_1 and h_2. Moreover, different polynomial degrees p_1 and p_2 can be used to define FE spaces. Hereon we will restrict ourselves to quads since they are the elements used in all the test cases presented in Section 7.

Additionally, we denote by Γ_1 and Γ_2 the internal interfaces of Ω_1 and Ω_2, respectively, induced by the triangulations $\mathcal{T}_{h_1}$ and $\mathcal{T}_{h_2}$. Note that $\Gamma_1 = \Gamma_2 = \Gamma$ if the interface Γ is a straight segment (for $d = 2$) or a plane (for $d = 3$), otherwise Γ_1 and Γ_2 can also be different. In case of non-conforming grids, we have signal interpolation if $\Gamma_1 = \Gamma_2$ or signal projection if $\Gamma_1 \neq \Gamma_2$ (see Fig. 1).

3.1 Steady case

In the steady case, for each partition $\mathcal{T}_{h_i}$, we first define the FE spaces

$$X_{h_i}^{q_i} = \{v \in C^0(\overline{\Omega_i}) : v_{|T_{i,m}} \in \mathcal{Q}_{q_i}, \ \forall T_{i,m} \in \mathcal{T}_{h_i}\}, \quad i = 1, 2 \tag{6}$$

where q_i is a chosen integer and $\mathcal{Q}_{q_i}$ represents the elements of the quads space, and then introduce the following finite dimensional trial spaces

$$V_{h_i} = \{v \in X_{h_i}^{q_i} : v_{|\partial\Omega_{i,D}} = 0\}, \quad i = 1, 2, \tag{7}$$

of dimension N_i, $i = 1, 2$. Using a Galerkin-FE approximation, each of the two problems in (1) yields a parametrized linear system

$$\mathbb{A}_{N_i}(\boldsymbol{\mu}_i)\mathbf{u}_i(\boldsymbol{\mu}_i) = \mathbf{f}_{N_i}(\boldsymbol{\mu}_i), \tag{8}$$

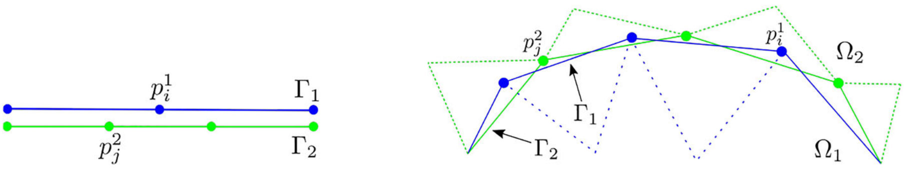

Fig. 1 Γ_1 and Γ_2 obtained through the triangulations $\mathcal{T}_{1,h_1}$ and $\mathcal{T}_{2,h_2}$. On the left, since $\Gamma_1 = \Gamma_2 = \Gamma$, we do an interpolation; on the right, since $\Gamma_1 \neq \Gamma_2$, we do a projection of the Dirichlet interface matching

where $\mathbb{A}_{N_i}(\boldsymbol{\mu}_i) \in \mathbb{R}^{N_i \times N_i}$ denotes the stiffness matrix, and $\mathbf{f}_{N_i}(\boldsymbol{\mu}_i) \in \mathbb{R}^{N_i}$ is the right-hand side vector, $i = 1, 2$. For the sake of space, we do not report the detailed derivation of these systems, which is indeed quite standard [47]. Note that we exploit the one-to-one correspondence between functions in $X_{h_i}^{q_i}$ and vectors in $\mathbb{R}^{N_i}$, as usually done when dealing with the Galerkin-FE method; here $\mathbf{u}_i(\boldsymbol{\mu}_i) \in \mathbb{R}^{N_i}$ denotes the vector of degrees of freedom of the approximation $u_{i,h}(\boldsymbol{\mu}_i) \in X_{h_i}^{q_i}$ of $u_i(\boldsymbol{\mu}_i)$.

Finally, at the interface, we prescribe Dirichlet conditions that read as

$$\mathbf{u}_{2_{|\Gamma}}(\boldsymbol{\mu}_2) = \mathbf{u}_{1_{|\Gamma}}(\boldsymbol{\mu}_1)$$

if the two interfaces are conforming, or

$$\mathbf{u}_{2_{|\Gamma_2}}(\boldsymbol{\mu}_2) = \Pi \mathbf{u}_{1_{|\Gamma_1}}(\boldsymbol{\mu}_1)$$

in the case of nonconforming interfaces, where Π is a suitable interpolation operator. For instance, in Section 7, Π will be a linear interpolation operator, i.e., we will interpolate high fidelity Dirichlet interface data through a linear interpolation algorithm based on VTK subroutines [48].

Remark 2 Note that the reduced order technique here presented can be applied regardless of the high fidelity interpolation scheme considered. Moreover, in the online ROM computations, such high fidelity interpolation is replaced by DEIM. Thus, in principle, any interpolation method can be considered in the high fidelity formulation, according to the precision required by the solution, e.g., the INTERNODES method define Π by means of rescaled localized radial basis functions (RL-RBF) [18, 49].

3.2 Unsteady case

The spatial discretization of the time-dependent problems (4)–(5) yields a dynamical system of the form

$$\begin{cases} \mathbb{M}_{N_i} \dfrac{d}{dt}\mathbf{u}_i(t; \boldsymbol{\mu}_i) + \mathbb{A}_{N_i}(\boldsymbol{\mu}_i)\mathbf{u}_i(t; \boldsymbol{\mu}_i) = \mathbf{f}_{N_i}(t; \boldsymbol{\mu}_i), \qquad t \in (0, T) \\ \mathbf{u}_i(0; \boldsymbol{\mu}_i) = \mathbf{u}_{i,0}(\boldsymbol{\mu}_i) \end{cases} \tag{9}$$

for $i = 1, 2$, where $\mathbb{M}_{N_i} \in \mathbb{R}^{N_i \times N_i}$ is the mass matrix related to the domain Ω_i and $\mathbf{u}_{i,0}(\boldsymbol{\mu}_i)$ is the vector of degrees of freedom of the approximation of the initial data $u_{i,0}(\boldsymbol{\mu}_i)$; see, e.g., [20] for further details.

Once discretized in space, time discretization can then be performed according to different numerical schemes [20, 50, 51], such as, e.g., backward differentiation formulas (BDF). Introducing a uniform partition of $(0, T)$ in N_t subintervals, each of length $\Delta t = T/N_t$, and denoting by $t^n = n\Delta t$ the nth time instant, we can approximate the solution at time t^n as

$$\mathbf{u}_i^n(\boldsymbol{\mu}_i) \simeq \mathbf{u}_i(t^n; \boldsymbol{\mu}_i) \quad \forall n = 0, \ldots, N_t,$$

and the time derivative through the backward Euler formula as

$$\frac{\partial \mathbf{u}_i(t^{n+1}; \boldsymbol{\mu}_i)}{\partial t} \approx \frac{\mathbf{u}_i^{n+1}(\boldsymbol{\mu}_i) - \mathbf{u}_i^n(\boldsymbol{\mu}_i)}{\Delta t} \qquad \forall n = 0, \ldots, N_t - 1.$$

Remark 3 Although the same time discretization has been used in both subproblems. A staggered numerical scheme can be implemented, employing different time discretizations on the master and the slave sub-problems. However, if the time step selected for the slave problem is not equal to, or a multiple of, that used for the master problem, a further interpolation method has to be added at the interface to ensure the correct transmission of the Dirichlet interface conditions. For the sake of simplicity, in this work, we only account for the case in which the same time discretization is employed.

The high-fidelity FOM in the unsteady case thus reads: find $\mathbf{u}_i^{n+1}(\boldsymbol{\mu}_i) \in \mathbb{R}^{N_i}$ such that

$$\begin{cases} \left(\frac{\mathbb{M}_{N_i}}{\Delta t} + \mathbb{A}_{N_i}(\boldsymbol{\mu}_i)\right)\mathbf{u}_{N_i}^{n+1}(\boldsymbol{\mu}_i) = \mathbf{f}_{N_i}^{n+1}(\boldsymbol{\mu}_i) + \frac{\mathbb{M}_{N_i}}{\Delta t}\mathbf{u}_{N_i}^{n}(\boldsymbol{\mu}_i), \quad n = 0, \ldots, N_t - 1, \\ \mathbf{u}_{N_i}^0(\boldsymbol{\mu}_i) = \mathbf{u}_{N_i,0}(\boldsymbol{\mu}_i), \end{cases} \tag{10}$$

for $i = 1, 2$. Finally, we impose Dirichlet interface conditions, under the form

$$\mathbf{u}_{2_{|\Gamma}}^{n+1}(\boldsymbol{\mu}_2) = \mathbf{u}_{1_{|\Gamma}}^{n+1}(\boldsymbol{\mu}_1) \quad \forall n = 0, \ldots, N_t - 1$$

if the two interfaces are conforming, or

$$\mathbf{u}_{2_{|\Gamma_2}}^{n+1}(\boldsymbol{\mu}_2) = \Pi \mathbf{u}_{1_{|\Gamma_1}}^{n+1}(\boldsymbol{\mu}_1) \quad \forall n = 0, \ldots, N_t - 1$$

in the non-conforming interfaces case, where Π is a suitable interpolation operator.

4 Reduced order modeling

The proposed ROM strategy is a modular procedure aiming at reducing separately the three main parts of the one-way coupled problem considered, that is, the master model, the interface Dirichlet conditions, and the slave model, relying on a POD-Galerkin-DEIM approach [27–29]. For both the master and the slave models, the goal is to approximate the FOM solution by means of a reduced basis obtained from a set of FOM snapshots — that is, solutions obtained through the FOM for selected values of the parameters — while we seek for a low-dimensional representation of the parametrized Dirichlet data (see Section 5). Note that the reduction of the slave problem depends not only on the number of basis functions introduced to approximate its solution, but also on the chosen approximation of the interface data.

4.1 Steady case

Let us start by considering the steady case of Section 3.1. Since the slave model is dependent on non-homogeneous Dirichlet conditions at least at the interface, it is convenient to rely on a lifting technique. Therefore, we can express the slave solution as

$$\mathbf{u}_2(\boldsymbol{\mu}_2) = \tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) + \mathbf{u}_{2,D}(\boldsymbol{\mu}_2)$$

where $\mathbf{u}_{2,D}(\boldsymbol{\mu}_2)$ is the lifting vector such that $\mathbf{u}_{2|\partial\Omega_{2,D}}(\boldsymbol{\mu}_2) = \mathbf{u}_{2,D}(\boldsymbol{\mu}_2)$. Then, the FOM slave model can be replaced by an equivalent FOM involving Dirichlet boundary conditions, in which the contribution of Dirichlet data is moved to the right hand side: find $\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2)$ such that

$$\mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) = \mathbf{f}_{N_2}(\boldsymbol{\mu}_2) - \mathbb{A}_{N_2}\mathbf{u}_{2,D}(\boldsymbol{\mu}_2). \tag{11}$$

Following a standard POD-Galerkin approach (see, e.g., [29]), during the offline stage, we define two sets of snapshots $\mathbf{S}_1 = \{\mathbf{u}_1(\boldsymbol{\mu}_1^k),\ \boldsymbol{\mu}_1^k \in \mathscr{P}^{d_1}\}$, $\tilde{\mathbf{S}}_2 = \{\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2^k),\ \boldsymbol{\mu}_2^k \in \mathscr{P}^{d_2}\}$, for the master and the slave model, respectively, by solving the FOM problems (8) for suitably chosen parameter values (sampled, e.g., through latin hypercube sampling [52, 53]).

For each set of snapshots, we construct a global reduced basis through POD; the corresponding matrix $\mathbb{V}_i \in \mathbb{R}^{N_i \times n_i}$, $n_i \ll N_i$, $i = 1, 2$, collects, columnwise, the obtained basis functions. To define the ROM, we rely on a Galerkin-RB strategy, that is, we project the original FOM onto the reduced spaces defined by each $\mathbb{V}_i$, $i = 1, 2$. Then, in the online stage, the approximation of the master solution can be sought under the approximated form

$$\mathbf{u}_1(\boldsymbol{\mu}_1) \approx \mathbb{V}_1\mathbf{u}_{n_1}(\boldsymbol{\mu}_1), \tag{12}$$

where $\mathbf{u}_{n_1}(\boldsymbol{\mu}_1)$ is the solution of the *reduced master problem*

$$\mathbb{A}_{n_1}(\boldsymbol{\mu}_1)\mathbf{u}_{n_1}(\boldsymbol{\mu}_1) = \mathbf{f}_{n_1}(\boldsymbol{\mu}_1), \tag{13}$$

where $\mathbb{A}_{n_1}(\boldsymbol{\mu}_1) = \mathbb{V}_1^T\mathbb{A}_{N_1}(\boldsymbol{\mu}_1)\mathbb{V}_1$ and $\mathbf{f}_{n_1}(\boldsymbol{\mu}_1) = \mathbb{V}_1^T\mathbf{f}_{N_1}(\boldsymbol{\mu}_1)$. Similarly, the approximation of the slave solution is given by

$$\mathbf{u}_2(\boldsymbol{\mu}_2) \approx \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2) + \mathbf{u}_{2,D}(\boldsymbol{\mu}_2), \tag{14}$$

where $\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)$ is the solution of the *reduced slave problem*

$$\mathbb{A}_{n_2}(\boldsymbol{\mu}_2)\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2) = \mathbf{f}_{n_2}(\boldsymbol{\mu}_2) - \mathbb{V}_2^T\mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\mathbf{u}_{2,D}(\boldsymbol{\mu}_2), \tag{15}$$

denoting by $\mathbb{A}_{n_2}(\boldsymbol{\mu}_2) = \mathbb{V}_2^T\mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\mathbb{V}_2$ and $\mathbf{f}_{n_2}(\boldsymbol{\mu}_2) = \mathbb{V}_2^T\mathbf{f}_{N_2}(\boldsymbol{\mu}_2)$. Note that even if $\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)$ has homogeneous Dirichlet boundary conditions, it depends on the interface data; thus, the interface conditions must be considered as a parameter-dependent quantity in the reduction of the slave model.

Finally, the parametric Dirichlet data exchange at the interface of the two domains has to be reduced. Assuming that there is a portion of the slave boundary $\partial\Omega_{2,D} \setminus \Gamma$ with Dirichlet boundary conditions different from the interface data, we can write $\mathbf{u}_{2,D}(\boldsymbol{\mu}_2) = \mathbf{u}_{2|\partial\Omega_{2,D}\setminus\Gamma}(\boldsymbol{\mu}_2) + \mathbf{u}_{2|\Gamma}(\boldsymbol{\mu}_2)$, so that $\mathbf{u}_{2|\partial\Omega_{2,D}\setminus\Gamma}(\boldsymbol{\mu}_2)$ is just another term of the right hand side of (15), while $\mathbf{u}_{2|\Gamma}(\boldsymbol{\mu}_2) = \mathbf{u}_{1|\Gamma}(\boldsymbol{\mu}_1)$. For simplicity, here we assume that $\mathbf{u}_{2,D}(\boldsymbol{\mu}_2)$ corresponds to the data coming from the master model, i.e., $\mathbf{u}_{2,D}(\boldsymbol{\mu}_2) = \mathbf{u}_{1|\Gamma}(\boldsymbol{\mu}_1)$.

4.2 Unsteady case

Let us now move to the unsteady case of Section 3.2. In this case, the set of snapshots is then made by FOM solutions collected at each time step, for each selected parameter value. As for the steady case, we consider the lifting technique to isolate the non-homogeneous Dirichlet interface conditions, i.e., the slave solution $\mathbf{u}_2^{n+1}(\boldsymbol{\mu}_2)$ becomes

$$\mathbf{u}_2^{n+1}(\boldsymbol{\mu}_2) = \tilde{\mathbf{u}}_2^{n+1}(\boldsymbol{\mu}_2) + \mathbf{u}_{2,D}^{n+1}(\boldsymbol{\mu}_2), \qquad n = 0, \dots, N_t - 1.$$

The slave FOM model therefore reads: find $\tilde{\mathbf{u}}_2^{n+1}(\boldsymbol{\mu}_2) \in \mathbb{R}^{N_2}$ such that

$$\begin{cases} \left(\frac{\mathbb{M}_{N_2}}{\Delta t} + \mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\right) \tilde{\mathbf{u}}_2^{n+1}(\boldsymbol{\mu}_2) = \mathbf{f}_{N_2}^{n+1}(\boldsymbol{\mu}_2) + \frac{\mathbb{M}_{N_2}}{\Delta t}\mathbf{u}_2^n(\boldsymbol{\mu}_2) \\ \qquad - \left(\frac{\mathbb{M}_{N_2}}{\Delta t} + \mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\right) \mathbf{u}_{2,D}^{(n+1)}(\boldsymbol{\mu}_2), \qquad n = 0, \dots, N_t - 1, \\ \tilde{\mathbf{u}}_2^0(\boldsymbol{\mu}_2) = \mathbf{u}_{2,0}(\boldsymbol{\mu}_2) - \mathbf{u}_{2,D}^0(\boldsymbol{\mu}_2). \end{cases}$$

Considering a POD-Galerkin approach, we define the snapshots sets as $\mathbf{S}_1 = \{\mathbf{u}_1^{t_1}(\boldsymbol{\mu}_1^k), \dots, \mathbf{u}_1^{t_{N_t}}(\boldsymbol{\mu}_1^k),\ \boldsymbol{\mu}_1^k \in \mathscr{P}^{d_1}\}$, and $\tilde{\mathbf{S}}_2 = \{\tilde{\mathbf{u}}_2^{t_1}(\boldsymbol{\mu}_2^k), \dots, \tilde{\mathbf{u}}_2^{t_{N_t}}(\boldsymbol{\mu}_2^k),\ \boldsymbol{\mu}_2^k \in \mathscr{P}^{d_2}\}$, for the master and the slave model, respectively. POD is then applied on each snapshots set, yielding also in this case a global reduced basis, encoded in the matrix $\mathbb{V}_i \in \mathbb{R}^{N_i \times n_i}$, $n_i \ll N_i$, $i = 1, 2$. The expression of the ROM using a POD-Galerkin method can be obtained pursuing a Galerkin projection onto the reduced space for both the master and the slave model.

The ROM for the master problem reads: find $\mathbf{u}_{n_1}^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{n_1}$ such that

$$\begin{cases} \left(\frac{\mathbb{M}_{n_1}}{\Delta t} + \mathbb{A}_{n_1}(\boldsymbol{\mu}_1)\right) \mathbf{u}_{n_1}^{n+1}(\boldsymbol{\mu}_1) = \mathbf{f}_{n_1}^{n+1}(\boldsymbol{\mu}_1) + \frac{\mathbb{M}_{n_1}}{\Delta t}\mathbf{u}_{n_1}^n(\boldsymbol{\mu}_1), \quad n = 0, \dots, N_t - 1, \\ \mathbf{u}_{n_1}^0(\boldsymbol{\mu}_1) = \mathbf{u}_{n_1,0}(\boldsymbol{\mu}_1), \end{cases}$$

with $\mathbb{M}_{n_1} = \mathbb{V}_1^T \mathbb{M}_{N_1} \mathbb{V}_1$, $\mathbb{A}_{n_1}(\boldsymbol{\mu}_1) = \mathbb{V}_1^T \mathbb{A}_{N_1}(\boldsymbol{\mu}_1)\mathbb{V}_1$, $\mathbf{f}_{n_1}^{n+1}(\boldsymbol{\mu}_1) = \mathbb{V}_1^T \mathbf{f}_{N_1}^{n+1}(\boldsymbol{\mu}_1)$, and $\mathbf{u}_{n_1,0}(\boldsymbol{\mu}_1)$ the projection of the initial datum for the master problem $\mathbf{u}_{N_1,0}(\boldsymbol{\mu}_1)$ on the reduced basis. Note that now the right hand side $\mathbf{f}_{N_1}^{n+1}(\boldsymbol{\mu}_1)$ is time dependent and must be reassembled at each time step. If the matrix $\mathbb{A}_{N_1}(\boldsymbol{\mu}_1)$ is time independent, then the reduced matrix $\mathbb{A}_{n_1}(\boldsymbol{\mu}_1)$ can be pre-computed and stored offline, and directly used online.

The ROM for the slave problem reads: find $\tilde{\mathbf{u}}_{n_2}^{n+1}(\boldsymbol{\mu}_2) \in \mathbb{R}^{n_2}$ such that

$$\begin{cases} \left(\frac{\mathbb{M}_{n_2}}{\Delta t} + \mathbb{A}_{n_2}(\boldsymbol{\mu}_2)\right) \tilde{\mathbf{u}}_{n_2}^{n+1}(\boldsymbol{\mu}_2) = \mathbf{f}_{n_2}^{n+1}(\boldsymbol{\mu}_2) + \frac{\mathbb{M}_{n_2}}{\Delta t}\mathbf{u}_{n_2}^n(\boldsymbol{\mu}_2) \\ \qquad - \left(\frac{\mathbb{V}_2^T \mathbb{M}_{N_2}}{\Delta t} + \mathbb{V}_2^T \mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\right) \mathbf{u}_{2,D}(\boldsymbol{\mu}_2)^{(n+1)}, \qquad n = 0, \dots, N_t - 1, \\ \tilde{\mathbf{u}}_{n_2}^0(\boldsymbol{\mu}_2) = \tilde{\mathbf{u}}_{n_2,0}(\boldsymbol{\mu}_2) - \mathbf{u}_{n_2,D}^0(\boldsymbol{\mu}_2). \end{cases}$$

Here $\mathbb{M}_{n_2} = \mathbb{V}_2^T \mathbb{M}_{N_2} \mathbb{V}_2$, $\mathbb{A}_{n_2}(\boldsymbol{\mu}_2) = \mathbb{V}_2^T \mathbb{A}_{N_2}(\boldsymbol{\mu}_2)\mathbb{V}_2$, $\mathbf{f}_{n_2}(\boldsymbol{\mu}_2) = \mathbb{V}_2^T \tilde{\mathbf{f}}_{N_2}^{n+1}(\boldsymbol{\mu}_2)$, and $\tilde{\mathbf{u}}_{n_2,0}(\boldsymbol{\mu}_2)$ and $\mathbf{u}_{n_2,D}^0(\boldsymbol{\mu}_2)$ are the projections on the slave reduced basis of the initial datum for the slave problem and the initial interface Dirichlet data, respectively. If $\mathbb{A}_{N_2}(\boldsymbol{\mu}_2)$ is time dependent, its reduced version can be stored in the offline phase, while the right hand side must be assembled at each time step.

Remark 4 The possible presence of nonlinear terms in the master or slave problems can also be treated relying on a suitable hyper-reduction technique using, for instance, the DEIM [37–41, 54]. For simplicity, in this paper, we only focus on linear problems.

5 Interface DEIM reduction

Continuity of the solution at the interface of two problems domains is essential, and usually easy to be achieved if the discretization meshes are conforming. In this case, given the possible different global numbering of the degrees of freedom of the two discretizations, a map between the interface grids numbering must be computed. Moreover, in presence of non-conforming meshes, some interpolation method must be applied. Especially when large domains and very fine discretizations are considered, in both conforming or non-conforming cases, this procedure might become very expensive.

When dealing with parametrized solution and, thus, parametrized interface conditions, we must reduce the computational costs of this information transfer, too. Using the DEIM [37–39], we aim at reducing the dimension of the data to be transferred between the two domains in the conforming case and, at the same time, to interpolate or project such data in presence of non-conforming grids. To do this, we exploit a well chosen set of basis functions, starting from a set of snapshots of the Dirichlet interface data. Since this reduction step is independent from both the master and the slave model, the resulting interface DEIM reduction can be used as an efficient interpolation or projection method across the non-conforming interface grids, both at the FOM and at the ROM level.

Denoting by $Y_{h_k} = \{\boldsymbol{\lambda} = \mathbf{v}_{|\Gamma},\ \mathbf{v} \in X_{h_k}^{q_k}\}$ the space of traces of functions on Γ, we define the operator $\Pi : Y_{h_1} \to Y_{h_2}$ to transfer the information from the master to the slave model. When Γ_1 and Γ_2 coincide, Π is the classical Lagrange interpolation operator defined by

$$\Pi\, \boldsymbol{\phi}_{h_2}(\mathbf{v}_{i_{|\Gamma_1}}) = \boldsymbol{\phi}_{h_2}(\mathbf{v}_{i_{|\Gamma_1}}), \quad i = 1, \ldots, N_{1_{|\Gamma_1}} \quad \forall \boldsymbol{\phi}_{h_2} \in Y_{h_2},$$

where $N_{1_{|\Gamma_1}}$ is the dimension of Γ_1. In the following, we will equally denote by $N_{2_{|\Gamma_2}}$ the dimension of Γ_2.

Let us recall that h_i and p_i are the mesh size and the polynomial degree, respectively, for the FE discretization of the problem set on Ω_i – see Section 3. In case of conforming interface grids, that is when $h_1 = h_2$ and $q_1 = q_2$, Π represents the map between the interface DoFs numbering; if the interface meshes are non-conforming, Π is the interpolation or projection operator from the master to the slave interface, e.g., a linear interpolation as in Section 7.

Following the standard DEIM approach [38], during the offline phase we compute the set of snapshots for the Dirichlet interface data, that is, we compute multiple instances of the parametrized interface master solution $\mathbf{u}_{1_{|\Gamma_1}}(\boldsymbol{\mu}_1)$ solving the master problem for each $\boldsymbol{\mu}_1^k \in \mathscr{P}^{d_1}$. The set of snapshots is then provided by the Dirichlet data already interpolated on the slave interface grids, that is, we consider as snapshots set

$$\mathbf{S}_D = \{\mathbf{u}_{2_{|\Gamma_2}}(\boldsymbol{\mu}_1^k), \quad \boldsymbol{\mu}_1^k \in \mathscr{P}^{d_1}\}, \tag{16}$$

where $\mathbf{u}_{2_{|\Gamma_2}}(\boldsymbol{\mu}_1^k) = \Pi(\mathbf{u}_{1_{|\Gamma_1}}(\boldsymbol{\mu}_1^k))$. According to the assumption of Section 4, i.e., that the only Dirichlet boundary is the interface between the master and the slave domain, we have $\mathbf{u}_{2,D}(\boldsymbol{\mu}_1^k) = \Pi(\mathbf{u}_{1_{|\Gamma_1}}(\boldsymbol{\mu}_1^k))$.

Then, using POD, we build the basis $\mathbf{\Phi}_D$ to define a low-dimensional representation of Dirichlet data, that is, to approximate $\mathbf{u}_{2,D}(\boldsymbol{\mu}_1^k)$ by

$$\mathbf{u}_{2,D}(\boldsymbol{\mu}_1^k) \simeq \mathbf{\Phi}_D \mathbf{u}_{2,M}(\boldsymbol{\mu}_1^k) \tag{17}$$

where $\mathbf{u}_{2,M}(\boldsymbol{\mu}_1^k)$ is a vector of coefficients of dimension $M \ll N_{2_{|\Gamma_2}}$. Moreover, using a greedy algorithm [40], we iteratively select M indices

$$\mathcal{I}_{2,D} \subset \{1, \dots, N_{2_{|\Gamma_2}}\}, \quad | \mathcal{I}_{2,D} |= M \tag{18}$$

from the basis $\mathbf{\Phi}_D$ which minimize the interpolation error over the snapshots set according to the maximum norm. Therefore, $\mathcal{I}_{2,D}$ collects those indices of the DoFs which FOM data are extracted from the slave interface at – they are usually referred to as *magic points* in ROM computations.

Then, in the online phase, given a new parameter $\boldsymbol{\mu}_1 \in \mathscr{P}^{d_1}$, the coefficient vector $\mathbf{u}_{2,M}(\boldsymbol{\mu}_1)$ can be found imposing M interpolation constraints at the M points corresponding to the selected indices, that is, by solving the linear system

$$\mathbf{\Phi}_{D_{|\mathcal{I}_{2,D}}} \mathbf{u}_{2,M}(\boldsymbol{\mu}_1) = \mathbf{u}_{2_{|\mathcal{I}_{2,D}}}(\boldsymbol{\mu}_1),$$

where $\mathbf{\Phi}_{D_{|\mathcal{I}_{2,D}}} \in \mathbb{R}^{M \times M}$ is the matrix containing the rows of $\mathbf{\Phi}_D$ corresponding to the set of indices $\mathcal{I}_{2,D}$. In practice, we can express

$$\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) \simeq \mathbf{\Phi}_D \mathbf{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}} \mathbf{u}_{2_{|\mathcal{I}_{2,D}}}(\boldsymbol{\mu}_1). \tag{19}$$

To include the interpolation or projection of the Dirichlet data, during the offline phase, we replace $\mathbf{u}_{2_{|\mathcal{I}_{2,D}}}(\boldsymbol{\mu}_1)$ with the interface solution of the master problem in the corresponding DoFs. Therefore, for each index $i_2 \in \mathcal{I}_{2,D}$, we extract the corresponding DoF $\mathbf{p}_2$ in Cartesian coordinates and look for

$$\mathbf{p}_1 = \min_{\mathbf{p}_1^j \in DoFs_{\Gamma_1}} (\text{dist}(\mathbf{p}_2 - \mathbf{p}_1^j)),$$

meaning the nearest DoF to $\mathbf{p}_2$ in the master interface. Then, we search the index of $\mathbf{p}_1$ (with respect to the master numeration) to construct a set of indices

$$\mathcal{I}_{1,D} = \{i_1^{i_2}\}_{i_2 \in \mathcal{I}_{2,D}}.$$

Hence, during the online phase, the values needed to reconstruct the Dirichlet data directly on the slave interface are given by the values of the master solution $\mathbf{u}_1$ extracted in correspondence to the i_1-th DoF according to $\mathcal{I}_{1,D}$, i.e., in the i_1th magic point, meaning that $\mathbf{u}_{2_{|\mathcal{I}_{2,D}}}(\boldsymbol{\mu}_1) = \mathbf{u}_{1_{|\mathcal{I}_{1,D}}}(\boldsymbol{\mu}_1)$ and

$$\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) \simeq \mathbf{\Phi}_D \mathbf{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}} \mathbf{u}_{1_{|\mathcal{I}_{1,D}}}(\boldsymbol{\mu}_1).$$

We summarize the interface DEIM reduction in Algorithm 1.

Remark 5 The indices in $\mathcal{I}_{2,D}$ are not necessarily listed in ascending order. Moreover, $\mathcal{I}_{1,D}$ must be ordered as $\mathcal{I}_{2,D}$.

Remark 6 When $\Gamma_1 = \Gamma_2 = \Gamma$ and the meshes are conforming, we can find a perfect match between corresponding interface DoFs, that is, we only reduce the dimension of the data to be transferred between the two problems. Instead, if the meshes are non-conforming or $\Gamma_1 \neq \Gamma_2$, the search of the corresponding DoFs yields an error, especially when the two meshes are very coarse. Obviously, if the master discretization is much finer than the slave one, the possibility to find a good match between the DoFs in the two interfaces increases. Therefore, to minimize such error, we suggest to choose a finer mesh in the master domain than the one in the slave domain. However, since the approximation errors obtained with the ROM schemes here considered are strongly dependent on the problems at hand, a systematic investigation of the best choice of (and the relation between) the master and slave interface discretization cannot be provided in general.

Remark 7 Differently from the POD used for the master and slave reduction, the DEIM-based interpolation presented above does not depend on time. Then, in the unsteady case, it is sufficient to include in the set of snapshots (16) the time-dependent interface data to be able to apply the presented method at the domains interface (see Section 7).

During the DEIM reduction of the interface data, we have considered all vectors related to $\mathbf{u}_1(\boldsymbol{\mu}_1)$ and $\mathbf{u}_2(\boldsymbol{\mu}_2)$ in FOM form. Therefore, when also the master and the slave models are reduced, during the online computation $\mathbf{u}_1(\boldsymbol{\mu}_1)$ and $\mathbf{u}_2(\boldsymbol{\mu}_2)$ must be reconstructed from the corresponding reduced vectors $\mathbf{u}_{n_1}(\boldsymbol{\mu}_1)$ and $\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)$. This is an expensive procedure especially when the problem is time dependent and the FOM solutions must be reconstructed at each time step. To reduce the computational costs, we compute the FOM solution only for the selected magic points, introducing an extraction matrix $\mathbb{U} \in \mathbb{R}^{M \times N_1}$ that can be assembled during the offline phase once the index set $\mathcal{I}_{1,D}$ has been computed. Then, each row of $\mathbb{U}$ has all entries equal to zero except for the one in the column corresponding to the index $i_1 \in \mathcal{I}_{1,D}$, which is equal to one. For example, suppose that $\mathcal{I}_{1,D}$ has only three elements, e.g., $\mathcal{I}_{1,D} = \{5, 3, N_i\}$, then $\mathbb{U} \in \mathbb{R}^{3 \times N_i}$ is such that

$$\mathbb{U} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Therefore, according to equation (12), the magic points can be directly computed as follows

$$\mathbf{u}_{1|\mathcal{I}_{1,D}}(\boldsymbol{\mu}_1) = \mathbb{U}\mathbb{V}_1\mathbf{u}_{n_1}(\boldsymbol{\mu}_1), \tag{20}$$

and the Dirichlet data can be found through

$$\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) \simeq \boldsymbol{\Phi}_D \boldsymbol{\Phi}^{-1}_{D|\mathcal{I}_{2,D}} \mathbb{U}\mathbb{V}_1\mathbf{u}_{n_1}(\boldsymbol{\mu}_1).$$

Note that all the terms except for $\mathbf{u}_{n_1}$ in the expression above are parameter-independent, and therefore the product $\boldsymbol{\Phi}_D \boldsymbol{\Phi}^{-1}_{D|\mathcal{I}_{2,D}} \mathbb{U}\mathbb{V}_1 \in \mathbb{R}^{M \times n_1}$ can be stored once and for all offline, to be used then online.

```
procedure [ROM ARRAYS] = OFFLINE(FOM arrays, $P_{1,train}$, $\epsilon_{tol_D}$)
    Dirichlet data snapshots
    for $\boldsymbol{\mu}_1 \in P_{1,train}$ do
        $\mathbf{u}_1 \leftarrow$ solve the master model in (8);
        $\mathbf{u}_{1_{|\Gamma_1}} \leftarrow$ extract interface master solution $\Gamma_1$;
        $\mathbf{u}_{2_{|\Gamma_2}} \leftarrow$ interpolate/project $\mathbf{u}_{1_{|\Gamma_1}}$ to the slave interface $\Gamma_2$;
        $\mathbf{S}_D = [\mathbf{S}_D, \mathbf{u}_{2_{|\Gamma_2}}]$;
    end for
    DEIM reduced-order arrays:
    $\boldsymbol{\Phi}_D \leftarrow \text{POD}(\mathbf{S}_D, \epsilon_{tol_D})$;   $\mathcal{I}_{2,D} \leftarrow$ DEIM-indices($\boldsymbol{\Phi}_D$);
    Master magic points:
    for $i_2 \in \mathcal{I}_{2,D}$ do
        $p_2 \leftarrow$ get Cartesian coordinates of $i_2$ DoF;
        $p_1 = \min_{p_1^j \in DoF_{\Gamma_1}} (\text{dist}(p_2 - p_1^j)) \leftarrow$ find the nearest DoF of $\Gamma_2$ to $p_2$;
        $i_1 \leftarrow$ get the master index for $p_1$;
        $\mathcal{I}_{1,D} = [\mathcal{I}_{1,D}, i_1]$;
    end for
end procedure

procedure [$\mathbf{u}_2$] = ONLINE QUERY(ROM arrays, FOM arrays,$\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$)
    $\mathbf{u}_1 \leftarrow$ solve master model in (8) with $\boldsymbol{\mu}_1$;
    $\mathbf{u}_{1|\mathcal{I}_{1,D}} \leftarrow$ extract magic points;
    $\mathbf{u}_{2_{|\Gamma_2}} = \boldsymbol{\Phi}_D \boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}} \mathbf{u}_{1|\mathcal{I}_{1,D}} \leftarrow$ DEIM approximation;
    apply $\mathbf{u}_{2_{|\Gamma_2}}$ and solve the slave model in (8) with $\boldsymbol{\mu}_2$.
end procedure
```

Algorithm 1 Interface DEIM procedure.

Similarly, it is possible to compute directly the reduced Dirichlet term $\mathbb{V}_2^T \mathbb{A}_{N_2} \mathbf{u}_{2,D}(\boldsymbol{\mu}_1)$ in equation (15) without reconstructing the FOM vector $\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) \in \mathbb{R}^{N_2}$. In fact, the lifting term is

$$\mathbb{V}_2^T \mathbb{A}_{N_2} \mathbf{u}_{2,D}(\boldsymbol{\mu}_1) = \mathbb{V}_2^T \mathbb{A}_{N_2} \boldsymbol{\Phi}_D \boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}} \mathbb{U} \mathbb{V}_1 \mathbf{u}_{n_1}(\boldsymbol{\mu}_1),$$

where also $\mathbb{V}_2$ and $\mathbb{A}_{N_2}$ are parameter independent. Hence, in the offline phase, the matrix product $\mathbb{V}_2^T \mathbb{A}_{N_2} \boldsymbol{\Phi}_D \boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}} \mathbb{U} \mathbb{V}_1 \in \mathbb{R}^{n_2 \times n_1}$ can be computed and stored. The complete reduction of the one-way coupled problem can be found in Algorithm 2. Note that, following the proposed procedure, in the online phase, only reduced dimension operations are needed.

```
procedure [ROM ARRAYS] = OFFLINE(FOM arrays, $P_{1,train}$, $P_{2,train}$, $\epsilon_{tol_1}$, $\epsilon_{tol_2}$, $\epsilon_{tol_D}$)
    Solution and Dirichlet data snapshots
    for $\boldsymbol{\mu}_1 \in P_{1,train}$ do
        for $\boldsymbol{\mu}_2 \in P_{2,train}$ do
            $\mathbf{u}_1 \leftarrow$ solve master model (8);
            $\mathbf{u}_2 \leftarrow$ solve slave model (8);
            $\mathbf{u}_{1_{|\Gamma_1}} \leftarrow$ extract interface master solution $\Gamma_1$;
            $\mathbf{u}_{2_{|\Gamma_2}} \leftarrow$ interpolate/project $\mathbf{u}_{1_{|\Gamma_1}}$ to the slave interface $\Gamma_2$;
            $\mathbf{S}_1 = [\mathbf{S}_1, \mathbf{u}_1]$; $\mathbf{S}_2 = [\mathbf{S}_2, \mathbf{u}_2]$; $\mathbf{S}_D = [\mathbf{S}_D, \mathbf{u}_{2_{|\Gamma_2}}]$;
        end for
    end for
    $\mathbb{V}_1 \leftarrow \mathrm{POD}(\mathbf{S}_1, \epsilon_{tol_1})$;
    $\tilde{\mathbf{S}}_2 \leftarrow$ zero entries on the interface rows;
    $\mathbb{V}_2 \leftarrow \mathrm{POD}(\tilde{\mathbf{S}}_2, \epsilon_{tol_2})$;
    Reduced-order matrices:
    $\{\mathbb{A}_{n_1}, \mathbf{f}_{n_1}\} \leftarrow$ projection of the full order master arrays onto $\mathbb{V}_1$;
    $\{\mathbb{A}_{n_2}, \mathbf{f}_{n_2}\} \leftarrow$ projection of the full order slave arrays onto $\mathbb{V}_2$;
    DEIM reduced-order arrays:
    $\boldsymbol{\Phi}_D \leftarrow \mathrm{POD}(\mathbf{S}_D, \epsilon_{tol}^{D})$;   $\mathcal{I}_{2,D} \leftarrow$ DEIM-indices($\boldsymbol{\Phi}_D$);
    Master magic points:
    for $i_2 \in \mathcal{I}_{2,D}$ do
        $p_2 \leftarrow$ get Cartesian coordinates of $i_2$ DoF;
        $p_1 = \min_{p_1^j \in DoF_{\Gamma_1}} (\mathrm{dist}(p_2 - p_1^j)) \leftarrow$ find the nearest DoF of $\Gamma_2$ to $p_2$;
        $i_1 \leftarrow$ get the master index for $p_1$;
        $\mathcal{I}_{1,D} = [\mathcal{I}_{1,D}, i_1]$;
    end for
    $\mathbb{U} \leftarrow$ build an extraction matrix related to the magic points in $\mathcal{I}_{1,D}$;
    save the matrix product $\mathbb{U}\mathbb{V}_1$;
    save matrices product for slave lifting term $\mathbb{V}_2^T \mathbb{A}_{N_2} \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}^{-1} \mathbb{U}\mathbb{V}_1$;
end procedure

procedure [$\mathbf{u}_2$] = ONLINE QUERY(ROM arrays, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$)
    $\mathbf{u}_{n_1} \leftarrow$ solve ROM master model with $\boldsymbol{\mu}_1$;
    $\mathbb{V}_2^T \mathbb{A}_{N_2} \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}^{-1} \mathbb{U}\mathbb{V}_1 \mathbf{u}_{n_1} \leftarrow$ assemble the lifting term;
    $\tilde{\mathbf{u}}_{n_2} \leftarrow$ solve ROM slave model with $\boldsymbol{\mu}_2$;
    $\mathbf{u}_2 = \mathbb{V}_2 \tilde{\mathbf{u}}_{n_2} + \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}^{-1} \mathbb{U}\mathbb{V}_1 \mathbf{u}_{n_1} \leftarrow$ assemble FOM slave solution.
end procedure
```

Algorithm 2 Complete ROM procedure.

6 A posteriori error estimates

In this section, we derive suitable a posteriori estimates for the norm of the errors obtained with the proposed reduced techniques in both the steady and the unsteady case. Since POD and DEIM techniques are standard, here we only consider the error between the high-fidelity slave solution and the reduced order slave solution, which is the final result of the coupled ROM, namely

$$\|\mathbf{u}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2. \tag{21}$$

To this end, we relate the slave error to the ROM errors obtained for the master solution and the interface data. Since the slave and the master solutions are constructed in two different reduced spaces, we consider only the 2-norm which can be equally applied in both cases.

Remark 8 Even if the presented steady and unsteady estimates might seem of little practical use, they can be useful to assess the importance of different error terms and then choose appropriate tolerances during each coupled ROM step. Indeed, given the modular nature of the model, controlling the accuracy of each reduction step, i.e., *(i)* of the master ROM solution, *(ii)* of the interface DEIM approximation, and *(iii)* of the slave ROM solution, ensures the final expected accuracy of the slave solution.

6.1 Steady case

Since we aim at finding an estimate of the error computed with the POD reduction and DEIM, referring to [29], we can define the discrete residual for a generic steady reduced-order problem according to equation (8) as

$$\mathbf{r}(\mathbf{u}_N(\boldsymbol{\mu})) = \mathbf{f}_N(\boldsymbol{\mu}) - \mathbb{A}_N(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(\boldsymbol{\mu}), \tag{22}$$

where N and n are the FOM and the ROM dimensions, respectively, and $\boldsymbol{\mu} \in \mathscr{P}^d$ is the parameter vector. Then, an error bound can be found by estimating the three error terms separately, according to the following proposition.

Proposition 1 *For any* $\boldsymbol{\mu}_1 \in \mathscr{P}^{d_1}$ *and* $\boldsymbol{\mu}_2 \in \mathscr{P}^{d_2}$, *let us denote by* $\|\mathbf{u}_{2,h_2}(\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2$ *the slave ROM error. Then, the following error estimates holds:*

$$\begin{aligned}\|\mathbf{u}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2 \;\le\; & \frac{1}{\sigma_{\min}(\mathbb{A}_{N_2}(\boldsymbol{\mu}_2))}\|\mathbf{r}_2(\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2))\|_2 \\ & + \|\boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}\|_2\|(\mathbb{I} - \boldsymbol{\Phi}_D\boldsymbol{\Phi}_D^{-1})\mathbf{u}_{2,D}(\boldsymbol{\mu}_1)\|_2 \\ & + \frac{C}{\sigma_{\min}(\mathbb{A}_{N_1}(\boldsymbol{\mu}_1))}\|\mathbf{r}_1(\mathbf{u}_1(\boldsymbol{\mu}_1))\|_2,\end{aligned}$$

where $\sigma_{\min}(\mathbb{A}_{N_i})$, $i = 1, 2$ *denote the smallest singular value of* $\mathbb{A}_{N_i}$, $\mathbf{r}_i$, $i = 1, 2$ *are the discrete residual* (22) *of the master model and slave model with homogeneous Dirichlet interface conditions respectively,* $\mathbb{I}$ *is the identity matrix and* $C = \|\boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\|_2$.

Proof Exploiting the lifting technique to handle the Dirichlet boundary conditions in (14), and using the triangular inequality, we can first split the error (21) as

$$\begin{aligned}\|\mathbf{u}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2 &= \|\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) + \mathbf{u}_{2,D}(\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2) - (\mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2))_{|\Gamma_2}\|_2\\ &\le \|\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)\|_2 + \|\mathbf{u}_{2,D}(\boldsymbol{\mu}_2) - (\mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2))_{|\Gamma_2}\|_2.\end{aligned}$$

Then, since $(\mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2))_{|\Gamma_2}$ denotes the computed interface Dirichlet data, according to (19), we can write

$$\begin{aligned}\|\mathbf{u}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2 &\le \|\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)\|_2 + \|\mathbf{u}_{2,D} - (\mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2))_{|\Gamma_2}\|_2\\ &= \|\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)\|_2\\ &\qquad + \|\mathbf{u}_{2,D}(\boldsymbol{\mu}) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\mathbb{V}_1\mathbf{u}_{n_1}(\boldsymbol{\mu}_1)\|_2,\end{aligned}$$

where $\boldsymbol{\Phi}_D$ is the matrix of basis functions for the interface data defined by the DEIM as in (17), and $\mathbb{U}$ is the extraction matrix of (20). Adding and subtracting the same quantity $\boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\mathbf{u}_1(\boldsymbol{\mu}_1)$ on the last term on the right-hand side of the above inequality, we finally obtained a relation between the three computed errors of the following form

$$\begin{aligned}\|\mathbf{u}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2 \le\ & \|\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)\|_2\\ & + \|\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\mathbf{u}_1(\boldsymbol{\mu}_1)\|_2\\ & + \|\boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\|_2\|\mathbf{u}_1(\boldsymbol{\mu}_1) - \mathbb{V}_1\mathbf{u}_{n_1}(\boldsymbol{\mu}_1)\|_2.\end{aligned} \tag{23}$$

Note that in the second term $\mathbf{u}_{2,D_{|\mathcal{I}_{2,D}}}(\boldsymbol{\mu}_1) = \mathbb{U}\mathbf{u}_1(\boldsymbol{\mu}_1)$, so that

$$\|\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\,\mathbf{u}_1(\boldsymbol{\mu}_1)\|_2 = \|\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbf{u}_{2,D_{|\mathcal{I}_{2,D}}}(\boldsymbol{\mu}_1)\|_2.$$

Denoting by $C = \|\boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\|_2$, we can finally bound the above quantities by

$$\begin{aligned}\|\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(\boldsymbol{\mu}_2)\|_2 &\le \tfrac{1}{\sigma_{\min}(\mathbb{A}_{N_2}(\boldsymbol{\mu}_2))}\|\mathbf{r}_2(\tilde{\mathbf{u}}_2(\boldsymbol{\mu}_2))\|_2,\\ \|\mathbf{u}_1(\boldsymbol{\mu}_1) - \mathbb{V}_1\mathbf{u}_{n_1}(\boldsymbol{\mu}_1)\|_2 &\le \tfrac{1}{\sigma_{\min}(\mathbb{A}_{N_1}(\boldsymbol{\mu}_1))}\|\mathbf{r}_1(\mathbf{u}_1(\boldsymbol{\mu}_1))\|_2\end{aligned} \tag{24}$$

and (see [29], chapters 3 and 10, for further details)

$$\|\mathbf{u}_{2,D}(\boldsymbol{\mu}_1) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\mathbf{u}_1(\boldsymbol{\mu}_1)\|_2 \le \|\boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}\|_2\|(\mathbb{I} - \boldsymbol{\Phi}_D\boldsymbol{\Phi}_D^{-1})\mathbf{u}_{2,D}(\boldsymbol{\mu}_1)\|_2. \quad \square$$

6.2 Unsteady case

To find an estimate of the reduced error (21) in the time-dependent case, following [55] and [56], we can define the generic residual

$$\mathbf{r}(t;\boldsymbol{\mu}) = \mathbb{A}_N(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu}) + \mathbf{f}_N(t;\boldsymbol{\mu}) - \mathbb{V}\frac{d}{dt}\mathbf{u}_n(t;\boldsymbol{\mu}), \qquad t \in (0,T). \tag{25}$$

Here, with a slight abuse of notation, we call $\mathbb{A}_N(\boldsymbol{\mu}) = -\mathbb{M}_N^{-1}\mathbb{A}_N(\boldsymbol{\mu})$ and $\mathbf{f}_N(t;\boldsymbol{\mu}) = \mathbb{M}_N^{-1}\mathbf{f}_N(t;\boldsymbol{\mu})$ according to the dynamical system formulation of equation (9), being $\boldsymbol{\mu} \in \mathscr{P}^d$ the parameter vector; moreover, let $\mathbb{I}$ be the identity matrix.

Proposition 2 *Let $\mathbb{A}_{N_i}(\boldsymbol{\mu}_i)$, $i = 1, 2$ be time-invariant matrices whose eigenvalues have non-negative real part for all $\boldsymbol{\mu}_1 \in \mathscr{P}^{d_1}$ and $\boldsymbol{\mu}_2 \in \mathscr{P}^{d_2}$, respectively. Then, for each $t \in (0, T)$, the following error estimate holds:*

$$\begin{aligned}
&\|\mathbf{u}_2(t;\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(t;\boldsymbol{\mu}_2)\|_2 \leq \\
&\qquad C_2(\boldsymbol{\mu}_2)\left(\|\tilde{\mathbf{u}}_2(0;\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(0;\boldsymbol{\mu}_2)\|_2 + \int_0^t \|\mathbf{r}_2(\tau;\boldsymbol{\mu}_2)\|_2 d\tau\right) \\
&\qquad + \|\boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}\|_2 \|(\mathbb{I} - \boldsymbol{\Phi}_D\boldsymbol{\Phi}_D^{-1})\mathbf{u}_{2,D}(t;\boldsymbol{\mu}_2)\|_2 \\
&\qquad + C_1(\boldsymbol{\mu}_1)C_3\left(\|\mathbf{u}_1(0;\boldsymbol{\mu}_1) - \mathbb{V}_1\mathbf{u}_{n_1}(0;\boldsymbol{\mu}_1)\|_2 + \int_0^t \|\mathbf{r}_1(\tau;\boldsymbol{\mu}_1)\|_2 d\tau\right),
\end{aligned}$$

where $C_1(\boldsymbol{\mu}_1)$ and $C_2(\boldsymbol{\mu}_2)$ are two constants such that

$$\sup_{t\in[0,T]} \|\exp(\mathbb{A}_{N_1}(\boldsymbol{\mu}_1)t)\|_2 \leq C_1(\boldsymbol{\mu}_1) \quad \textit{and} \quad \sup_{t\in[0,T]} \|\exp(\mathbb{A}_{N_2}(\boldsymbol{\mu}_2)t)\|_2 \leq C_2(\boldsymbol{\mu}_2),$$

$C_3 = \|\boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\|_2$, and $\mathbf{r}_i$, $i = 1, 2$ are the discrete residual (25) of the master and slave model with homogeneous Dirichlet boundary conditions, respectively.

Proof Fixing a time instant $t \in [0, T]$, as in proposition 1, it is possible to link the slave error (21) to the master and interface errors according to (23), that is,

$$\begin{aligned}
\|\mathbf{u}_2(t;\boldsymbol{\mu}_2) - \mathbb{V}_2\mathbf{u}_{n_2}(t;\boldsymbol{\mu}_2)\|_2 \leq\ & \|\tilde{\mathbf{u}}_2(t;\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(t;\boldsymbol{\mu}_2)\|_2 \\
& + \|\mathbf{u}_{2,D}(t;\boldsymbol{\mu}_1) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\mathbf{u}_1(t;\boldsymbol{\mu}_1)\|_2 \\
& + \|\boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\|_2\|\mathbf{u}_1(t;\boldsymbol{\mu}_1) - \mathbb{V}_1\mathbf{u}_{n_1}(t;\boldsymbol{\mu}_1)\|_2,
\end{aligned}$$

where $\boldsymbol{\Phi}_D$ is the basis functions matrix defined in (17), and $\mathbb{U}$ the extraction matrix of (20). Moreover, following proposition 4.1 of [55] (see Appendix A for the complete proof), for an unsteady (linear) problem handled through POD, it holds that, for each $\boldsymbol{\mu} \in \mathscr{P}^d$,

$$\|\mathbf{u}(t;\boldsymbol{\mu}) - \mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu})\|_2 \leq C(\boldsymbol{\mu})\left(\|\mathbf{u}(0;\boldsymbol{\mu}) - \mathbb{V}\mathbf{u}_n(0;\boldsymbol{\mu})\|_2 + \int_0^t \|\mathbf{r}(\tau;\boldsymbol{\mu})\|_2 d\tau\right),$$

where $C(\boldsymbol{\mu}) = \sup_{t\in[0,T]} \|\exp(\mathbb{A}_N(\boldsymbol{\mu})t)\|_2$ if $\mathbb{A}_N(\boldsymbol{\mu})$ is time invariant and has eigenvalues with negative real part. This means that we can write

$$\begin{aligned}
\|\tilde{\mathbf{u}}_{2,h_2}(t;\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(t;\boldsymbol{\mu}_2)\|_2 \leq\ & C_2(\boldsymbol{\mu}_2)\|\tilde{\mathbf{u}}_2(0;\boldsymbol{\mu}_2) - \mathbb{V}_2\tilde{\mathbf{u}}_{n_2}(0;\boldsymbol{\mu}_2)\|_2 \\
& + C_2(\boldsymbol{\mu}_2)\textstyle\int_0^t\|\mathbf{r}_2(\tau;\boldsymbol{\mu}_2)\|_2 d\tau
\end{aligned}$$

and

$$\begin{aligned}
\|\mathbf{u}_1(t;\boldsymbol{\mu}_1) - \mathbb{V}_1\mathbf{u}_{n_1}(t;\boldsymbol{\mu}_1)\|_2 \leq\ & C_1(\boldsymbol{\mu}_1)\|\mathbf{u}_1(0;\boldsymbol{\mu}_1) - \mathbb{V}_1\mathbf{u}_{n_1}(0;\boldsymbol{\mu}_1)\|_2 \\
& + C_1(\boldsymbol{\mu}_1)\textstyle\int_0^t\|\mathbf{r}_1(\tau;\boldsymbol{\mu}_1)\|_2 d\tau,
\end{aligned}$$

for suitable constants $C_1(\boldsymbol{\mu})$ and $C_2(\boldsymbol{\mu})$.

Furthermore, as explained in Section 5, the DEIM applied at the interface is independent from the time variable, which means that the corresponding error can be estimated as in the steady case [29] as

$$\|\mathbf{u}_{2,D}(t;\boldsymbol{\mu}_1) - \boldsymbol{\Phi}_D\boldsymbol{\Phi}^{-1}_{D_{|\mathcal{I}_{2,D}}}\mathbb{U}\mathbf{u}_1(t;\boldsymbol{\mu}_1)\|_2 \leq \|\boldsymbol{\Phi}_{D_{|\mathcal{I}_{2,D}}}\|_2\|(\mathbb{I} - \boldsymbol{\Phi}_D\boldsymbol{\Phi}_D^{-1})\mathbf{u}_{2,D}(t;\boldsymbol{\mu}_1)\|_2.$$

Hence, the proof is complete denoting $C_3 = \|\mathbf{\Phi}_D \mathbf{\Phi}_{D|\mathcal{I}_{2,D}}^{-1} \mathbb{U}\|_2$. □

7 Numerical results

In this section, we investigate the numerical performances of the proposed reduced strategies on three one-way coupled problems: a steady-steady problem, an unsteady-steady problem, and an unsteady-unsteady problem. All the simulations, both in the online and offline stages, are performed in serial on a notebook with Intel Core i7-10710U processor and 16 GB of RAM. The mathematical models and numerical methods presented in this section have been implemented in C++ and Python languages and are based on life$^{\text{x}}$ (https://lifex.gitlab.io/) [57], a new in-house high-performance C++ FE library mainly focused on cardiac applications based on deal.II FE core [58] (https://www.dealii.org).

7.1 Test case *i*: steady model - steady model

Let us first consider a steady-steady coupled problem, with a reaction-diffusion problem and a Laplacian as master and slave models, respectively, with suitable boundary conditions:

$$\begin{cases} -\nabla \cdot (\alpha \nabla u) + \beta u = f & \text{in } \Omega_1 \\ u = 0 & \text{on } \partial\Omega_{1,D} \setminus \Gamma \\ \frac{\partial u}{\partial n_1} = 0 & \text{on } \Gamma, \end{cases} \quad \text{and} \quad \begin{cases} -\Delta v = 0 & \text{in } \Omega_2 \\ \frac{\partial v}{\partial n_2} = 0 & \text{on } \partial\Omega_{2,N}, \end{cases}$$

coupled with the following Dirichlet conditions at the interface

$$v = u \quad \text{on } \Gamma.$$

We define $f(x, y, z) = \frac{\pi}{4} y x^2 \sin\left(\frac{\pi}{2} y\right) e^{z-1}$, and we vary the two parameters α and β in $[0.5, 5]$.

The models are solved in three-dimensional domains represented by two concentric hollow spheroids centered in the origin; in particular, Ω_1 is the internal spheroid with inner and outer radii equal to 0.5 m and 1.5 m, while Ω_2 is the external one, with inner and outer radii equal to 1.5 m and 3.5 m. The interface Γ between the two domains is the spherical surface corresponding to the external boundary Γ_1 of Ω_1 and the internal boundary Γ_2 of Ω_2 (see Fig. 2).

We solve this coupled problem first considering a different discretization, meaning $h_1 \neq h_2$ on Ω_1 and Ω_2 and the same FEM order, later, we will consider the same discretization on Ω_1 and Ω_2 but different FE orders, meaning $q_1 \neq q_2$.

7.1.1 Different mesh sizes

Figure 3 shows some FOM solutions obtained with FEM-$\mathbb{Q}_1$ elements for both master and slave models, but different mesh sizes, with $h_1 = 0.1079$ m and $h_2 = 0.4211$ m, so that $N_1 = 202818$ and $N_2 = 26146$ (see Fig. 2).

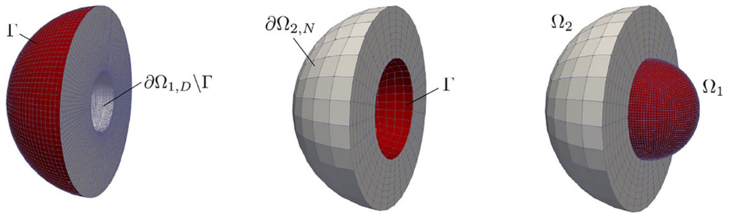

Fig. 2 *Test case i — different discretizations.* Master (left) and slave (center) domains, two hollow spheroids discretized with different meshes, i.e., $h_1 = 0.107953\ m$ and $h_2 = 0.421191\ m$. The master domain is inside the slave domain (right). In red, the interface boundary Γ

In the training phase, the high fidelity Dirichlet interface data are interpolated on the slave interface using the VTK life^{x} function that reads reference VTK data from a polygonal surface during construction, performs a linear interpolation of the point data array of the reference surface, and returns the interpolation results at specific input points. Therefore, given a parameters set, during the computation, *(i)* once solved the master problem, the solution vector must be saved in Paraview readable files, *(ii)* then, the reference surface must be extracted manually from the master solution using the ParaView software in a post-processing procedure and saved as a VTP file and, finally, *(iii)* in the slave assembling of the system, the VTK function must read the data from such file and compute the interface vector $\mathbf{u}_{2,D}$ through a linear interpolation on the slave interface DoFs. Unfortunately, this method depends on the user expertise and is quite expensive, especially in case of large dimensional interface surface and/or fine discretizations. For the test cases, in this paper, with the considered domains and discretizations, we have measured an extraction costs of about 6 min for each simulation — note that different machine characteristics and user experience might rapidly increase the total computational cost of the interface treatments.

After the interpolation method has been applied, $\mathbf{u}_{2,D}$ can be stored and used for the interface DEIM training, so that the POD-DEIM-POD ROM can be constructed

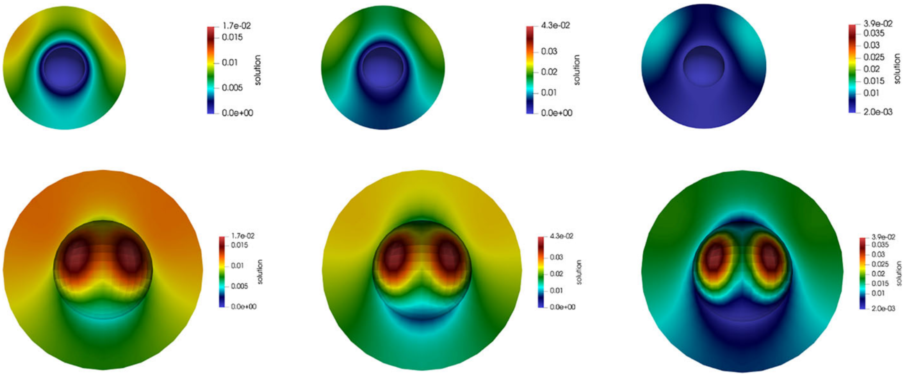

Fig. 3 *Test case i — different discretizations.* Master (top) and slave (bottom) solutions for three different instances of the parameters vector $\boldsymbol{\mu} = [\alpha, \beta]$

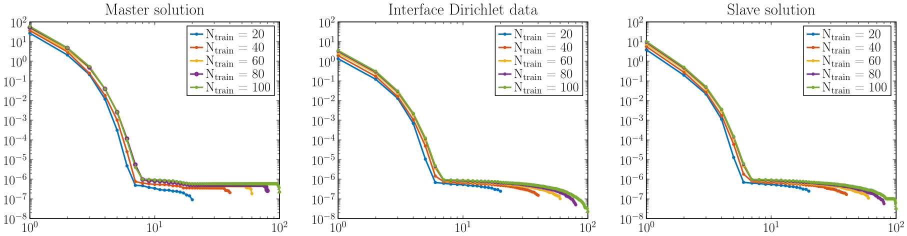

Fig. 4 *Test case i — different discretizations.* Singular values decay of the master solution (left), interface Dirichlet data (center), and slave solution (right)

according to Algorithm 2. Since we apply POD also to compute the basis functions for the interface data in the DEIM (see Section 5), we will refer to the POD tolerance not only for the master and the slave ROM, but also for the interface reduction problem.

Then, we first evaluate the singular values decay of the master solutions, the slave solutions, and the interface data, by varying the dimension N_{train} =

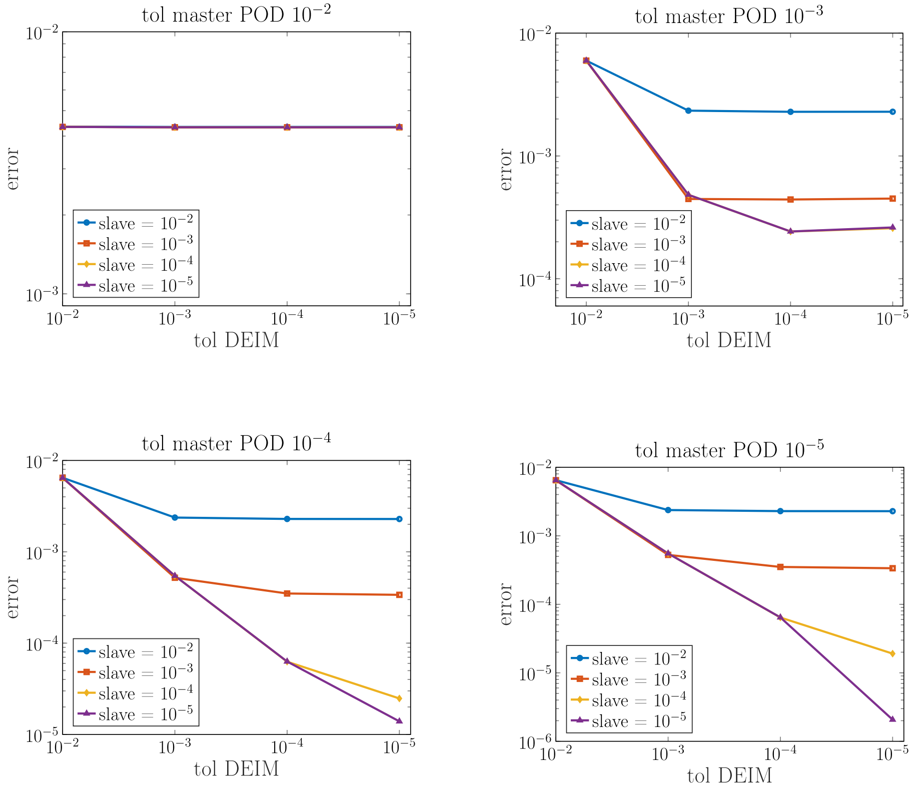

Fig. 5 *Test case i — different discretizations.* $L^2(\Omega_2)$ mean slave solution errors over N_{test} trials fixing the master POD tolerance and varying the interface and the slave POD tolerances

$\{20, 40, 60, 80, 100\}$ of the training set. The decay of the singular values, reported in Fig. 4, shows that the training set needed to get a sufficiently rich reduction has dimension at least $N_{\text{train}} = 60$. Moreover, the eigenvalues decay of the slave solution and interface data are very similar, conveying the strong dependency of the slave solution from the Dirichlet data. We then select additional $N_{\text{test}} = 50$ values of the parameters vector to test our method.

The POD technique applied to reduce the master model is standard; thus, we consider only the slave error as proof of the ability of our strategy to reconstruct the correct solution (see Section 6). In particular, we define the absolute slave error as the mean of the 2-norm error (21) over the N_{test} solutions. Figures 5 and 6 show the errors computed fixing the prescribed POD tolerance used to reduce the master model and interface data, respectively. We recall that the slave solution is dependent on the interface Dirichlet data, which in turn is influenced by the master ROM solution. Thus, the slave error depends both on the master POD and interface DEIM errors. In particular, as expected, a good approximation of the master solution but not of the interface data (and vice versa) yields high errors for the slave solution independently of the slave reduction operated. For instance, a 10^{-5} accuracy on the slave reduction

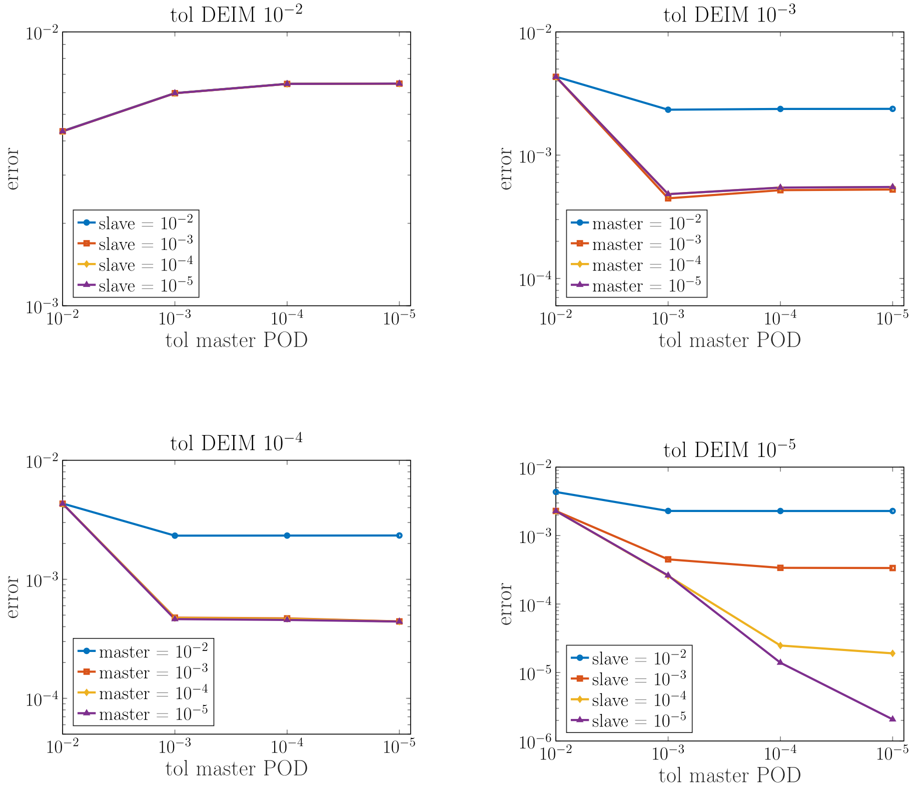

Fig. 6 *Test case i — different discretizations.* $L^2(\Omega_2)$ mean slave solution errors over N_{test} trials fixing the interface POD tolerance and varying the master and slave POD tolerances

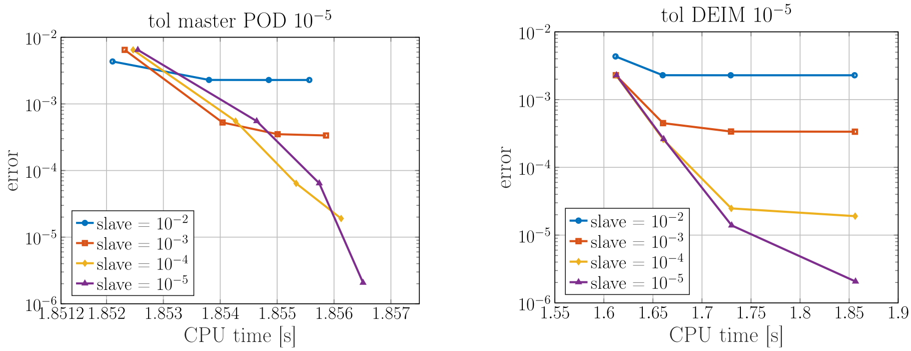

Fig. 7 *Test case i — different discretizations.* $L^2(\Omega_2)$ mean slave solution error vs the CPU time fixing the master POD tolerance (left) and the interface POD tolerance (right) to 10^{-5} and varying the tolerances used for the reduction of the other quantities

and a 10^{-2} accuracy for the master or the interface reduction yields an overall slave error of about 10^{-2}. Hence, a good approximation of all quantities is required to get a good estimate of the final solution, i.e., on average, the same order of accuracy for each step of the reduction must be imposed.

Regarding efficiency, Fig. 7 reports the error decay fixing the prescribed POD accuracy of the master and interface data reduction versus CPU time. Increasing the POD accuracy in one of the three reduction steps does not dramatically increase the total cost, especially when the prescribed tolerance for the interface reduction is fixed, entailing that the major computational cost is given by the master ROM solution. For instance, with a 10^{-5} accuracy on the master POD, passing from a 10^{-2} to a 10^{-5} accuracy on the slave and the interface POD will cause an increase of only 0.0044s (that is, an increase of about 0.23%) in the total computational costs. A 0.04% reduction in the total cost is alternatively obtained if a 10^{-4} and a 10^{-5} accuracy is imposed on the interface and the slave POD, respectively. Therefore, a very accurate reduction can be computed without loosing ROM efficiency.

Table 1 *Test case i — different discretizations*

	High fidelity model		Reduced order model			
	#FE DoFs	FE solution time	#RB	Offline time	Online time	Speed up
Master model	202k	∼ 19.76s	8	∼ 1901s	∼ 1.82s	10.9x
Slave model	26k	∼ 1.85s	6	∼ 103s	∼ 0.04s	46.3x
Interface data		∼ 6m	7	∼ 484s	0.00s	
Coupled model		∼ 381.61s		∼ 2488s	∼ 1.86s	205.2x

FOM and ROM dimensions with CPU times. We highlight the performances of the ROM with respect to the interface Dirichlet data treatment and the speed up using colors from red (worst) to green (best)

Finally, in Table 1, we report the dimensions and performances of the FOM and ROM offline and online stages for one instance of the parameter vector, with master, slave, and interface POD tolerances equal to 10^{-5}. Up to a very expensive offline phase, according to Fig. 7, a satisfying speed up of about 200 times is obtained for the slave model given that, during the online reduced computation, data reading and interpolation procedures are avoided, saving up about 98% of the computational costs of the interface extraction. However, the greatest computational cost reduction is gained by the complete coupled ROM due to the absence of the manual interface extraction method, thus implying a saving of about 100% on the interface extraction.

7.1.2 Different FE orders

We repeat the same experiment considering the same mesh size for for the discretization of the two domains and different FE orders. Specifically, we choose $q_1 = 2$ and $q_2 = 1$, and $h_1 = h_2 = 0.421191$ so that $N_1 = 202818$ and $N_2 = 26146$. Also in this case, we select $N_{\text{test}} = 50$ values of parameter vector to test the coupled ROM. In Figs. 8 and 9, we report the slave error (21) over the N_{test} trials fixing the master and

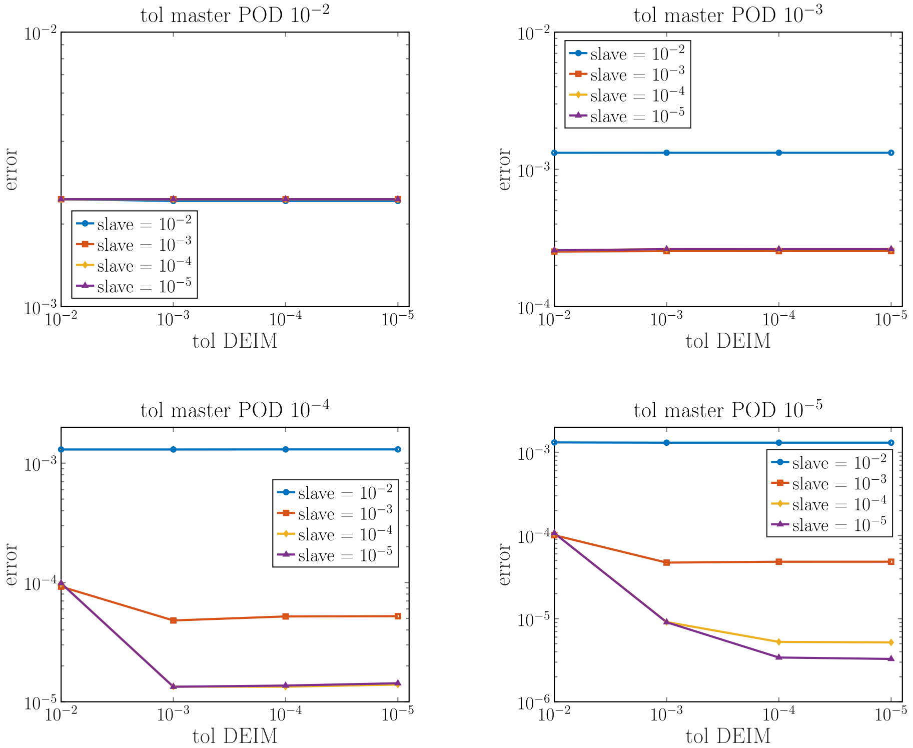

Fig. 8 *Test case i — different FE orders.* $L^2(\Omega_2)$ mean slave solution errors over N_{test} trials fixing the master POD tolerance and varying the interface and the slave POD tolerances

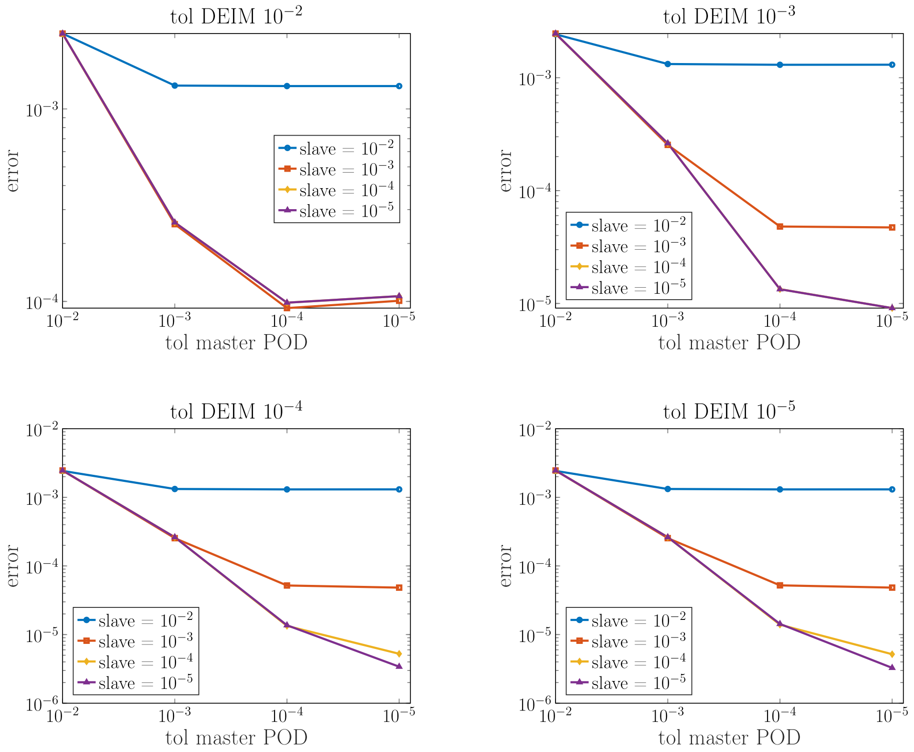

Fig. 9 *Test case i — different FE order.* $L^2(\Omega_2)$ mean slave solution errors over N_{test} trials fixing the interface POD tolerance and varying the master and the slave POD tolerances

the interface POD tolerances. Once more, a good approximation of the master solution and of the interface data provides an accurate slave solution, even if the influence

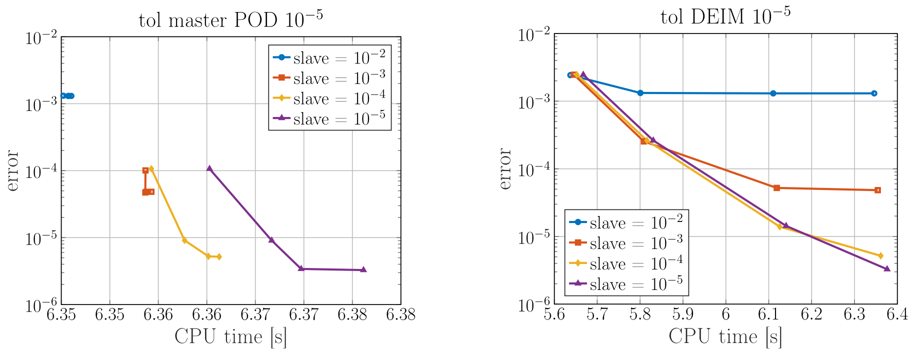

Fig. 10 *Test case i — different FE order.* $L^2(\Omega_2)$ mean slave solution errors vs. CPU time fixing the master and the interface POD tolerances to 10^{-5}, and varying the tolerances used for the reduction of the other quantities

Table 2 *Test case i — different FE order*

	High fidelity model		Reduced order model			
	#FE DoFs	FE solution time	#RB	Offline time	Online time	Speed up
Master model	202k	$\sim$ 19.49s	8	$\sim$ 1950s	$\sim$ 6.30s	3.1x
Slave model	26k	$\sim$ 2.54s	6	$\sim$ 566s	$\sim$ 0.05s	50.8x
Interface data		$\sim$ 6m	7	$\sim$ 646s	0.00s	
Coupled model		$\sim$ 382.03s		$\sim$ 3162s	$\sim$ 6.35s	59.9x

High fidelity and reduced order model dimensions and CPU times. We highlight the performances of the ROM model with respect to the interface Dirichlet data treatment and the speed up using colors from red (worst) to green (best)

of the master solution seems to be higher than before — indeed, the error decay is faster when fixing the interface POD tolerance than the corresponding master one.

Figure 10 and Table 2 show similar results regarding computational costs. The overall speed up decreases to a third of the one obtained in the first test case, mainly due to the high CPU time of the online master ROM. Such computational costs are not related to the reduced order technique applied, but to the cost of assembling the full-order stiffness matrix of the master model, which is higher when employing high order FEs compared to the $\mathbb{Q}_1$ case. Indeed, 6.25s are here needed for the assembling of $\mathbb{A}_{N_1}$, whereas the remaining reduced order operations only take 0.25s. This is, however, independent of the ROM scheme for coupled problems here presented, which results in an overall good performance, given the absence of interface conditions to handle manually.

In Fig. 11, we compare the interface results for a given instance of parameters α and β, and the location of the magic points selected through the DEIM procedure, when employing the same FE order but different mesh sizes — that is the first test case — or else $\mathbb{Q}_2 - \mathbb{Q}_1$ FE orders and the same mesh size on the master and slave problems. The picture shows that even if the interface solutions remain quite similar,

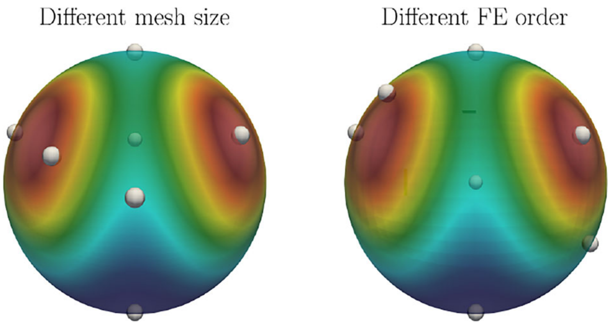

Fig. 11 Magic points location on the interface of the master and slave domains, when using different mesh size but the same FE order (left), or different FE order but the same mesh size (right). The model solution computed for $\alpha = 0.95$ and $\beta = 3.65$ is shown

the magic points — although about the same in number — can have slightly different locations. Moreover, in Fig. 12, we report the approximation errors over the interface data, fixing the tolerance for the reduction of the master model and varying that of the DEIM at the interface. The plots show that, independently of the master reduction, when dealing with a higher FE order, the DEIM is able to better approximate the interface solution, even if very similar accuracy is achieved in both cases for a prescribed DEIM tolerance of 10^{-5}. A possible explanation of these results is that the interface solutions are smoother when $\mathbb{Q}_2$ elements are employed than using $\mathbb{Q}_1$ elements, possibly enhancing the reduction process.

7.2 Test case *ii*: unsteady model - steady model

We now apply the proposed ROM to a steady-unsteady coupled problem. In particular, we choose the heat equation as master model and a simple Laplacian as slave model. Hence,

$$\begin{cases} \frac{\partial u}{\partial t} - \nabla \cdot (\alpha \nabla u) = f & \text{in } \Omega_1 \times (0, T) \\ u = 0 & \text{on } \partial\Omega_{1,D} \setminus \Gamma \times (0, T) \\ \frac{\partial u}{\partial n_1} = 0 & \text{on } \Omega_{1,N} \times (0, T) \\ \frac{\partial u}{\partial n_1} = 0 & \text{on } \Gamma \times (0, T) \\ u(0) = 0 & \text{on } \Omega_1, \end{cases} \quad \text{and} \quad \begin{cases} -\Delta v = 0 & \text{in } \Omega_2 \\ \frac{\partial v}{\partial n_2} = 0 & \text{on } \partial\Omega_{2,N}, \end{cases}$$

with the usual coupling conditions at the interface

$$u = v \quad \text{on } \Gamma.$$

We define $f(x, y, z, t) = 1 - \sin(\pi y)\cos(\frac{\pi}{2}x)$, the time interval $(0, 1)$ and we choose to vary α in $[0, 5]$. We remark that the time variable is considered as a second parameter for the slave model. The two problems are set on two three-dimensional cubes with a common face Γ. We choose $\partial\Omega_{1,D} \setminus \Gamma$ as the face of Ω_1 opposite to Γ, $\partial\Omega_{1,N}$ as the faces of Ω_1 perpendicular to Γ, and $\partial\Omega_{2,N} = \partial\Omega_2 \setminus \Gamma$. We used different

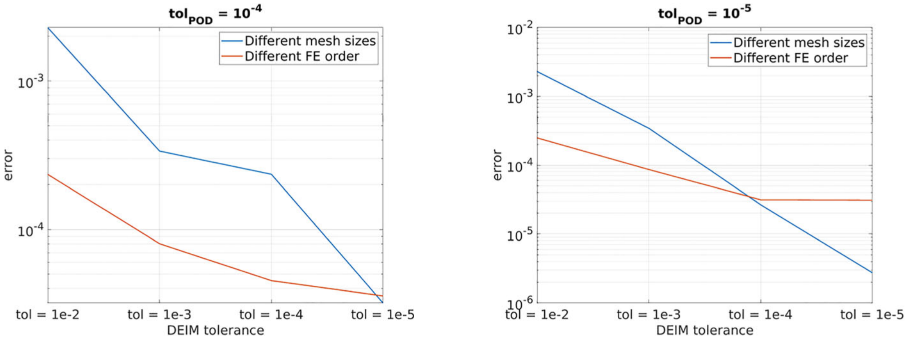

Fig. 12 $L^2(\Gamma)$ means interface errors over N_{test} trials fixing the POD tolerance for the master problem to 10^{-4} (left) and 10^{-5} (right), and varying the DEIM tolerance for the interface reduction. The plots compare the interface errors decay when employing different mesh size (blue line) or different FE order (red line) between the master and slave domain

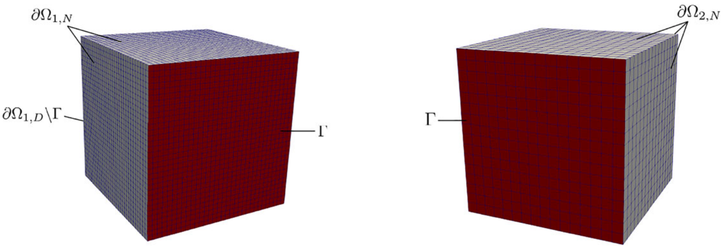

Fig. 13 *Test case ii.* Master (left) and slave (right) domains, with the interface boundary Γ reported in red

mesh sizes on Ω_1 and Ω_2, and the same FEM-$\mathbb{Q}_1$ elements. In particular, we fix $h_1 = 0.0541$ m and $h_2 = 0.1082$ m, yielding a size of $N_1 = 35937$ and $N_2 = 4913$ for the two problems. See Fig. 13 for a graphical representation of the domains and Fig. 14 for some FOM solutions at the interface.

We solve the master model using a BDF scheme of order 1 with $\Delta t = 10^{-2}$. Then, we evaluate the singular values decay of the master and slave solutions and interface data varying $N_{\text{train}} = \{10, 20, 40, 60, 80\}$. The corresponding snapshot matrices are formed by $N_s = N_t N_{\text{train}}$ full-order vectors, in which $N_t = 100$ is the number of time-steps used to solve the heat equation. The eigenvalue decay reported in Fig. 15 shows that $N_{\text{train}} = 40$ is enough to get a sufficiently rich reduction. As before, the eigenvalue decays of slave solution and interface data are quite similar.

We select $N_{\text{test}} = 5$ values of α to test our procedure and we estimate the reduced error on the slave domain as in test case *i*, considering the mean of the 2-norm error over the $N_t N_{\text{test}}$ trial of the ROM and FOM slave solutions. We report the errors in Figs. 16 and 17. We found that, once again, the reduced slave solution depends more on the reduction of the master solution than on that of the interface data. In any case,

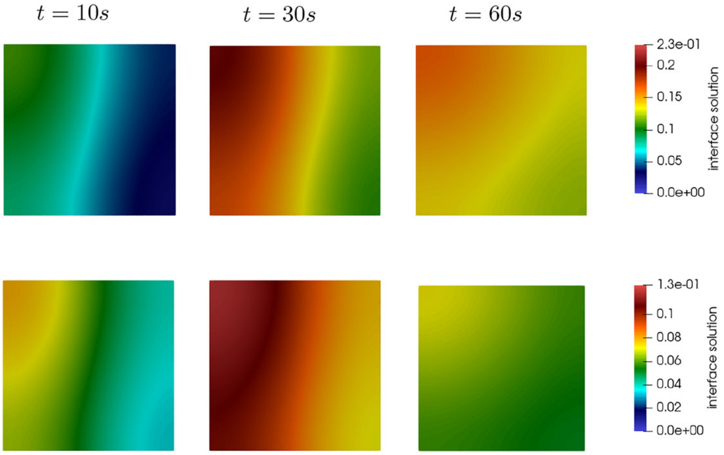

Fig. 14 *Test case ii.* Interface solutions for three different time instant (from left to right) and two different instances of α (from top to bottom)

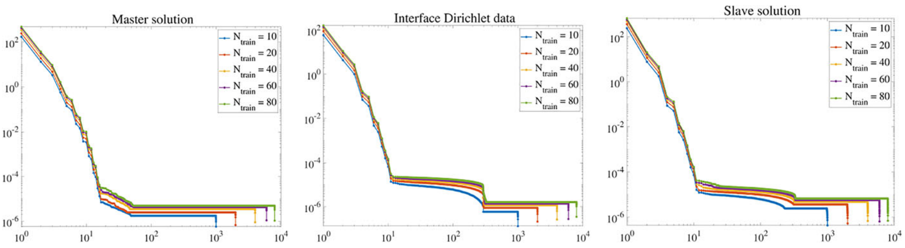

Fig. 15 *Test case ii.* Singular values decay of the master solution (left), interface Dirichlet data (center), and slave solution (right)

a good approximation of both quantities is required to obtain a good approximation of the slave solution, as for the steady-steady test case of the previous section.

Figure 18 and Table 3 outline the performances of the ROM related to a fixed prescribed POD tolerance of 10^{-5}. The reported time values refer to a complete simulation in time with 100 time steps. Compared to the steady-steady test case, the overall performances of the ROM worsens since some expensive tasks are repeated

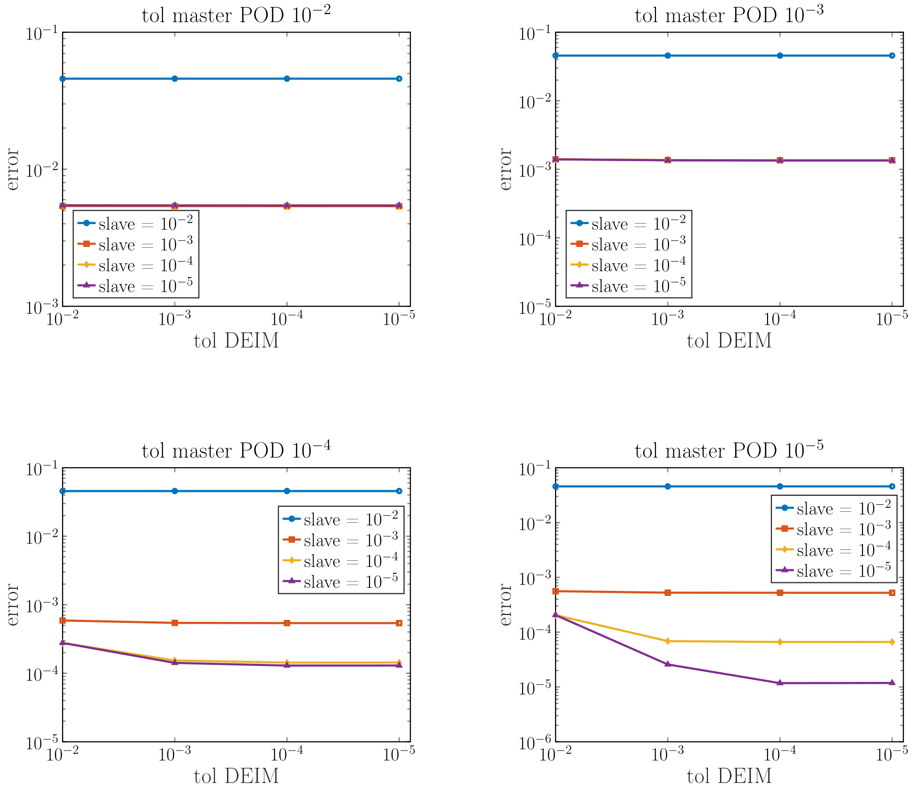

Fig. 16 *Test case ii.* $L^2(\Omega_2)$ mean slave solution errors over N_{test} trials fixing the master POD tolerance and varying the interface and the slave POD tolerances

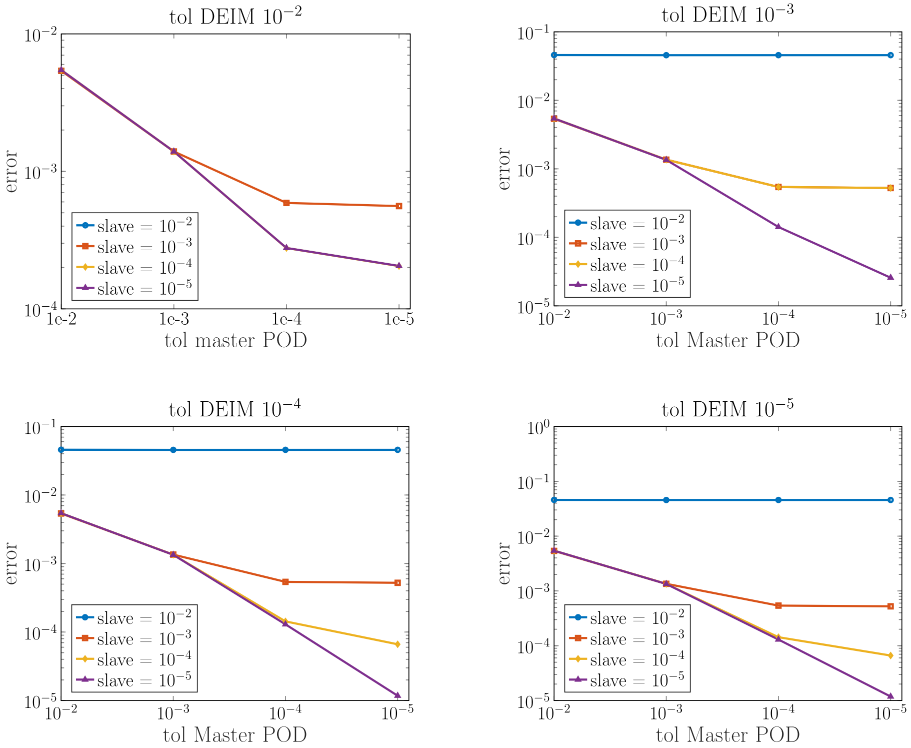

Fig. 17 *Test case ii.* $L^2(\Omega_2)$ mean slave solution errors over N_{test} trials fixing the interface POD tolerance and varying the master and the slave POD tolerances

in the ROM at each time step. In particular, according to the applied BDF formula, the right hand side of the master model depends of the FOM solution and therefore must be reconstructed at each time step. This task can be avoided considering, for example, a hyper reduction technique for the right hand side. In any case, an overall speed up of about 2 or 3 times can be obtained for each submodel, and a total speed up of 7 times can be achieved for the coupled problem, since the manual interface extraction and interpolation are not required in the online phase — two tasks that would require almost the 75% of the FOM CPU time. As before, we point out that considering an accuracy in the reduction of 10^{-5} for the three parts of the model will increase of only about 3% the total computational costs of the simulation with respect to a reduction with POD tolerances equal to 10^{-5} for the master model and to 10^{-2} for the slave and the interface subproblems.

7.3 Test case *iii*: unsteady model - unsteady model

This last test case addresses a simplified mass transfer problem used to describe the exchange of substances in biology between blood and the arterial wall. The problem unknowns are the solute concentration convected along the vessel by blood

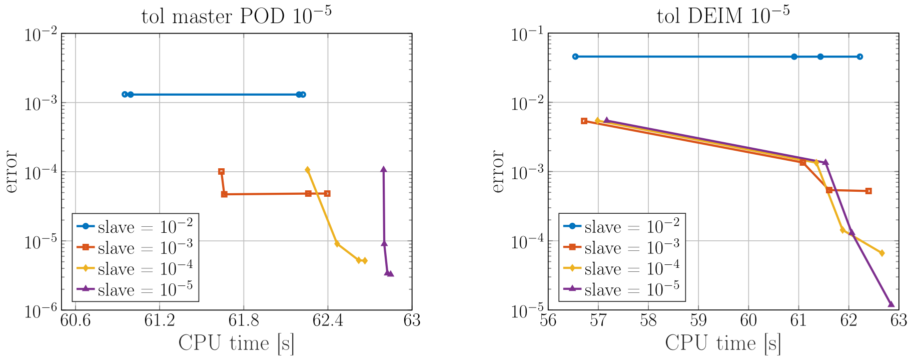

Fig. 18 *Test case ii.* $L^2(\Omega_2)$ mean slave solution errors vs. CPU time fixing the master POD tolerance (left) and the interface POD tolerance (right) to 10^{-5}, and varying the tolerances used for the reduction of the other quantities

and absorbed by the arterial wall under the blood stress induced on the vascular tissue. This fluid-wall model is based on an advection-diffusion equation to describe the solute dynamics in the arterial lumen, coupled with a pure diffusive equation accounting for the mass diffusion in the arterial wall [59, 60]. Usual coupling conditions are of Robin type; here, however, we perform a further simplification considering an isolated arterial vessel. Hence, we first solve the advection-diffusion equation for the blood transport and, then, the pure diffusive equation in the arterial wall imposing our usual interface Dirichlet conditions. Specifically, denoting $C_f(\mathbf{x}, t)$ and $C_w(\mathbf{x}, t)$ the dimensionless concentrations of the solute in the lumen Ω_f and in the wall Ω_w, respectively, we end up with the following problems:

$$\begin{cases} \frac{\partial C_f}{\partial t} + \mathbf{v} \cdot \nabla C_f - \alpha_f \Delta C_f = 0 & \text{in } \Omega_f \times (0, T) \\ C_f = \zeta & \text{on } \Sigma_{f,in} \times (0, T) \\ \alpha_f \nabla C_f \cdot \mathbf{n}_f = 0 & \text{on } \Sigma_{f,out} \cup \Gamma \times (0, T) \\ C_f(0) = 2.58 \cdot 10^{-1} & \text{in } \Omega_f. \end{cases}$$

Table 3 *Test case ii*

	High fidelity model		Reduced order model			
	#FE DoFs	FE solution time	#RB	Offline time	Online time	Speed up
Master model	36*k*	∼ 103.55s	9	∼ 4780s	∼ 55.99s	1.9*x*
Slave model	5*k*	∼ 19.66s	7	∼ 847s	∼ 6.86s	2.9*x*
Interface data		∼ 6m	15	∼ 862s	0.00s	
Coupled model		∼ 483.21s		∼ 6489s	∼ 62.84s	7.7*x*

High fidelity and reduced order model dimensions and CPU times. We highlight the performances of the ROM model with respect to the interface Dirichlet data treatment and the speed up using colors from red (worst) to green (best)

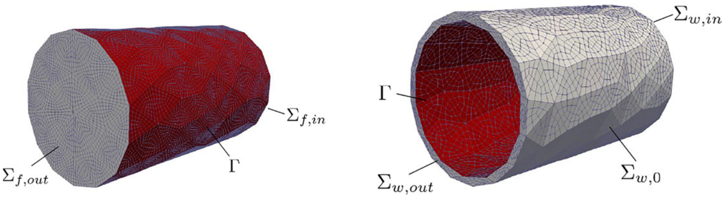

Fig. 19 *Test case iii.* Fluid (left) and wall (right) domains, a small tube of radius $r = 0.3$ cm with a corresponding wall of thickness equal to the 10% of the fluid domain lumen. Two different discretizations are considered, i.e., $h_f = 0.0863505$ cm and $h_w = 0.0914553$ m. In red, the interface boundary Γ

and

$$\begin{cases} \frac{\partial C_w}{\partial t} - \alpha_w \Delta C_w = 0 & \text{in } \Omega_w \times (0, T) \\ C_w = C_f & \text{on } \Gamma \times (0, T) \\ C_w = 0 & \text{on } \Sigma_{w,0} \times (0, T) \\ \alpha_w \nabla C_w \cdot \mathbf{n}_w = 0 & \text{on } \Sigma_{w,in/out} \times (0, T) \\ C_w(0) = 2.58 \cdot 10^{-1} & \text{in } \Omega_w. \end{cases}$$

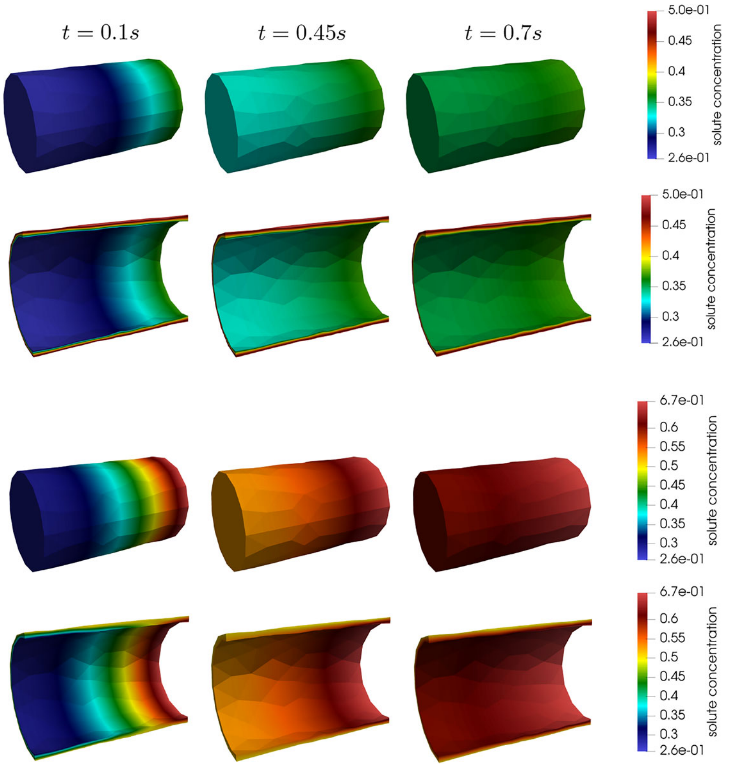

Fig. 20 *Test case iii.* Fluid solute concentration (first and third rows) and wall solution concentration considering a section of the wall domain (second and fourth rows) for three different time instants (columns), with $\zeta = 0.37$ (first two rows) and $\zeta = 0.67$ (second two rows)

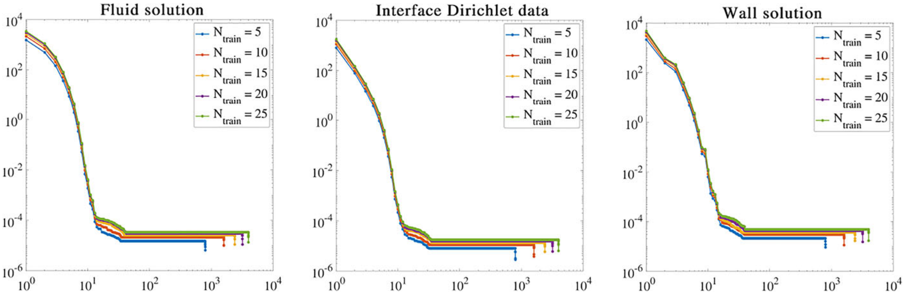

Fig. 21 *Test case iii.* Singular values decay of the fluid solution (left), interface Dirichlet data (center), and wall solution (right)

where we use subscripts f and w to refer to the fluid or the wall, respectively, in place of the usual indices 1 and 2. Here, $\mathbf{v}$ is the fluid velocity vector and α_f and α_w are the blood and wall solute diffusivity constants, respectively.

Then, we define as Ω_f a small tube of radius $r = 0.3$ cm and length 1 cm, while Ω_w has a thickness equal to 10% of the vessel lumen (see Fig. 19). We fixed the

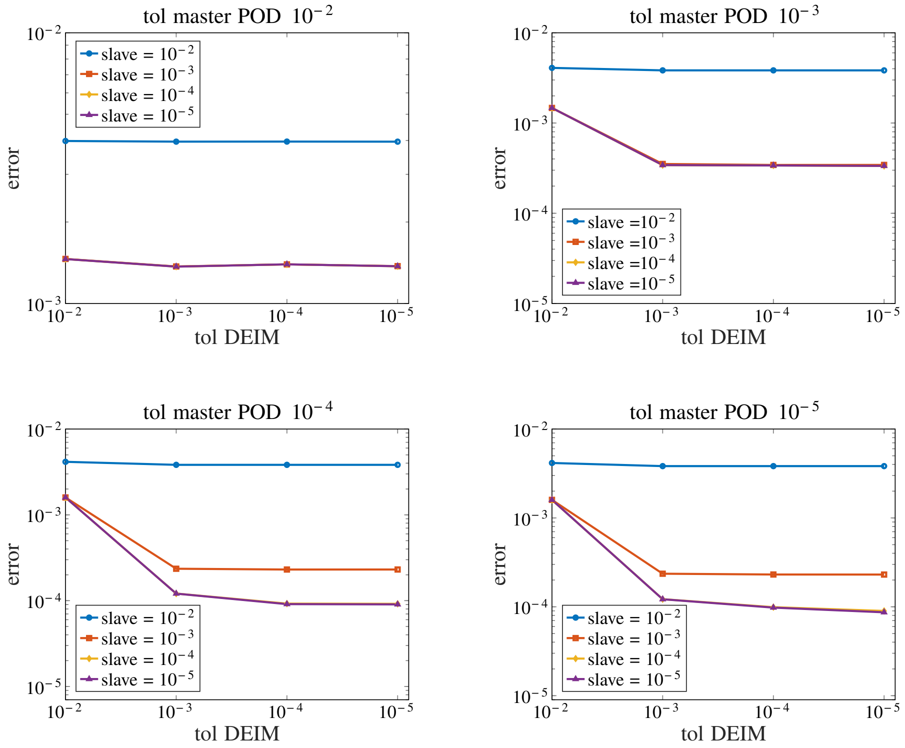

Fig. 22 *Test case iii.* $L^2(\Omega_2)$ mean wall solution error over N_{test} trials fixing the fluid POD tolerance and varying the interface and the wall POD tolerances

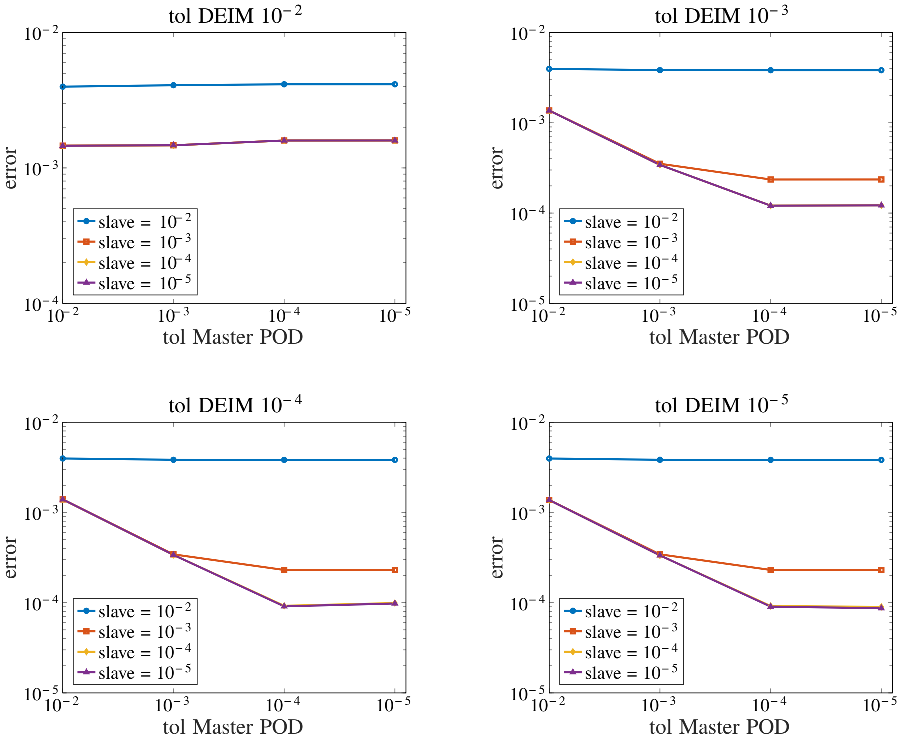

Fig. 23 *Test case iii.* $L^2(\Omega_2)$ mean wall solution errors over N_{test} trials fixing the interface POD tolerance and varying the fluid and the wall POD tolerances

initial concentration of the solute for both fluid and wall, namely $C_f(0) = C_w(0) = 2.58 \cdot 10^{-1}$, and we impose a parabolic profile to the fluid velocity with constant flow rate $Q = 2.0$ cm^3/s. Moreover, we choose $\alpha_f = 1.2 \cdot 10^{-3}$ cm^2/s and $\alpha_w = 0.9 \cdot 10^{-3}$ cm^2/s, so that the Péclet number of both problems is of order 10^3. The two physical parameters, α_f and α_w, could in principle vary. However, this will not influence the proposed techniques, but only the overall approximation due to RB methods. Therefore, for the sake of simplicity, we choose to fix the two parameters to ensure a sufficiently large Péclet number to have a predominant advection effect, which is easily captured by RB schemes.

Regarding the high-fidelity discretization, we define $T = 0.8$ s (a cardiac beat in a real biological setting), $\Delta t = 5 \cdot 10^{-3}$ s and apply a BDF scheme of order 1. We discretize differently Ω_f and Ω_w, choosing $h_f = 0.0863505$ cm and $h_w = 0.0914553$ cm so that $N_f = 144813$ and $N_w = 29624$ (see Fig. 20 for some snapshots of the fluid and wall solution). For the fluid model we consider — besides a varying time – $\zeta \in [0.1, 1]$ as parameter, which describes the solution constant concentration at the inlet; for the wall model, reduction is performed with respect to time and the interface conditions, as usual.

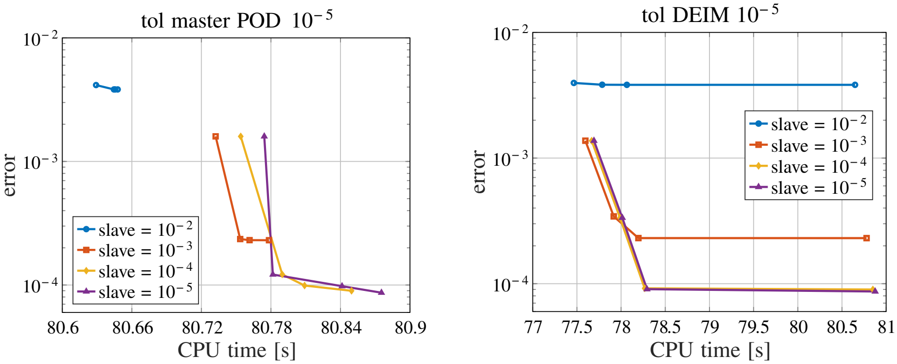

Fig. 24 *Test case iii.* $L^2(\Omega_2)$ mean wall solution error vs the CPU time fixing the fluid POD tolerance (left) and the interface POD tolerance (right) to 10^{-5} and varying the tolerances used for the reduction of the other quantities

Again, first we evaluate the singular value decay related to different snapshots sets obtained with $N_{\text{train}} = \{5, 10, 15, 20, 25\}$ samples of ζ, obtained through LHS. As for the test case *ii*, given the time-dependent nature of the coupled problem, the corresponding set of snapshots has dimension $N_s = N_t N_{\text{train}}$, where $N_t = 160$ is the number of time steps considered for each simulation. The eigenvalue decays reported in Fig. 21 show that $N_{\text{train}} = 15$ is enough to get a good reduction.

Then, we select $N_{\text{test}} = 3$ values of ζ and we estimate the reduced error on the slave domain, opting in this case for a relative 2-norm error

$$\frac{\|\mathbf{u}_{2,h_2}(\boldsymbol{\mu}_2) - \mathbb{V}_2 \mathbf{u}_{n_2}(\boldsymbol{\mu}_2)\|_2}{\|\mathbf{u}_{2,h_2}(\boldsymbol{\mu}_2)\|_2}$$

computed for each reduced solution vector. Figures 22 and 23 report the reduced errors obtained prescribing a fixed accuracy of the fluid solution and the interface data reduction, respectively. Despite the higher complexity of this test case, we highlight that the interface reduction does not impact on the final solution in terms of

Table 4 *Test case iii*

	High fidelity model		Reduced order model			
	#FE DoFs	FE solution time	#RB	Offline time	Online time	Speed up
Fluid model	144k	$\sim$ 394.68s	7	$\sim$ 5974s	$\sim$ 68.62s	5.7x
Wall model	29k	$\sim$ 250.49s	9	$\sim$ 3884s	$\sim$ 12.26s	20.4x
Interface data		$\sim$ 6m	7	$\sim$ 5525s	0.00s	
Coupled model		$\sim$ 945.17s		$\sim$ 15383s	$\sim$ 80.88s	11.7x

High fidelity and reduced order model dimensions and CPU times. We highlight the performances of the ROM model with respect to the interface Dirichlet data treatment and the speed up using colors from red (worst) to green (best) accuracy, as shown in test cases *i* and *ii*. In particular, prescribing a POD tolerance for the interface reduction between 10^{-3} and 10^{-5} does not change the final error on the slave solution; this latter is instead influenced by the slave reduced solution, as expected.

Time performances for a 10^{-5} accuracy on the master, slave, and interface POD aree instead reported in Fig. 24 and Table 4. As for test case *ii*, the computational costs refer to a complete simulation in time — in this case, a 160 time-step solution — including necessary repeated operations such as the assembling of the reduced right hand side and of the FOM solutions for both fluid and wall models. All these bottlenecks can be eventually overcome considering, especially for the master reduction, a different assembling of the right hand side and hyper-reduction techniques according to their complexity. Moreover, we point out that the gained speed up is of about 20 times, corresponding to a saving up of about 95% of the computational costs of the wall simulations due to both the ROM strategy implemented and, more importantly, of the interface non conformity considered. Together with a saving up of 100% of the interface extraction, this ensures a 12 times speed up of the complete coupled problem solution, corresponding to a reduction of the 90% of the CPU time compared to the FOM solution. Finally, we highlight that an accuracy of 10^{-5} in the slave solution requires only an increase of about 0.3% of the total cost of the solution of the same models imposing an accuracy of 10^{-5} on the master reduction, and of 10^{-2} on the slave and interface reduction, i.e., passing from a final accuracy of 10^{-2} to 10^{-5} of the slave solution.

8 Conclusions

In this paper, we have proposed a new RB method to deal with parametrized one-way coupled PDEs. This strategy can be used in combination with domain decomposition techniques when one-way coupled problems must be solved independently from each other and in sequence, following the Dirichlet interface condition direction. The efficiency of the coupled ROM is ensured by the modular nature of the proposed strategy, enabling the possibility to treat in very different ways the master and slave reduction, including different FE degrees. In particular, the main building blocks of this method are the slave and master models, to be reduced with tailored RB strategies, and the interface Dirichlet data, which is treated (and passed between the interface domains) through DEIM, without applying other expensive techniques, such as, e.g., Lagrange multipliers. Special emphasis has been put in the importance of using DEIM to handle interface data between conforming and, more importantly, non-conforming interface grids.

A posteriori reduced error estimates for the proposed method in both the steady and the unsteady cases have been derived, showing the strong relation between the slave error and the master and interface errors.

A series of numerical tests have then shown that our reduction strategy can be applied to a wealth of different coupled problems. The efficiency in the coupled ROM online phase outperforms the high-fidelity counterpart, gaining an overall speed up in the complete coupled problem computation from 200 times for the most simple steady case to 12 times for the more complex ones. However, the biggest advantage in CPU time can be seen in the interface treatment, both in the saving up of the 100% of the interface extraction time and in the general speed up obtained through the slave model reduction, which can ultimately ensure a saving up of the 95% of CPU time also in the most complex cases, such as in the fluid-wall mass transport problem of Section 7.3. Such results are related to the FOM technique employed in this work to treat interface non-conformity. This interpolation method is, indeed, specific of our computational environment and represents a bottleneck of our simulations. Other methods and implementations could be alternatively exploited, featuring possibly different computational costs. Nonetheless, the error can be carefully controlled at each step of the reduction with small influence, as a whole, of the accuracy imposed on the interface reduction.

On the basis of the results obtained with simple partitioned one-way coupled problems, we expect to be able to apply the present strategy to more complex and relevant coupled problems. A natural extension of the presented strategy also concerns the use of other kinds of interface (e.g., Neumann-like) conditions, and the treatment of more challenging two-way coupled problems; both these aspects represent the focus of a forthcoming publication.

Appendix A: A posteriori error estimator for unsteady reduced basis models

To find an a posteriori error estimate for the ROM approximation in a time-dependent case, according to [55], we can start by considering the following parametrized linear dynamical system for a vector $\mathbf{u}(t;\boldsymbol{\mu}) \in \mathbb{R}^n$:

$$\begin{cases} \frac{d}{dt}\mathbf{u}(t;\boldsymbol{\mu}) = \mathbb{A}_N(t;\boldsymbol{\mu})\mathbf{u}(t;\boldsymbol{\mu}) + \mathbf{f}_N(t;\boldsymbol{\mu}), & t \in (0,T) \\ \mathbf{u}(0;\boldsymbol{\mu}) = \mathbf{u}_0(\boldsymbol{\mu}) \end{cases}$$

Here the matrix $\mathbb{A}_N(t;\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$ and the vector $\mathbf{f}_N(t;\boldsymbol{\mu}) \in \mathbb{R}^N$, where N denotes the dimension of the reference FOM space, are $\boldsymbol{\mu}$-dependent. Moreover, we define the projection matrix $\mathbb{V} \in \mathbb{R}^{N\times n}$ defined through RB methods, where $n \le N$ is the ROM dimension. Then, the reduced dynamical system is:

$$\begin{cases} \frac{d}{dt}\mathbf{u}_n(t;\boldsymbol{\mu}) = \mathbb{A}_n(t;\boldsymbol{\mu})\mathbf{u}_n(t;\boldsymbol{\mu}) + \mathbf{f}_n(t;\boldsymbol{\mu}), & t \in [0,T] \\ \mathbf{u}_n(0;\boldsymbol{\mu}) = \mathbf{u}_{n,0}(\boldsymbol{\mu}) \end{cases} \tag{A1}$$

where $\mathbb{A}_n(t;\boldsymbol{\mu}) = \mathbb{V}^T\mathbb{A}_N(t;\boldsymbol{\mu})\mathbb{V}$, $\mathbf{f}_n(t;\boldsymbol{\mu}) = \mathbb{V}^T\mathbf{f}_N(t;\boldsymbol{\mu})\mathbb{V}$, $\mathbf{u}_n(t;\boldsymbol{\mu})$ is the reduced approximation, i.e., $\mathbf{u}_N(t;\boldsymbol{\mu}) \approx \mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu})$, and $\mathbf{u}_{n,0}(\boldsymbol{\mu})$ is the projection of $\mathbf{u}_0(\boldsymbol{\mu})$ onto the reduced space.

Let us now denote the error and the residual as

$$\begin{aligned} &\mathbf{e}(t;\boldsymbol{\mu}) := \mathbf{u}(t;\boldsymbol{\mu}) - \mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu}), \\ &\mathbf{r}(t;\boldsymbol{\mu}) := \mathbb{A}_N(t;\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu}) + \mathbf{f}_N(t;\boldsymbol{\mu}) - \mathbb{V}\tfrac{d}{dt}\mathbf{u}_n(t;\boldsymbol{\mu}), \end{aligned}$$

respectively; given a symmetric positive definite matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$, let us denote by $\langle\cdot,\cdot\rangle_{\mathbf{G}}$ the induced inner product, and the induced norm as $\|\mathbf{u}\|_{\mathbf{G}} := \sqrt{\langle\mathbf{u},\mathbf{u}\rangle_{\mathbf{G}}}$ on $\mathbf{R}^N$.

Similarly, $\|\mathbb{A}\|_{\mathbf{G}} := \sup_{\|\mathbf{u}\|_{\mathbf{G}}} \|\mathbb{A}\mathbf{u}\|_{\mathbf{G}}$, for $\mathbb{A} \in \mathbb{R}^{N\times N}$. For example, if $\mathbf{G} = \mathbb{I}_{N\times N}$, i.e., it is the identity matrix, than we obtain the simple 2-norm used in this work. Then, the following a posteriori error estimate can be stated:

Proposition 3 (*A posteriori error estimate*) *Assuming that* $\mathbb{A}_N(t;\boldsymbol{\mu}) = \mathbb{A}_N(\boldsymbol{\mu})$ *is time-invariant and has eigenvalues with negative real part for all* $\boldsymbol{\mu} \in \mathscr{P}$, *than the solution is bounded by*

$$\sup_t \|\exp(\mathbb{A}_N(\boldsymbol{\mu})t)\|_{\mathbf{G}} \leq C_1(\boldsymbol{\mu}),$$

where $C_1(\boldsymbol{\mu})$ *is a computable constant. Then, the following error estimates holds:*

$$\|\mathbf{u}(t;\boldsymbol{\mu}) - \mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu})\|_{\mathbf{G}} \leq C_1(\boldsymbol{\mu}) \left(\|\mathbf{e}(0;\boldsymbol{\mu})\|_{\mathbf{G}} + \int_0^T \|\mathbf{r}(\tau;\boldsymbol{\mu})\|_{\mathbf{G}} d\tau \right). \tag{A2}$$

Proof From the residual definition, we obtain that

$$\mathbb{V}\frac{d}{dt}\mathbf{u}_n(t;\boldsymbol{\mu}) = \mathbb{A}_N(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu}) + \mathbf{f}_N(t;\boldsymbol{\mu}) - \mathbf{r}(t;\boldsymbol{\mu}).$$

Subtracting this equation from the original system, we get the evolution system

$$\begin{cases} \frac{d}{dt}\mathbf{e}(t;\boldsymbol{\mu}) = \mathbb{A}_N(\boldsymbol{\mu})\mathbf{e}(t;\boldsymbol{\mu}) + \mathbf{r}(t;\boldsymbol{\mu}) \\ \mathbf{e}(0;\boldsymbol{\mu}) = \mathbf{u}_0(\boldsymbol{\mu}) - \mathbb{V}\mathbf{u}_{n,0}(\boldsymbol{\mu}). \end{cases} \tag{A3}$$

for the error, that admits the explicit solution

$$\mathbf{e}(t;\boldsymbol{\mu}) = \exp(\mathbb{A}_N(\boldsymbol{\mu})t)\mathbf{e}(0;\boldsymbol{\mu}) + \int_0^T \exp(\mathbb{A}_N(\boldsymbol{\mu})(T-\tau))\mathbf{r}(\tau;\boldsymbol{\mu})d\tau.$$

The thesis follows thanks to the assumption $\|\exp(\mathbb{A}_N(\boldsymbol{\mu})s)\|_{\mathbf{G}} \leq C_1(\boldsymbol{\mu})$ for $s \in \mathbb{R}^+$. □

Error relations similar to (A2) can also be found for time dependent systems, meaning when $\mathbb{A}_N(t;\boldsymbol{\mu})$ depends on time, by a suitable modification of $C_1(\boldsymbol{\mu})$. To do this, we first point out that the error evolution system (A3) holds also for time-variants systems. Then, integrating, we get

$$\mathbf{e}(t;\boldsymbol{\mu}) = \mathbf{e}(0;\boldsymbol{\mu}) + \int_0^T \mathbb{A}_N(\tau;\boldsymbol{\mu})\mathbf{e}(\tau;\boldsymbol{\mu}) + \mathbf{r}(\tau;\boldsymbol{\mu})d\tau.$$

Denoting by $\boldsymbol{\Phi}(t) := \|\mathbf{e}(t;\boldsymbol{\mu})\|_{\mathbf{G}}$, $\boldsymbol{\alpha}(t) := \|\mathbf{e}(0;\boldsymbol{\mu})\|_{\mathbf{G}} + \int_0^T \|\mathbf{r}(\tau;\boldsymbol{\mu})\|_{\mathbf{G}} d\tau$ and $\boldsymbol{\beta}(t) := \|\mathbb{A}_N(\tau;\boldsymbol{\mu})\|_{\mathbf{G}}$, we can obtain

$$\boldsymbol{\Phi}(t) \leq \boldsymbol{\alpha}(t) + \int_0^T \boldsymbol{\beta}(\tau)\boldsymbol{\Phi}(\tau)d\tau.$$

Moreover, assuming an upper bound $\|\mathbb{A}_N(t;\boldsymbol{\mu})\|_{\mathbf{G}} \leq C_3(\boldsymbol{\mu})$ for $t \in [0,T]$, $\boldsymbol{\mu} \in \mathscr{P}$, using the Gronwall inequality, we can write

$$\boldsymbol{\Phi}(t) \leq \boldsymbol{\alpha}(t) + \int_0^T \boldsymbol{\alpha}(\tau)\boldsymbol{\beta}(\tau)\exp\left(\int_s^T \boldsymbol{\beta}(r)dr\right)d\tau \leq \boldsymbol{\alpha}(t)(1 + C_3(\boldsymbol{\mu})t\exp(C_3 t)).$$

Then, equation (A2) can be found denoting by $C_1 := 1 + C_3(\boldsymbol{\mu})T\exp(C_3 T)$.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Bazilevs, Y., Takizawa, K., Tezduyar, T.: Computational Fluid-Structure Interaction: Methods and Applications. https://doi.org/10.1002/9781118483565 (2013)
2. Discacciati, M., Quarteroni, A.: Navier-Stokes/Darcy coupling: modeling, analysis, and numerical approximation. Revista Matemática Complutense **22**. https://doi.org/10.5209/rev_REMA.2009.v22.n2.16263 (2009)
3. Korvink, J., Paul, O.: MEMS: A Practical Guide to Design, Analysis and Applications. https://doi.org/10.1007/978-3-540-33655-6 (2005)
4. Piersanti, R., Regazzoni, F., Salvador, M., Corno, A., Dede', L., Vergara, C., Quarteroni, A.: 3D-0D closed-loop model for the simulation of cardiac biventricular electromechanics. arXiv:2108.01907 (2021)
5. Quarteroni, A., Dede, L., Manzoni, A., Vergara, C.: Mathematical Modelling of the Human Cardiovascular System: Data, Numerical Approximation, Clinical Applications. https://doi.org/10.1017/9781108616096 (2019)
6. Wong, J., Göktepe, S., Kuhl, E.: Computational modeling of chemo-electro-mechanical coupling: a novel implicit monolithic finite element approach. International Journal for Numerical Methods in Biomedical Engineering **29**. https://doi.org/10.1002/cnm.2565 (2013)
7. Zhao, Y., Su, X.: Computational Fluid-Structure Interaction: Methods, Models, and Applications. Academic Press, New York (2018)
8. Bonomi, D., Manzoni, A., Quarteroni, A.: A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics. Comput. Methods Appl. Mech. Eng. **324**. https://doi.org/10.1016/j.cma.2017.06.011 (2017)
9. Forti, D., Rozza, G.: Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid-structure interaction coupling problems. International Journal of Computational Fluid Dynamics **28**(3-4), 158–169 (2014). https://doi.org/10.1080/10618562.2014.932352
10. Fresca, S., Manzoni, A., Dede, L., Quarteroni, A.: POD-enhanced deep learning-based reduced order models for the real-time simulation of cardiac electrophysiology in the left atrium. Front. Physiol. **12**. https://doi.org/10.3389/fphys.2021.679076 (2021)
11. Geneser, S., Kirby, R., MacLeod, R.: Application of stochastic finite element methods to study the sensitivity of ECG forward modeling to organ conductivity. IEEE Transaction on Biomedical Engineering **55**(1), 31–40 (2008). https://doi.org/10.1109/TBME.2007.900563
12. Pacciarini, P., Rozza, G.: Reduced basis approximation of parametrized advection-diffusion PDEs with High Péclet Number. In: Numerical Mathematics and Advanced Applications-ENUMATH 2013, pp. 419–426. Springer, Lausanne (2015)

13. Pagani, S., Manzoni, A., Quarteroni, A.: Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method. Comput. Methods Appl. Mech. Eng. **340**. https://doi.org/10.1016/j.cma.2018.06.003 (2018)
14. Swenson, D., Geneser, S., Stinstra, J., Kirby, R., MacLeod, R.: Cardiac position sensitivity study in the electrocardiographic forward problem using stochastic collocation and boundary element methods. Ann. Biomed. Eng. **39**, 2900 (2011). https://doi.org/10.1007/s10439-011-0391-5
15. Bernardi, C., Maday, Y., Rapetti, F.: Basics and some applications of the mortar element method. GAMM-Mitteilungen **28**. https://doi.org/10.1002/gamm.201490020 (2005)
16. Chan, T., Smith, B., Zou, J.: Overlapping schwarz methods on unstructured meshes usingnon-matching coarse grids. Numer. Math. **73**, 149–167 (1996). https://doi.org/10.1007/s002110050189
17. Deparis, S., Forti, D., Gervasio, P., Quarteroni, A.: INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces. Computers & Fluids **141**. https://doi.org/10.1016/j.compfluid.2016.03.033 (2016)
18. Gervasio, P., Quarteroni, A.: The INTERNODES method for non-conforming discretizations of PDEs. Communications on Applied Mathematics and Computation **1**, 361–401 (2019). https://doi.org/10.1007/s42967-019-00020-1
19. Hesch, C., Gil, A., Arranz Carreño, A., Bonet, J., Betsch, P.: A mortar approach for fluid-structure interaction problems: immersed strategies for deformable and rigid bodies. Comput. Methods Appl. Mech. Eng. **278**. https://doi.org/10.1016/j.cma.2014.06.004 (2014)
20. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (1999)
21. Ripepi, M., Verveld, M., Karcher, N., Franz, T., Abu-Zurayk, M., Görtz, S., Kier, T.: Reduced-order models for aerodynamic applications, loads and MDO. CEAS Aeronaut. J. **9**. https://doi.org/10.1007/s13272-018-0283-6 (2018)
22. Amsallem, D., Cortial, J., Farhat, C.: Toward real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information. AIAA J. **48**, 2029–2037 (2010)
23. Ballarin, F., Rozza, G., Maday, Y.: Reduced-order semi-implicit schemes for fluid-structure interaction problems. In: Benner, P., Ohlberger, M., Patera, A., Rozza, G., Urban, K. (eds.) Model Reduction of Parametrized Systems, pp. 149–167. Springer, Cham (2017)
24. Lassila, T., Quarteroni, A., Rozza, G.: A reduced basis model with parametric coupling for fluid-structure interaction problems. SIAM Journal on Scientific Computing **34**(2). https://doi.org/10.1137/110819950 (2012)
25. Lassila, T., Manzoni, A., Quarteroni, A., Rozza, G.: A reduced computational and geometrical framework for inverse problems in hemodynamics. International Journal for Numerical Methods in Biomedical Engineering **29**(7), 741–776 (2013). https://doi.org/10.1002/cnm.2559
26. Ballarin, F., Rozza, G.: POD-Galerkin monolithic reduced order models for parametrized fluid-structure interaction problems: POD-galerkin monolithic ROM for parametrized FSI problems. Int. J. Numer. Methods Fluids **82**. https://doi.org/10.1002/fld.4252 (2016)
27. Benner, P., Ohlberger, M., Cohen, A., Willcox, K.: Model Reduction and Approximation: Theory and Algorithms Society for Industrial and Applied Mathematics, Philadelphia, PA. https://doi.org/10.1137/1.9781611974829 (2017)
28. Hesthaven, J., Rozza, G., Stamm, B.: Certified Reduced Basis Methods for Parametrized Partial Differential Equations. https://doi.org/10.1007/978-3-319-22470-1 (2016)
29. Quarteroni, A., Manzoni, A., Negri, F.: Reduced Basis Methods for Partial Differential Equations. An Introduction. Springer, Cham (2016)
30. Løvgren, A.E., Maday, Y., Rønquist, E.M.: A reduced basis element method for the steady Stokes problem. ESAIM: Mathematical Modelling and Numerical Analysis **40**(3), 529–552 (2006). https://doi.org/10.1051/m2an:2006021
31. Iapichino, L., Quarteroni, A., Rozza, G.: A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. Comput. Methods Appl. Mech. Eng. **221–222**, 63–82 (2012). https://doi.org/10.1016/j.cma.2012.02.005
32. Iapichino, L., Quarteroni, A., Rozza, G.: Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries. Computers & Mathematics with Applications **71**(1), 408–430 (2016). https://doi.org/10.1016/j.camwa.2015.12.001
33. Pegolotti, L., Pfaller, M.R., Marsden, A.L., Deparis, S.: Model order reduction of flow based on a modular geometrical approximation of blood vessels. Comput. Methods Appl. Mech. Eng. **380**, 113762 (2021). https://doi.org/10.1016/j.cma.2021.113762

34. Dal Santo, N., Deparis, S., Manzoni, A., Quarteroni, A.: Multi space reduced basis preconditioners for large-scale parametrized PDEs. SIAM J. Sci. Comput. **40**(2), 954–983 (2018)
35. Eftang, J., Patera, A.: A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and a posteriori error estimation. Advanced Modeling and Simulation in Engineering Sciences **1**, 3 (2014). https://doi.org/10.1186/2213-7467-1-3
36. P, H., Bao, D., Knezevic, D.J., Patera, A.T.: A static condensation reduced basis element method : approximation and a posteriori error estimation. ESAIM: Mathematical Modelling and Numerical Analysis **47**(1), 213–251 (2013). https://doi.org/10.1051/m2an/2012022
37. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C. R. Math. Acad. Sci. Paris **339**(9), 667–672 (2004)
38. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**(5), 2737–2764 (2010). https://doi.org/10.1137/090766498
39. Grepl, M., Maday, Y., Nguyen, N., Patera, A.: Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations/ ESAIM: Mathematical Modelling and Numerical Analysis **41**. https://doi.org/10.1051/m2an:2007031 (2007)
40. Maday, Y., Nguyen, N., Patera, A., Pau, G.S.H.: A general multipurpose interpolation procedure: The magic points. Communications on Pure and Applied Analysis **8**. https://doi.org/10.3934/cpaa.2009.8.383 (2008)
41. Negri, F., Manzoni, A., Amsallem, D.: Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. J. Comput. Phys. **303**, 431–454 (2015). https://doi.org/10.1016/j.jcp.2015.09.046
42. Boulakia, M., Cazeau, S., Fernández, M., Gerbeau, J.F., Zemzemi, N.: Mathematical modeling of electrocardiograms: a numerical study. Ann. Biomed. Eng. **38**, 1071–97 (2010). https://doi.org/10.1007/s10439-009-9873-0
43. Bjørstad, P.E., Brenner, S.C., Halpern, L., Kim, H.H., Kornhuber, R., Rahman, T., Widlund, O.B.: Domain Decomposition Methods in Science and Engineering XXIV. Lecture Notes in Computational Science and Engineering. Springer, Cham (2018)
44. Bernardi, C., Maday, Y., Patera, A.T.: A new non conforming approach to domain decomposition: the mortar element method. A New Non Conforming Approach to Domain Decomposition: The Mortar Element Method 13–51 (1994)
45. Gervasio, P., Quarteroni, A.: INTERNODES for heterogeneous couplings. In: Bjørstad, P.E., Brenner, S.C., Halpern, L., Kim, H.H., Kornhuber, R., Rahman, T., Widlund, O.B. (eds.) Domain Decomposition Methods in Science and Engineering XXIV, pp. 59–71. Springer, Cham (2018)
46. Gervasio, P., Quarteroni, A.: Analysis of the INTERNODES method for non-conforming discretizations of elliptic equations. Comput. Methods Appl. Mech. Eng. **334**. https://doi.org/10.1016/j.cma.2018.02.004 (2018)
47. Quarteroni, A., Valli, A.: Numerical Approximation of Partial Differential Equations. Springer Series in Computational Mathematics. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-85268-1
48. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit (4th Ed.) (2006)
49. Deparis, S., Forti, D., Quarteroni, A.: A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids. SIAM J. Sci. Comput. **36** (2014)
50. Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. Classic in applied mathematics. SIAM, Philadelphia (1995)
51. Kreiss, O., Ortiz, O.E.: Introduction to Numerical Methods for Time Dependent Differential Equations. Wiley, Hoboken (2014)
52. Mckay, M., Beckman, R., Conover, W.: A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. Technometrics **21**, 239–245 (1979). https://doi.org/10.1080/00401706.1979.10489755
53. Iman, R., Helton, J.: An investigation of uncertainty and sensitivity analysis techniques for computer-models. Risk Anal. **8**, 71–90 (2006). https://doi.org/10.1111/j.1539-6924.1988.tb01155.x
54. Farhat, C., Grimberg, S., Manzoni, A., Quarteroni, A.: Algorithms computational bottlenecks for PROMs: precomputation and hyperreduction. In: Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W.H.A., Silveira, L.M. (eds.) Snapshot-Based Methods and Algorithms, pp. 181–244. De Gruyter, Berlin (2020)

55. Haasdonk, B., Ohlberger, M.: Efficient reduced models for parametrized dynamical systems by offline/online decomposition. Math. Comput. Model. Dyn. Syst. **17**(2), 145–161 (2009). https://doi.org/10.1080/13873954.2010.514703
56. Wirtz, D., Sorensen, D.C., Haasdonk, B.: A posteriori error estimation for DEIM reduced nonlinear dynamical systems. SIAM J. Sci. Comput. **36**. https://doi.org/10.1137/120899042 (2012)
57. Africa, P.C., Piersanti, R., Fedele, M., Dede, L., Quarteroni, A.: Lifex–heart module: a high-performance simulator for the cardiac function Package 1: Fiber generation. arXiv:2201.03303 (2022)
58. Arndt, D., Bangerth, W., Blais, B., Fehling, M., Gassmöller, R., Heister, T., Heltai, L., Köcher, U., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.P., Proell, S., Simon, K., Turcksin, B., Wells, D., Zhang, J.: The deal.II library, version 9.3. Journal of Numerical Mathematics (2021)
59. Quarteroni, A., Veneziani, A., Zunino, P.: Mathematical and numerical modeling of solute dynamics in blood flow and arterial walls. SIAM J. Numer. Anal. **39**, 1488–1511 (2002). https://doi.org/10.1137/S0036142900369714
60. Karner, G., Perktold, K., Zehentner, H.P.: Computational modeling of macromolecule transport in the arterial wall. Comput. Methods Biomech. Biomed. Engin. **4**(6), 491–504 (2001). https://doi.org/10.1080/10255840108908022

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Elena Zappon[1] · **Andrea Manzoni**[1] · **Alfio Quarteroni**[1,2]

Andrea Manzoni
andrea1.manzoni@polimi.com

Alfio Quarteroni
alfio.quarteroni@polimi.it

[1] MOX - Dipartimento di Matematica, Politecnico di Milano, P.zza Leonardo da Vinci 32, Milan, I-20133, Italy

[2] Institute of Mathematics, Ecole Polytechnique Federale de Lausanne, Station 8, Lausanne, CH-1015, Switzerland