MOX-Report No. 43/2022

A reduced order model for domain decompositions with non-conforming interfaces

Zappon E.; Manzoni A.; Gervasio P.; Quarteroni A.

A reduced order model for domain decompositions with non-conforming interfaces

Elena Zappon[a,*], Andrea Manzoni[a], Paola Gervasio[b], Alfio Quarteroni[a,c]

[a]*MOX - Dipartimento di Matematica, Politecnico di Milano, P.zza Leonardo da Vinci 32, I-20133 Milano, Italy*
[b]*DICATAM, Università degli Studi di Brescia,Via Branze, 38, 25123 Brescia, Italy*
[c]*Institute of Mathematics, Ecole Polytechnique Federale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland (Professor Emeritus)*

Abstract

In this paper we propose a reduced order modeling strategy for two-way Dirichlet-Neumann parametric coupled problems solved with domain-decomposition (DD) sub-structuring methods. We split the original coupled differential problem into two sub-problems with Dirichlet and Neumann interface conditions, respectively. After discretization by (*e.g.*) the finite element method, the full-order model (FOM) is solved by Dirichlet-Neumann iterations between the two sub-problems until interface convergence is reached. We, then, apply the reduced basis (RB) method to obtain a low-dimensional representation of the solution of each sub-problem. Furthermore, we use the discrete empirical interpolation method (DEIM) applied at the interface level to achieve a fully reduced-order representation of the DD techniques implemented. To deal with interface data when non-conforming FE interface discretizations are considered, we employ the INTERNODES method combined with the interface DEIM reduction. The reduced-order model (ROM) is then solved by sub-iterating between the two reduced-order sub-problems until convergence of the approximated high-fidelity interface solutions. The ROM scheme is numerically verified on both steady and unsteady coupled problems, in the case of non-conforming FE interfaces.

Keywords: Two-way coupled problems, Dirichlet-Neumann coupling, Reduced order modeling, Discrete empirical interpolation method, Interface non-conformity, Domain-decomposition, Reduced basis method

1. Introduction

Reduced order modeling (ROM) techniques are numerical methods able to solve differential problems several orders of magnitude faster than conventional, high-fidelity full order models (FOMs), achieving real-time decision making and operational modeling in several contexts, ranging from fluid dynamics [1, 2, 3, 4, 5, 6] to biomedicine [7, 8, 9].

In several cases, domain decomposition (DD) [10, 11] techniques are necessary to *(i)* split the domain in two or more regions in which either the same or different methods are used to approximate the solution [12, 13, 14, 15], to *(ii)* solve systems that stem from the assembly of independently generated meshes [16, 17, 18] or to *(iii)* frame coupled problems where the physical nature of the involved submodels is very different [19]. In the last two cases, interface non-conformity issues can arise and ad-hoc techniques, such as, *e.g.* the MORTAR method [20, 21, 22] or the INTERNODES method [23, 24, 25] are to be implemented in order to ensure the correct exchange of information at the interface.

Domain decomposition schemes coupled with ROM techniques, especially the reduced basis (RB) method [26, 27, 28], have been first used in [29], where ROMs are applied only on small regions of the domains, for

---

instance for PDEs with discontinuous solutions in those regions where the discontinuity occurs. Similar strategies have been implemented to tackle the numerical simulation of problems in fluid dynamics [30, 31, 32, 33, 34, 35, 36], aerospace engineering [37] and structural mechanics [38, 39], as well as for the optimization of complex systems [40, 41]. A common feature shared by several of these works is the application of a reduced order model in small parts of the domains. Interface conditions, on the other hand, have been usually treated in high-fidelity form through the application of different techniques, *e.g.* Lagrangian multipliers or Fourier basis functions, for sub-iterating schemes and/or solving the coupled model in monolithic form, naturally imposing interface constraints. In [**?**] also Dirichlet and Neumann data on conforming interfaces data are also considered in reduced form, while the Dirichlet-Neumann DD iterations structure is preserve by the reduced algorithm. However, at our knowledge, interface non-conformity has never been considered, given its intrinsic complexity.

In this work, we present a Dirichlet-Neumann DD-ROM relying on the RB method able to transfer interface data across non-conforming interfaces. In particular, we consider parametric second-order elliptic and/or parabolic problems solved with Dirichlet-Neumann sub-structuring domain decomposition algorithms. Therefore, we split the problem domain into two non-overlapping subdomains with a common interface and we define two parametrized subproblems with Dirichlet or Neumann interface conditions. RB methods are then applied at the subproblems level to approximate both the subproblems solution together with the Dirichlet or Neumann interface conditions.

To solve the coupled problem and compute the snapshots (for different parameter instances) required to train our ROM, we rely on the finite element (FE) method as high-fidelity FOM. In particular, the FOM solution is sought by sub-iterating between the two submodels solutions until convergence, being this latter reached when the difference between the solution at both sides of the interface falls under a prescribed tolerance. In presence of a non-conforming interface, FOM solutions can be sought through FE based methods such as the MORTAR method or the INTERNODES. In our code, however, we surrogate the non-conforming coupling by solving for each selected parameter instance the coupled problems twice: indeed, we define two possible FE discretizations in the complete domain (each one being obtained by extending to the other subdomain the spatial discretization set in the first one) and compute the solution of each subproblem using both discretizations, one for each simulation, featuring interface conformity. Then, we extract the solution of each of the subproblems (in the discretization originally set for the corresponding subdomain), therefore obtaining two sets of solution snapshots as well as the corresponding Dirichlet and Neumann traces at the common interface that can be seen as the original model solution when interface nonconformity is considered (see Remark 3 and Fig. 1).

The RB method is then applied to define a low-dimensional representation of the solution in each subdomain, while the discrete empirical interpolation method (DEIM) is applied to both Dirichlet and Neumann interface data in order to achieve a fully reduced-order representation of the DD method adopted. In the online phase, a Galerkin projection is used to reduce the subproblems' dimension obtaining two reduced order subproblems. The solution of the coupled problem, for any new parameter instance, is then found by iterating between the solutions of the two reduced subproblems until (a suitable norm of) the difference between the two solutions at the interface, once traced back at the high-fidelity level, falls below a prescribed tolerance. In this phase, we transfer Dirichlet and Neumann interface data by applying the INTERNODES method and by using the DEIM as well as a low-order piecewise constant interpolation. This approach, which extends the work presented in [42] is still modular, and allows us to achieve a complete reduction of the model at hand, which can be seen as a two-way coupled model, including interface non-conforming grids cases.

The organization of the paper is the following: in Section 2 we present the formulation and the high-fidelity discretization of the parametrized problem, considering both steady and unsteady cases. Section 3 is devoted to the reduced-order formulation of the two sub-problems, while Section 4 is dedicated to the interface Dirichlet and Neumann reduced formulation. In Section 5, the algorithm is numerically verified by means of two test cases dealing with second-order linear PDEs, considering both an elliptic and a parabolic problem. Section 6 then reports some final remarks and possible perspectives of this work.

2. Two-way coupled problem and its FE discretization

We introduce the parameter dependent two-way coupled problem. Such problem can arise from the application of splitting domain decomposition methods to single-physics or multi-physics models. Let us start from the single-physics case, defined on a open bounded domain $\Omega \subset \mathbb{R}^n$, $(n = 2, 3)$ with Lipschitz boundary $\partial\Omega$. $\partial\Omega_D$ and $\partial\Omega_N$ denote the Dirichlet and the Neumann boundary, such that $\overline{\partial\Omega_D} \cup \overline{\partial\Omega_N} = \partial\Omega$ and $\partial\Omega_D \cap \partial\Omega_N = 0$. Given the set of parameters $\boldsymbol{\mu} \in \mathscr{P}^d$, $d \geq 1$, we search for $u(\boldsymbol{\mu})$ in Ω such that

$$\begin{cases} \mathcal{L}(\boldsymbol{\mu})u(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) & \text{in } \Omega \\ u(\boldsymbol{\mu}) = g_D(\boldsymbol{\mu}) & \text{on } \partial\Omega_D \\ \partial_{\mathcal{L}(\boldsymbol{\mu})}u(\boldsymbol{\mu}) = g_N(\boldsymbol{\mu}) & \text{on } \partial\Omega_N, \end{cases} \tag{1}$$

where $\mathcal{L}(\boldsymbol{\mu})$ is a second-order elliptic operator, $f(\boldsymbol{\mu})$, $g_D(\boldsymbol{\mu})$ and $g_N(\boldsymbol{\mu})$ are functions defined in Ω, $\partial\Omega_D$ and $\partial\Omega_N$, respectively, and $\partial_{\mathcal{L}(\boldsymbol{\mu})}u(\boldsymbol{\mu})$ is the conormal derivative associated with the operator $\mathcal{L}(\boldsymbol{\mu})$ on $\partial\Omega$.

Now, we split the computational domain Ω into two non-overlapping subdomains Ω_1 and Ω_2 with Lipschitz boundary $\partial\Omega_i$, $i = 1, 2$ and a common interface $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$. For each $i = 1, 2$, we call $\partial\Omega_{i,D} = \partial\Omega_i \cap \partial\Omega_D$ and $\partial\Omega_{i,N} = \partial\Omega_i \cap \partial\Omega_N$.

The two-domains formulation of model (1) reads [11]:

$$\begin{cases} \mathcal{L}_i(\boldsymbol{\mu})u_i(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) & \text{in } \Omega_i, \ i = 1, 2 \\ u_1(\boldsymbol{\mu}) = u_2(\boldsymbol{\mu}) & \text{on } \Gamma \\ \partial_{\mathcal{L}_1(\boldsymbol{\mu})}u_1(\boldsymbol{\mu}) + \partial_{\mathcal{L}_2(\boldsymbol{\mu})}u_2(\boldsymbol{\mu}) = 0 & \text{on } \Gamma. \end{cases} \tag{2}$$

The *Dirichlet-Neumann iterative scheme* [11, 43] is applied to solve (2). Therefore, starting from the initial guesses $u_1^0(\boldsymbol{\mu})$ and $u_2^0(\boldsymbol{\mu})$, for each $k \geq 0$, we search for $u_1^{k+1}(\boldsymbol{\mu})$ in Ω_1 and $u_2^{k+1}(\boldsymbol{\mu})$ in Ω_2 such that

$$\begin{cases} \mathcal{L}_1(\boldsymbol{\mu})u_1^{k+1}(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) & \text{in } \Omega_1 & \text{(3a)} \\ u_1^{k+1}(\boldsymbol{\mu}) = u_2^k(\boldsymbol{\mu}) & \text{on } \Gamma & \text{(3b)} \\ u_1^{k+1}(\boldsymbol{\mu}) = g_D(\boldsymbol{\mu}) & \text{on } \partial\Omega_{1,D} & \text{(3c)} \\ \partial_{\mathcal{L}_1(\boldsymbol{\mu})}u_1^{k+1}(\boldsymbol{\mu}) = g_N(\boldsymbol{\mu}) & \text{on } \partial\Omega_{1,N} & \text{(3d)} \end{cases}$$

and

$$\begin{cases} \mathcal{L}_2(\boldsymbol{\mu})u_2^{k+1}(\boldsymbol{\mu}) = f(\boldsymbol{\mu}) & \text{in } \Omega_2 & \text{(4a)} \\ \partial_{\mathcal{L}_2(\boldsymbol{\mu})}u_2^{k+1}(\boldsymbol{\mu}) - \partial_{\mathcal{L}_1(\boldsymbol{\mu})}u_1^{k+1}(\boldsymbol{\mu}) = 0 & \text{on } \Gamma & \text{(4b)} \\ u_2^{k+1}(\boldsymbol{\mu}) = g_D(\boldsymbol{\mu}) & \text{on } \partial\Omega_{2,D} & \text{(4c)} \\ \partial_{\mathcal{L}_2(\boldsymbol{\mu})}u_2^{k+1}(\boldsymbol{\mu}) = g_N(\boldsymbol{\mu}) & \text{on } \partial\Omega_{2,N}, & \text{(4d)} \end{cases}$$

where $\partial_{\mathcal{L}_i(\boldsymbol{\mu})}\mathbf{u}(\boldsymbol{\mu})$ is the conormal derivative associated with the operator $\mathcal{L}_i(\boldsymbol{\mu})$ on $\partial\Omega_i$. Moreover, a relaxation techniques [11] is usually applied to ensure and accelerate the scheme convergence.

Then, we name the problem and the corresponding solution set in Ω_1 *slave model* and *slave solution*, and the ones set in Ω_2 *master model* and *master solution*, respectively.

For each $i = 1, 2$, we first define the local spaces

$$V_i = \{v \in H^1(\Omega_i) \ | \ v = 0 \text{ on } \partial\Omega_{i,D}\} \quad \text{and} \quad V_i^0 = \{v \in V_i \ | \ v = 0 \text{ on } \Gamma\}, \tag{5}$$

and the space of traces of the elements of V on the interface Γ, meaning

$$\Lambda = \{\lambda \in H^{1/2}(\Gamma) \ | \ \exists v \in V : v_{|\Gamma} = \lambda\}. \tag{6}$$

Moreover, we denote by $\mathcal{R}_i : \Lambda \to V_i$ s.t. $(\mathcal{R}_i\eta)_{|\Gamma} = \eta \ \forall \eta \in \Lambda$ any possibile linear and continuous lifting

3

operator from the interface Γ to Ω_i.

We consider two a-priori independent discretizations $\mathcal{T}_{h_1}$ and $\mathcal{T}_{h_2}$ on the domains Ω_1 and Ω_2 that can imply a mesh non-conformity at the interface. For instance, $\mathcal{T}_{h_i}$ can be made of simplices (triangles of tetrahedra) or quads (quadrilaterals or hexahedra), depending on the mesh size, a positive parameter $h_i > 0$. Moreover, different mesh sizes h_1 and h_2, or different polynomial degrees p_1 or p_2, can be selected. Then, we call Γ_1 and Γ_2 the internal interfaces induced by $\mathcal{T}_{h_1}$ and $\mathcal{T}_{h_2}$ of Ω_1 and Ω_2, respectively: we talk of geometrical conformity if $\Gamma_1 = \Gamma_2$ and of geometrical non-conformity if $\Gamma_1 \neq \Gamma_2$. Finally, we assume that for any $T_{i,m} \in \mathcal{T}_{h_i}$, $\partial T_{i,m} \cap \partial \Omega$ fully belongs to $\partial \Omega_{i,D}$ or $\partial \Omega_{i,N}$, and that the interfaces Γ_i do not cut any $T_{i,m} \in \mathcal{T}_{h_i}$. According to the test cases in Section 5, hereon we consider only quads elements.

For each partition $\mathcal{T}_{h_i}$ we define the finite element approximation spaces as

$$X_{h_i}^{q_i} = \{v \in C^0(\overline{\Omega_i}) : v_{|T_{i,m}} \circ F_{i,m} \in \mathbb{Q}_{q_i}, \forall T_{i,m} \in \mathcal{T}_{h_i}\},$$

in which $F_{i,m}$ is a smooth bijection that maps the reference quad $(-1,1)^n$ into the quad $T_{i,m}$, and q_i are chosen integers. The finite dimensional spaces to define the discrete formulation of the exploited problems will be

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

while the spaces of traces on Γ are

$$Y_{h_i} = \{\lambda = v_{|\Gamma}, v \in X_{h_i}\} \quad \text{and} \quad \Lambda_{h_i} = \{\lambda = v_{|\Gamma}, v \in V_{h_i}\}. \tag{8}$$

Then, for $i = 1, 2$, we set the linear and continuous discrete lifting operator

$$\mathcal{R}_{h_i} : \Lambda_{h_i} \to V_{h_i}, \quad \text{s.t.} \quad (\mathcal{R}_{h_i} \eta_{h_i})_{|\Gamma} = \eta_{h_i}, \quad \forall \eta_{h_i} \in \Lambda_{h_i}. \tag{9}$$

In a practical implementation, $\mathcal{R}_{h_i} \eta_{h_i}$ is a finite element interpolant that imposes the same values of η_{h_i} at the FE nodes of Γ and zeros at any other FE node of $\mathcal{T}_{h_i} \backslash \Gamma_i$.

We also introduce two independent transfer operators able to exchange information between the independent grids on the interface Γ, namely

$$\Pi_{12} : Y_{h_2} \to Y_{h_1} \quad \text{and} \quad \Pi_{21} : Y_{h_1} \to Y_{h_2}.$$

In the non-conforming case, if Γ_1 and Γ_2 coincide, such operators could be the classical Lagrange interpolation operators, while Π_{jk} are the identity operators when the meshes are conforming. Instead, if the mesh are non-conforming and $\Gamma_1 \neq \Gamma_2$, Π_{12} and Π_{21} could be, *e.g.*, Rescaled Localized Radial Basis Function operators, as for the INTERNODES [23, 25, 44].

To exploit a FE-Galerkin approximation to set the high-fidelity FOM and get the algebraic formulation of problems (3) and (4), it is useful to consider local vectors and matrices. In particular, we define the following set of indices associated with the nodes $\mathbf{x}_i \in \mathcal{T}_{h_i}$:

$$\begin{aligned}
\mathcal{I}_{\overline{\Omega}_i} &= \{1, \dots, N_i\}, \qquad \mathcal{I}_i = \{j \in \mathcal{I}_{\overline{\Omega}_i} : \mathbf{x}_j \in \overline{\Omega}_i \backslash (\overline{\partial \Omega_{D,i}} \cup \mathring{\Gamma}_i)\} \\
\mathcal{I}_{\Gamma_i} &= \{j \in \mathcal{I}_{\overline{\Omega}_i} : \mathbf{x}_j \in \overline{\Gamma_i}\}, \qquad \mathcal{I}_{D_i} = \{j \in \mathcal{I}_{\overline{\Omega}_i} : \mathbf{x}_j \in \overline{\partial \Omega_{D,i}}\},
\end{aligned} \tag{10}$$

being N_i the cardinality of $\mathcal{I}_{\overline{\Omega}_i}$.

Then, for each $i = 1, 2$, we set the local stiffness matrices $\mathbb{A}_i(\boldsymbol{\mu})$ so that

$$\mathbb{A}_{i,i}(\boldsymbol{\mu}) = \mathbb{A}_i(\mathcal{I}_i, \mathcal{I}_i; \boldsymbol{\mu})$$

is the submatrix of $\mathbb{A}_i(\boldsymbol{\mu})$ of the rows and columns of $\mathbb{A}_i(\boldsymbol{\mu})$ whose indices belong to $\mathcal{I}_i$, *i.e.* either the internal nodes of Ω_i and those on $\partial \Omega_{i,N}$. Similarly, we can define $\mathbb{A}_{\Gamma_i,\Gamma_i}(\boldsymbol{\mu}) = \mathbb{A}_i(\mathcal{I}_{\Gamma_i}, \mathcal{I}_{\Gamma_i}; \boldsymbol{\mu})$, $\mathbb{A}_{i,\Gamma_i}(\boldsymbol{\mu}) = \mathbb{A}_i(\mathcal{I}_i, \mathcal{I}_{\Gamma_i}; \boldsymbol{\mu})$, $\mathbb{A}_{\Gamma_i,i}(\boldsymbol{\mu}) = \mathbb{A}_i(\mathcal{I}_{\Gamma_i}, \mathcal{I}_i; \boldsymbol{\mu})$ and $\mathbb{A}_{i,D}(\boldsymbol{\mu}) = \mathbb{A}_i(\mathcal{I}_i, \mathcal{I}_{D_i}; \boldsymbol{\mu})$.

Moreover, if $\mathbf{f}_{N_i}(\boldsymbol{\mu})$ and $\underline{\mathbf{u}}_{N_i}(\boldsymbol{\mu})$ are the right hand side vector and the vector of degrees of freedom of the approximated solution in $\overline{\Omega}_i$, respectively, we set

$$\mathbf{f}_i(\boldsymbol{\mu}) = \mathbf{f}_{N_i}(\mathcal{I}_i; \boldsymbol{\mu}), \quad \mathbf{f}_{\Gamma_i}(\boldsymbol{\mu}) = \mathbf{f}_{N_i}(\mathcal{I}_\Gamma; \boldsymbol{\mu}),$$

$$\mathbf{u}_i(\boldsymbol{\mu}) = \mathbf{u}_{N_i}(\mathcal{I}_i; \boldsymbol{\mu}), \quad \mathbf{u}_{\Gamma_i}(\boldsymbol{\mu}) = \mathbf{u}_{N_i}(\mathcal{I}_\Gamma; \boldsymbol{\mu}).$$

Then, by applying the INTERNODES method (see, *e.g.* [25]) the algebraic form of (3) reads as : for each $k \geq 0$, find $\mathbf{u}_1^{k+1}(\boldsymbol{\mu})$ solution of

$$\begin{cases} \mathbb{A}_{1,1}(\boldsymbol{\mu})\mathbf{u}_1^{k+1}(\boldsymbol{\mu}) = \mathbf{f}_1(\boldsymbol{\mu}) - \mathbb{A}_{1,D}(\boldsymbol{\mu})\mathbf{g}_{D,1}(\boldsymbol{\mu}) - \mathbb{A}_{1,\Gamma_1}(\boldsymbol{\mu})\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) \\ \mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) = \mathbb{R}_{12}\mathbf{u}_{\Gamma_2}^k(\boldsymbol{\mu}). \end{cases} \tag{11}$$

where $\mathbb{R}_{12}$ is the rectangular matrix associated with Π_{12}.

The algebraic formulation of problem (4) reads as: for each $k \geq 0$, find $\mathbf{u}_2^{k+1}(\boldsymbol{\mu})$ such that

$$\begin{cases} \mathbb{A}_{2,2}(\boldsymbol{\mu})\mathbf{u}_2^{k+1}(\boldsymbol{\mu}) = \mathbf{f}_2(\boldsymbol{\mu}) + \mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu}) - \mathbb{A}_{2,D}(\boldsymbol{\mu})\mathbf{g}_{D,2}(\boldsymbol{\mu}) \\ \mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu}) = -\mathbb{M}_{\Gamma_2}\mathbb{R}_{21}\mathbb{M}_{\Gamma_1}^{-1}\mathbf{r}_{N_1}^{k+1}(\boldsymbol{\mu}), \end{cases} \tag{12}$$

where, for $i = 1, 2$, $\mathbb{M}_{\Gamma_i}$ is the *interface mass matrix* on Γ_i and $\mathbb{R}_{21}$ is the matrix associated with Π_{21}, while

$$\mathbf{r}_{N_i}^{k+1}(\boldsymbol{\mu}) = \left(\mathbb{A}_i(\boldsymbol{\mu})\mathbf{u}_{N_i}^{k+1}(\boldsymbol{\mu}) - \mathbf{f}_i(\boldsymbol{\mu}) \right)_{|\Gamma_i}. \tag{13}$$

Therefore, the complete solutions of subproblems (3) and (4) are

$$\mathbf{u}_{N_i}^{k+1}(\boldsymbol{\mu}) = \begin{bmatrix} \mathbf{u}_i^{k+1}(\boldsymbol{\mu}) \\ \mathbf{u}_{\Gamma_i}^{k+1}(\boldsymbol{\mu}) \\ \mathbf{g}_{D,i}(\boldsymbol{\mu}) \end{bmatrix}.$$

Further details on the derivation of these systems, which aree indeed quite standard in the DD literature, can be found, e.g., in [11, 25].

The residual vector is the algebraic counterpart of an element of the dual space Y'_{h_i} of Y_{h_i} – see, e.g., [45, Chapter 3] – then we define by $\mathbf{z}_{N_i}(\boldsymbol{\mu})$ the element obtained from $\mathbf{r}_{N_i}(\boldsymbol{\mu})$ by solving

$$\mathbb{M}_{\Gamma_i}\mathbf{z}_{N_i}(\boldsymbol{\mu}) = \mathbf{r}_{N_i}(\boldsymbol{\mu}). \tag{14}$$

Here $\mathbf{z}_{N_i}(\boldsymbol{\mu})$ is the algebraic counterpart of the Riesz' element associated with the residual $\mathbf{r}_{N_i}(\boldsymbol{\mu})$. In other words, the interface mass matrix becomes the transfer matrix from the Lagrange basis to the dual one and viceversa [25, 46] and the array $\mathbf{z}_{N_i}(\boldsymbol{\mu})$ represents the residual in primal form.

Therefore, the matrix $\mathbb{R}_{21}$ transfer the function of Y_{h_1} whose nodal values are stored in $\mathbf{z}_{N_i}$ (corresponding to the interface residual vectors $\mathbf{r}_{N_1}$) at the nodes on Γ_2.

Note that the conforming interface case can be recovered by taking $\mathbb{R}_{12}$ and $\mathbb{M}_{\Gamma_2}\mathbb{R}_{21}\mathbb{M}_{\Gamma_1}^{-1}$ both equal to the identity matrix.

Remark 1. When $\partial\Gamma_i \cap \partial\Omega_{i,D} \neq 0$, the residual $\mathbf{r}_{N_i}$ should be corrected to take into account the interpolation process on all the degrees of freedom of Γ_i, including those on $\partial\Gamma_i$ (see *e.g.* [25]). Even if the reduced technique presented in this paper will work in both cases, hereon we will consider only $\partial\Gamma_i \cap \partial\Omega_{i,D} = 0$.

Remark 2. The above formulation can be easily extended to time-dependent second-order parabolic PDE problems. In such cases, suitable numerical schemes are implemented to handle the time discretization and Dirichlet-Neumann subdomains iterations must be applied for each time step of the approximated solution [11]. The application of our method to a time-dependent test case will be addressed in Section 5.

Remark 3. The master and the slave solution snapshots can be directly collected from the FOM computations by solving (11)-(13) when *(i)* conforming discretizations are considered in the two subdomains or *(ii)* when interpolation/projection methods are implemented to handle non-conforming grids, *e.g.* MORTAR methods or INTERNODES. In the non-conforming case, however, if FOM interpolation or projection methods are lacking, the snapshots for each subproblem can be surrogated solving for each parameter instance the FOM problem twice, *i.e.* with two different conforming discretizations. In particular, we first define two possible FE grids in the complete domains by setting on each first subdomain a chosen spatial discretization and extending it (conformingly) to the corresponding second one. Now, both coupled problems feature interface conformity and can be solved with Dirichlet-Neumann iterations. Then, we collect the snapshots of each subproblem and the relative interface data in the discretization set in the non-conforming case for the corresponding subdomain, obtaining two sets of solution and their Dirichlet and Neumann data. Indeed, this snapshots can be seen as the model solution when interface non-conformity is considered (see Fig. 1 for a schematic scatch of the used procedure). In this paper, such techniques is used to collect the snapshots for the test cases reported in Section 5.
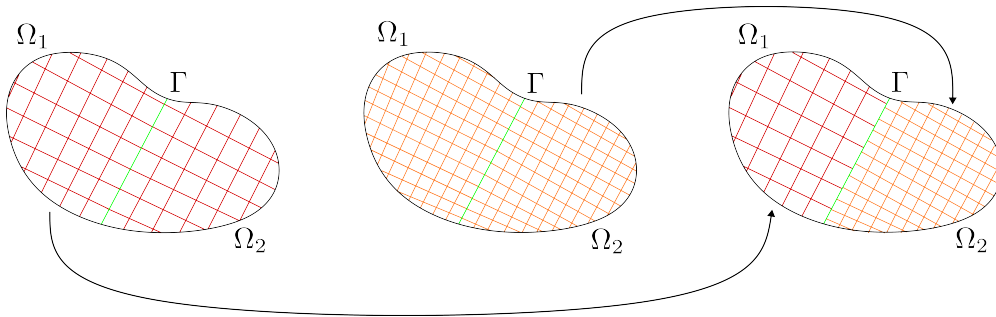

Figure 1: Schematic representation of the two discretizations of the domain Ω used to compute the FOM snapshots (left and center) and the discretization of the domain Ω used to compute the ROM snapshots (right).

3. Master and slave reduced order problems

The strategy we propose here aims at both reducing separately the two subproblems and using reduced techniques to treat also the Dirichlet and Neumann interface conditions arising from the Dirichlet-Neumann subdomains iterations applied at the reduced level. In particular, this ROM technique is an extension of the one proposed in [42] and combines different RB methods, one set for each subproblem, and the DEIM to treat both the interface conditions, by defining independent reduced order representations of the involved quantities.

We first approximate the FOM solution of the master and slave models by means of a small number of basis functions defined in the corresponding subdomain Ω_i and selected through a POD procedure. Moreover, using the DEIM, we identify a suitable set of basis functions for the master and slave interface snapshots and use them to transfer Dirichlet and Neumann data across conforming or non-conforming interface grids. Lastly, considering the same Dirichlet-Neumann iteration scheme of the high-fidelity FOM, we iterate between the reduced solutions of the two subproblems imposing the continuity of both the interface solutions and fluxes at each iterations (see Fig. 2).

The reduced form of master and slave problems is described in this Section, while we derive the procedure to reduce parameter-dependent Dirichlet and Neumann interface conditions in Section 4.

We define the reduced version of problems (11) and (12) relying on a POD-Galerkin approach [45]. Therefore, in the offline stage we collect the set of snapshots solving the sub-FOMs for a suitable set of parameter values. In particular, we choose as snapshots the FOM slave and master solutions *at convergence* of the sub-iterations, *i.e.* $\mathbf{S}_1 = \{\mathbf{u}_1(\boldsymbol{\mu}_\ell), \boldsymbol{\mu}_\ell \in \mathscr{P}^d\}$ and $\mathbf{S}_2 = \{\mathbf{u}_2(\boldsymbol{\mu}_\ell), \boldsymbol{\mu}_\ell \in \mathscr{P}^d\}$, respectively. Sampling of the parameter space is usually done considering a latin hypercube sampling (LHS) method [47, 48].
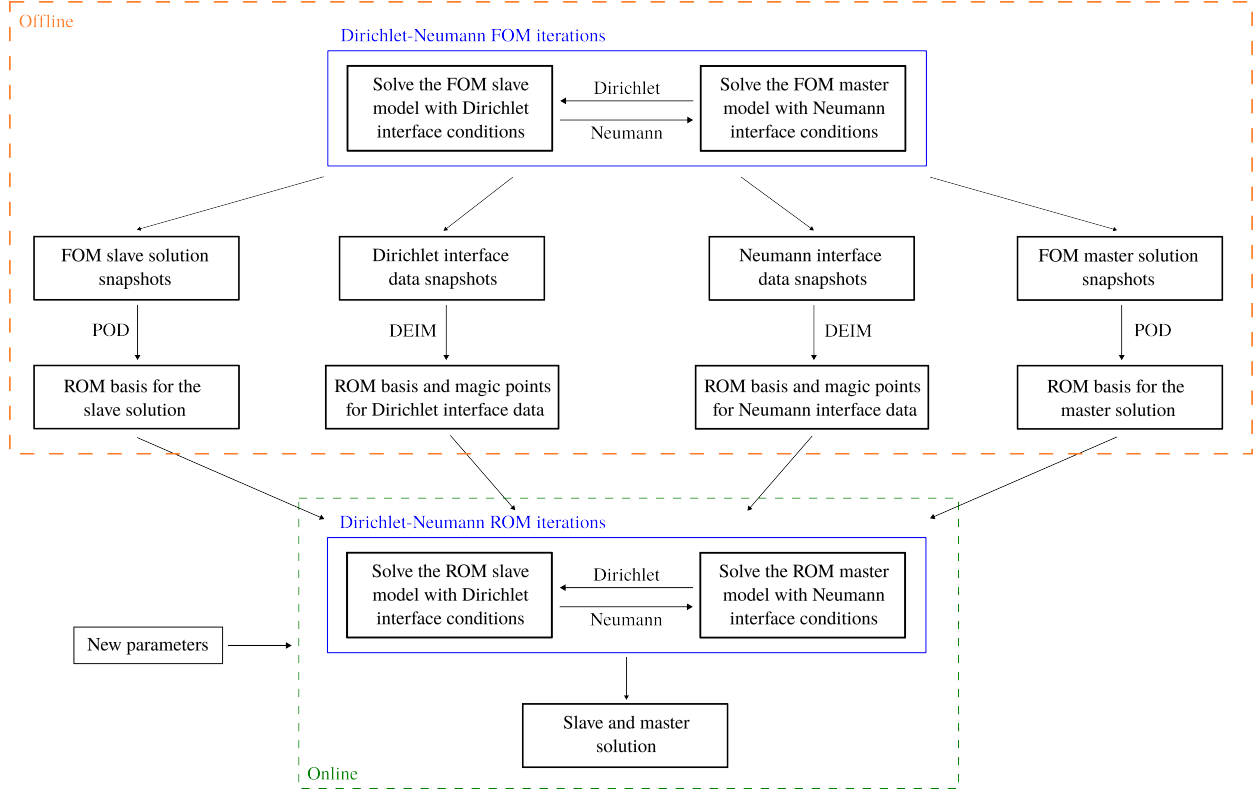
6

Figure 2: Schematic representation of the reduced order Dirichlet-Neumann domain decomposition algorithm.

Remark 4. From now on, the index k is omitted when quantities at convergence of the Dirichlet-Neumann FOM iterations are considered.

The POD technique is applied to each set of snapshots $\mathbf{S}_1$ and $\mathbf{S}_2$ and a corresponding set of reduced basis functions is computed and stored for the approximation of the solution on each subdomain. We denote by n_i the cardinality of the set of reduced basis functions. Defining $\mathbb{V}_i \in \mathbb{R}^{N_i \times n_i}$, $n_i \ll N_i$, the matrices whose columns yield the obtained basis functions, the ROM seeks for an approximation of the FOM solutions under the form

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

and

$$\mathbf{u}_2(\boldsymbol{\mu}) \approx \mathbb{V}_2 \mathbf{u}_{n_2}(\boldsymbol{\mu}).$$

Projecting problems (11) and (12) onto the reduced spaces defined by $\mathbb{V}_i$, starting from an initial guess $\mathbf{u}_{n_i}^0(\boldsymbol{\mu})$ and $\mathbf{u}_{n_2}^0(\boldsymbol{\mu})$, in the online phase, for each $k \geq 0$, we search for the reduced solutions $\mathbf{u}_{n_1}^{k+1}(\boldsymbol{\mu}) \in \mathbb{R}^{n_1}$ and $\mathbf{u}_{n_2}^{k+1}(\boldsymbol{\mu}) \in \mathbb{R}^{n_2}$ such that

$$\begin{cases} \mathbb{A}_{n_1}(\boldsymbol{\mu}) \mathbf{u}_{n_1}^{k+1}(\boldsymbol{\mu}) = \mathbf{f}_{n_1}(\boldsymbol{\mu}) - \mathbb{V}_1^T \mathbb{A}_{1,D}(\boldsymbol{\mu}) \mathbf{g}_{D,1}(\boldsymbol{\mu}) - \mathbb{V}_1^T \mathbb{A}_{1,\Gamma_1}(\boldsymbol{\mu}) \mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) \\ \mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) = \mathbb{R}_{12} \mathbf{u}_{\Gamma_2}^k(\boldsymbol{\mu}) \end{cases} \tag{15}$$

and

$$\begin{cases} \mathbb{A}_{n_2}(\boldsymbol{\mu}) \mathbf{u}_{n_2}^{k+1}(\boldsymbol{\mu}) = \mathbf{f}_{n_2}(\boldsymbol{\mu}) + \mathbb{V}_2^T \mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu}) - \mathbb{V}_2^T \mathbb{A}_{2,D} \mathbf{g}_{D,2} \\ \mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu}) = -\mathbb{M}_{\Gamma_2} \mathbb{R}_{21} \mathbb{M}_{\Gamma_1}^{-1} \mathbf{r}_{N_1}^{k+1}(\boldsymbol{\mu}), \end{cases} \tag{16}$$

where

$$\mathbb{A}_{n_i}(\boldsymbol{\mu}) = \mathbb{V}_i^T \mathbb{A}_{i,i}(\boldsymbol{\mu}) \mathbb{V}_i, \qquad \mathbf{f}_{n_i}(\boldsymbol{\mu}) = \mathbb{V}_i^T \mathbf{f}_i(\boldsymbol{\mu}), \qquad i = 1, 2.$$

7

Notice that the second equations in problems (15) and (16) (the interface equations) are defined in the FOM space, whereas the first ones are in the ROM space. The reduced version of the interface equations will be derived in the following Section.

Remark 5. For simplicity, in this paper we only consider the case of linear PDE problems. In the case of nonlinear problems, the presence of nonlinear terms in the master and slave formulations can be handled through suitable hyper-reduction techniques like, *e.g*, the DEIM [49, 50, 51]. The ROM approach would be still modular, requiring the reduction of each subproblem and the treatment of interface data as shown in the next section.

Remark 6. Time-dependent problems can be reduced with RB methods considering the time variable as an additional parameter of the model. Indeed, in such case the FOM solutions at each time step of the simulation are collected in the set of snapshots, and the reduced basis allows to approximate the time-dependent solution of each subproblem involving vectors of (reduced) degrees of freedom $\mathbf{u}_{n_1}$, $\mathbf{u}_{n_2}$ that are also time-dependent (see Section 5).

4. Parametric interface data reduction

Dealing with interface conditions highlighted, especially when using non-conforming grids, requires special care. Since the subproblems (15) and (16) are parameter-dependent, the interface data naturally inherit the parameter dependency and the DEIM [49, 50, 51, 52, 53, 54, 55] can be applied to both reduce the dimension of such data and to transfer the information across the interface grids. Indeed, using DEIM means: *(i)* compute a set of basis functions for the quantity of interest employing POD, *(ii)* use a greedy algorithm to identify a small number of dofs as weights for the corresponding basis functions (in place of the weights used in simple POD). The nodes corresponding to such dofs define the so-called reduced mesh, effectively determining a relation between the FE grids and the reduced space. Therefore, when considering interface reduction, a small number of interface nodes can be selected through the DEIM to described the complete vector of parametrized interface data. In the conforming case, DEIM can be used directly on the quantity of interest, *i.e.* the interface solution in case of Dirichlet data, or the interface residual in case of Neumann data. For non-conforming interface grids, the residuals (in primal form) must be involved to treat the Neumann terms [17].

In Subsection 4.1 and 4.2 we define the reduction of the Dirichlet and Neumann interface conditions, respectively, when non-conforming interface grids are considered. These new ROM interface conditions are used to substitute the interface equations of problems (15) and (16).

4.1. Parameter-dependent Dirichlet data

The parametric interpolation method of the Dirichlet data used in this work is similar to the one introduced in [42]. Such technique relies on the DEIM and can be applied in case of both conforming and non-conforming interface grids.

First, in the offline phase we collect from the slave domain Ω_1 the interface snapshots, *i.e.* we extract the interface (Dirichlet) degrees of freedom obtained for different instances of the parameter vectors from the FOM computation. Notice that, as for the solution reduction, we select only the interface dofs at convergence of the FOM Dirichlet-Neumann iterations, namely

$$\mathbf{S}_D = \{\mathbf{u}_{\Gamma_1}(\boldsymbol{\mu}_\ell), \ \boldsymbol{\mu}_\ell \in \mathscr{P}^d\}.$$

Let us denote by $N_{1,\Lambda}$ the number of FOM dofs on Γ_1. A low-dimensional representation of the interface dofs can then be computed by determining a set of $M_1 \ll N_{1,\Lambda}$ POD basis functions from $\mathbf{S}_D$ that we store in the matrix $\boldsymbol{\Phi}_D$, with the purpose of getting

$$\mathbf{u}_{\Gamma_1}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}_D \mathbf{u}_{1,M_1}(\boldsymbol{\mu}),$$

where $\mathbf{u}_{1,M_1}(\boldsymbol{\mu})$ is a vector of M_1 coefficients. Furthermore, with a greedy algorithm [53], we select iteratively M_1 indices in $\{1,\dots,N_{1,\Lambda}\}$, by minimizing the interpolation error over the interface snapshots set $\mathbf{S}_D$, according to the maximum norm. the set of such indices is denoted by

$$\mathcal{I}_{1,D} \subset \{1,\dots,N_{1,\Lambda}\}, \quad \text{with cardinality } |\mathcal{I}_{1,D}| = M_1. \tag{17}$$

The points corresponding to th indices of $\mathcal{I}_{1,D}$, are usually referred to as *magic points* on Γ_1, and are used to impose the Dirichlet interface conditions on Γ_1 for the reduced online problem. Let us denote by $\mathbf{u}_{\Gamma_1,|\mathcal{I}_{1,D}}(\boldsymbol{\mu})$ the vector of the FOM dofs at the magic points.

In the online phase, at each Dirichlet-Neumann iteration k, we ask that the reduced interface vector $\mathbf{u}_{1,M}^{k+1}(\boldsymbol{\mu})$ satisfies the relation

$$\boldsymbol{\Phi}_{D|\mathcal{I}_{1,D}} \mathbf{u}_{1,M}^{k+1}(\boldsymbol{\mu}) = \mathbf{u}_{\Gamma_{1|\mathcal{I}_{1,D}}}^{k+1}(\boldsymbol{\mu}),$$

where $\boldsymbol{\Phi}_{D|\mathcal{I}_{1,D}} \in \mathbb{R}^{M_1 \times M_1}$ is the sub-matrix of $\boldsymbol{\Phi}_D$ containing the $\mathcal{I}_{1,D}$ rows specified in (see *e.g.* [51] for the well-posedness of the above procedure.). The FOM interface dofs on Γ_1 can be then approximated as

$$\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D|\mathcal{I}_{1,D}}^{-1} \mathbf{u}_{\Gamma_{1|\mathcal{I}_{1,D}}}^{k+1}(\boldsymbol{\mu}). \tag{18}$$

Now we replace $\mathbf{u}_{\Gamma_{1|\mathcal{I}_{1,D}}}^{k+1}$ with the values of the master solution $\mathbf{u}_{N_2}^k$ at other points on Γ_2, the ones corresponding to the magic points on Γ_1. Thus, given the position $\mathbf{p}_1$ of the magic point corresponding to the index $i_{1,D} \in \mathcal{I}_{1,D}$ in Cartesian coordinates, we search for the corresponding node in the master interface, *i.e.* for the point $\mathbf{p}_2 \in \Gamma_2$ such that

$$\mathbf{p}_2 = \min_{\mathbf{p}_2^j \in \mathrm{nodes}_{\Gamma_2}} (\mathrm{dist}(\mathbf{p}_1 - \mathbf{p}_2^j)).$$

Remark 7. When the solution of the above minimization problem is not unique, we choose as $\mathbf{p}_2$ the last node found by the algorithm that satisfies such relation.

Then, we can define the set $\mathcal{I}_{2,D}$ of the indices on the master grid corresponding to the indices in $\mathcal{I}_{1,D}$, *i.e.*

$$\mathcal{I}_{2,D} = \{i_{2,D}(i_{1,D})\}_{i_{1,D} \in \mathcal{I}_{1,D}}.$$

Notice that $\mathcal{I}_{2,D}$ is computed in the offline phase.

Finally, in the online phase, we replace the FOM interface values ath the magic points on Γ_1 are with the values of the master solution at the magic points on Γ_2, *i.e.*,

$$\left(\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu})\right)_{i_{1,D}} = \left(\mathbf{u}_{\Gamma_2}^k(\boldsymbol{\mu})\right)_{i_{2,D}(i_{1,D})}, \quad i_{1,D} \in \mathcal{I}_{1,D}. \tag{19}$$

More briefly, we write $\mathbf{u}_{\Gamma_{1|\mathcal{I}_{1,D}}}^{k+1}(\boldsymbol{\mu}) = \mathbf{u}_{\Gamma_{2|\mathcal{I}_{2,D}}}^k(\boldsymbol{\mu})$, so that (18) becomes

$$\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D|\mathcal{I}_{1,D}}^{-1} \mathbf{u}_{\Gamma_{2|\mathcal{I}_{2,D}}}^k(\boldsymbol{\mu}).$$

Remark 8. The substitution (19) can be interpreted as a low order interpolation process: first we build the piecewise function $\tilde{\mathbf{u}}_{\Gamma_2}^k(\boldsymbol{\mu})$ (the orange one in Fig. 3) that interpolates the values of $\mathbf{u}_{\Gamma_2}^k(\boldsymbol{\mu})$ at the magic points on Γ_2 (the blue dots); then the values $\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu})$ at the magic points on Γ_1 (the red symbols) are obtained by evaluating the function $\tilde{\mathbf{u}}_{\Gamma_2}^k(\boldsymbol{\mu})$ at such points.

Note that $\mathbf{u}_{\Gamma_{2|\mathcal{I}_{2,D}}}^k(\boldsymbol{\mu})$ refers to the approximation of the FOM solution of the master problem that must, therefore, be computed from the ROM solution $\mathbf{u}_{n_2}^k(\boldsymbol{\mu})$ during the online phase. However, only a part of the
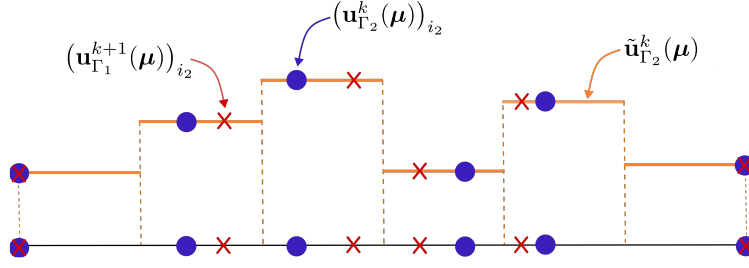
Figure 3: Geometrical interpretation of the interface reduction with the DEIM. The blue points represent the magic points on Γ_2 and the values of $\mathbf{u}_{\Gamma_2}^k$ at these magic points, the orange lines represent the piecewise constant interpolating function $\tilde{\mathbf{u}}_{\Gamma_2}^k$, while the red crosses are the magic points on Γ_1 and the values of $\mathbf{u}_{\Gamma_1}^{k+1}$ at these magic points.

approximated FOM master solution is needed, meaning the one at the magic points on Γ_2. One can store the FOM solution at the interface only at the magic points by pre-multiplying the ROM master solution by those rows of $\mathbb{V}_2$ corresponding to the magic points, *i.e.* by the matrix $\mathbb{V}_{2|\mathcal{I}_{2,D}}$.

Therefore, the action of the operator $\mathbb{R}_{12}$ in (15) (in *both* conforming and non-conforming cases), can be summarized by

$$\mathbb{R}_{12}\mathbf{u}_{\Gamma_2}^k(\boldsymbol{\mu}) = \boldsymbol{\Phi}_D\boldsymbol{\Phi}_{D_{|\mathcal{I}_{1,D}}}^{-1}\mathbb{V}_{2|\mathcal{I}_{2,D}}\mathbf{u}_{n_2}^k(\boldsymbol{\mu}),$$

and the interface condition in (15) becomes

$$\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) = \boldsymbol{\Phi}_D\boldsymbol{\Phi}_{D_{|\mathcal{I}_{1,D}}}^{-1}\mathbb{V}_{2|\mathcal{I}_{2,D}}\mathbf{u}_{n_2}^k(\boldsymbol{\mu}).$$

In this way, the last term of (15) can be approximated as

$$\mathbb{V}_1^T\mathbb{A}_{1,\Gamma_1}(\boldsymbol{\mu})\mathbf{u}_{\Gamma_1}^{k+1}(\boldsymbol{\mu}) = \mathbb{V}_1^T\mathbb{A}_{1,\Gamma_1}(\boldsymbol{\mu})\mathbb{R}_{12}\mathbf{u}_{\Gamma_2}^k(\boldsymbol{\mu}) = \mathbb{V}_1^T\mathbb{A}_{1,\Gamma_1}(\boldsymbol{\mu})\boldsymbol{\Phi}_D\boldsymbol{\Phi}_{D_{\mathcal{I}_{1,D}}}^{-1}\mathbb{V}_{2|\mathcal{I}_{2,D}}\mathbf{u}_{n_2}^k(\boldsymbol{\mu}), \quad (20)$$

where the matrix product $\mathbb{V}_1^T\mathbb{A}_{1,\Gamma_1}\boldsymbol{\Phi}_D\boldsymbol{\Phi}_{D_{\mathcal{I}_{1,D}}}^{-1}\mathbb{V}_{2|\mathcal{I}_{2,D}}$ does not depend on the solution and can be pre-computed and stored in the offline phase.

Note that even if the interface snapshots stored in $\boldsymbol{\Phi}_D$ contain the dofs on Γ_2 at convergence of the Dirichlet-Neumann iterations, the reduced coupled model is solved iterating between the reduced master and slave models. For this reason, we have written the index k to the quantities computed in the online stage.

Remark 9. As for the FOM computation, an initial guess of the Dirichlet interface conditions must be considered, but only at the magic points on Γ_2. Therefore, for $k = 0$, the approximated FOM solution $\mathbb{V}_{2|\mathcal{I}_{2,D}}\mathbf{u}_{n_2}^0$ at the magic points can be substituted with the FOM initial guess $\mathbf{u}_{2|\mathcal{I}_{2,D}}^0$.

Remark 10. Note that if the coupled problem is unsteady, to take into account the time variations of the solution, the interface data at convergence of the Dirichlet-Neumann ˘iterations for each time instant $n = 1, \ldots, N_t$ must be collected in the set of snapshots. Once the POD basis has been computed on this set of snapshots, the interpolation of the Dirichlet data can be performed as in the steady case.

4.2. Parameter-dependent Neumann data

The DEIM used to interpolate the parametric Dirichlet interface conditions can be also applied to the parametric Neumann interface conditions; as before, let us detail the case of a steady problem.

If the interface grids are conforming, for each $k \geq 0$,

$$\mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu}) = \mathbf{r}_{N_1}^{k+1}(\boldsymbol{\mu})$$

so that the DEIM can be used on the interface residual. Instead, in the non-conforming case, the interface mass matrices are involved (see Section 2). However, recalling definition (14) of $\mathbf{z}_{N_i}(\boldsymbol{\mu})$, *i.e.* the algebraic

counterpart of the Riesz' element associated with the residual, the second equation of (16) can be replaced by

$$\mathbf{z}_{N_2}^{k+1}(\boldsymbol{\mu}) = -\mathbb{R}_{21}\mathbf{z}_{N_1}^{k+1}(\boldsymbol{\mu}). \tag{21}$$

Therefore, vector $\mathbf{z}_{N_2}^{k+1}$ is the quantity to be reduced and reconstructed using the DEIM.

For the sake of generality, in this Subsection we derive the Neumann data approximation for the non-conforming case, however the same procedure also holds for the conforming case.

Starting from (13)-(14), for each $\boldsymbol{\mu}$ we compute the interface residual snapshots,

$$\mathbf{S}_N = \{\mathbf{z}_{N_2}(\boldsymbol{\mu}_\ell), \boldsymbol{\mu}_\ell \in \mathscr{P}^d\}.$$

We remind that these snapshots are saved at convergence of the FOM Dirichlet-Neumann iterations.

Applying the POD, a set of M_2 basis functions is selected and stored in $\boldsymbol{\Phi}_N$, being $M_2 \ll N_{2,\Lambda}$ (recall that $N_{2,\Lambda}$ is the number of mesh points on Γ_2). Given a generic $\boldsymbol{\mu}$, the vector $\mathbf{z}_{N_2}$ is, therefore, approximated as

$$\mathbf{z}_{N_2}(\boldsymbol{\mu}) = \boldsymbol{\Phi}_N \mathbf{z}_{2,M_2}(\boldsymbol{\mu}).$$

Then, M_2 magic points on Γ_2 are selected through a greedy algorithm and their indices in the master grid numbering are collected in the set

$$\mathcal{I}_{2,N} \subset \{1, \dots, N_{2,\Lambda}\}, \quad \text{with cardinality } |\mathcal{I}_{2,N}| = M_2.$$

In the online phase, at each Dirichlet-Neumann iteration k, we need to find $\mathbf{z}_{2,M_2}^{k+1}(\boldsymbol{\mu})$ such that

$$\boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}} \mathbf{z}_{2,M_2}^{k+1}(\boldsymbol{\mu}) = \mathbf{z}_{N_{2_{|\mathcal{I}_{2,N}}}}^{k+1}(\boldsymbol{\mu}),$$

where $\boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}$ is the restriction of $\boldsymbol{\Phi}_N$ to the indices associated with the magic points identified by $\mathcal{I}_{2,N}$, while $\mathbf{z}_{2|\mathcal{I}_{2,N}}^{k+1}(\boldsymbol{\mu})$ is the restriction of $\mathbf{z}_{N_2}^{k+1}(\boldsymbol{\mu})$ at such magic points. Then, we can approximate

$$\mathbf{z}_{N_2}^{k+1}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}^{-1} \mathbf{z}_{N_{2_{|\mathcal{I}_{2,N}}}}^{k+1}(\boldsymbol{\mu}).$$

Now, $\mathbf{z}_{2|\mathcal{I}_{2,N}}^{k+1}(\boldsymbol{\mu})$ is replaced by the values of $\mathbf{z}_{N_1}^{k+1}(\boldsymbol{\mu})$ extracted at the magic points on Γ_1 corresponding to those on Γ_2. As done in the previous Subsection, to find such magic points on Γ_1, we first need to select the set of indices $\mathcal{I}_{1,N}$ on the master interface Γ_1 corresponding to the magic points identified by $\mathcal{I}_{2,N}$. Therefore, denoting by $\mathbf{p}_2$ the node in cartesian coordinates corresponding to the index $i_{2,N}$ in $\mathcal{I}_{2,N}$, we search for

$$\mathbf{p}_1 = \min_{\mathbf{p}_1^j \in \text{nodes}_{\Gamma_1}} (\text{dist}(\mathbf{p}_2 - \mathbf{p}_1^j)).$$

Then, calling $i_{1,N}(i_{2,N})$ the index of $\mathbf{p}_1$ in the numbering of the slave grid, we define the set of indices in the slave grid corresponding to the nodes defined by the indices in $\mathcal{I}_{2,N}$, *i.e.*

$$\mathcal{I}_{1,N} = \{i_{1,N}(i_{2,N})\}_{i_{2,N} \in \mathcal{I}_{2,N}}.$$

Thus, in the online phase, we impose that

$$\left(\mathbf{z}_{\Gamma_2}^{k+1}(\boldsymbol{\mu})\right)_{i_{2,N}} = \left(\mathbf{z}_{\Gamma_1}^{k+1}(\boldsymbol{\mu})\right)_{i_{1,N}(i_{2,N})}, \quad \text{with } i_{2,N} \in \mathcal{I}_{2,N}$$

or more briefly $\mathbf{z}_{2|\mathcal{I}_{2,N}}^{k+1}(\boldsymbol{\mu}) = \mathbf{z}_{1|\mathcal{I}_{1,N}}^{k+1}(\boldsymbol{\mu})$, so that

$$\mathbf{z}_{N_2}^{k+1}(\boldsymbol{\mu}) \approx \mathbb{R}_{21}\mathbf{z}_{N_1}^{k+1}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}^{-1} \mathbf{z}_{1|\mathcal{I}_{1,N}}^{k+1}(\boldsymbol{\mu})$$

and, similarly to the Dirichlet interpolation, the operator $\mathbb{R}_{21}$ here is represented by

$$\mathbb{R}_{21}\mathbf{z}_{N_1}^{k+1}(\boldsymbol{\mu}) = \boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}^{-1} \mathbf{z}_{1_{|\mathcal{I}_{1,N}}}^{k+1}(\boldsymbol{\mu}),$$

Finally, recalling formula (14), we can recover the (Neumann) interface condition in the second equation of (16) as

$$\mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu}) \approx -\mathbb{M}_{\Gamma_2}\boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}^{-1} \mathbb{M}_{\Gamma_{1_{|\mathcal{I}_{1,N}}}} \mathbf{r}_{N_{1_{|\mathcal{I}_{1,N}}}}^{k+1}(\boldsymbol{\mu}). \tag{22}$$

Note that the matrix product $\mathbb{V}_2^T\mathbb{M}_{\Gamma_2}\boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{1,N}}}^{-1} \mathbb{M}_{\Gamma_{1_{|\mathcal{I}_{2,N}}}}$ (that appears in (16) in evaluating $\mathbb{V}_2^T\mathbf{r}_{N_2}^{k+1}(\boldsymbol{\mu})$) does not depend on $\boldsymbol{\mu}$ and can be therefore computed and stored in the offline phase.

Moreover, recalling the definition (13) of the interface residual, we can recover the dependency of such vectors on the reduced master solution, and it holds

$$\mathbf{r}_{N_{1_{|\mathcal{I}_{1,N}}}}^{k+1}(\boldsymbol{\mu}) \approx \left((\mathbb{A}_1(\boldsymbol{\mu})\mathbb{V}_1\mathbf{u}_{n_1}^{k+1}(\boldsymbol{\mu}) - \mathbf{f}_1(\boldsymbol{\mu}))_{|\Gamma_1}\right)_{|\mathcal{I}_{1,N}}. \tag{23}$$

However, only the interface residual at the magic points must be computed, reducing the number of floating point operations to compute the matrix-vector operations $\mathbb{A}_1(\boldsymbol{\mu})\mathbb{V}_1\mathbf{u}_{n_1}^{k+1}(\boldsymbol{\mu})$ from $N_1 \times n_1$ to $M_2 \times n_1$.

We notice that magic points on Γ_1 (Γ_2, respectively) found during the Dirichlet phase do not necessarily coincide with the magic points on the same interface during the Neumann phase.

We summarize the interface DEIM reduction, considering both Dirichlet and Neumann processing, in algorithm 1–2, while the complete reduction of the two-way coupled model can be found in algorithms 3–4 concerning the offline training and the online query of the ROM.

We remark that when the interface grids are conforming, a perfect matching between the corresponding nods on the master and slave interface is found. However, this does not happen in the non-conforming case. In such case we are introducing interpolation errors for both the Dirichlet and Neumann interface approximations, especially when the grids differ substantially. Considering the numerical tests of Section 5, to minimize such errors a possible remedy is to consider a finer discretization on the master domain than in the slave one, since the Dirichlet approximation seems to suffer more from the interface difference than the Neumann one.

Another remedy consists in employing more accurate interpolation operators, like Lagrange interpolation, when the interface are geometrically conforming, or the Radial Basis functions interpolations in presence of geometrical non-conforming interfaces. This will be the subject of a future work.

Moreover, given the smaller number of DoFs in the slave interface than in the master one - considering a coarser discretization in the slave domains, as done above - we check the continuity of the interface solution using $\ell2$ norm of the difference between the approximated solutions (expressed in a high-fidelity format) restricted at the slave interface DoFs, *i.e.*

$$\|\mathbf{u}_{1_{|\mathcal{I}_{1,D}}}^{k+1} - \mathbf{u}_{2_{|\mathcal{I}_{2,D}}}^{k+1}\|_{\ell_2} < \epsilon.$$

Remark 11. If the coupled problem is unsteady, similarly to the case of parameter-dependent Dirichlet data, to take into account the time variations of the solution, the interface data at convergence of the sub-iterations for each time instant $n = 1, \ldots, N_t$ must be collected in the set of snapshots. Once the POD basis has been computed on this set of snapshots, the interpolation of the Neumann data can be performed as in the steady case.

5. Numerical results

In this Section we present numerical results obtained solving *(i)* a steady problem, namely a Dirichlet boundary value problem for a linear diffusion-reaction equation, and *(ii)* a time-dependent problem, namely an initial-boundary value problem for the heat equation with Neumann boundary conditions. In particular,

---

Algorithm 1 Interface DEIM training procedure

1: **procedure** [ROM ARRAYS] = OFFLINE(FOM arrays, $\mathscr{P}_{train}$, ϵ_{tol_D}, ϵ_{tol_N}, tol)
2:     *Dirichlet and Neumann data snapshots*
3:     **for** $\boldsymbol{\mu} \in \mathscr{P}_{train}$ **do**
4:         $\mathbf{u}_1$, $\mathbf{u}_2 \leftarrow$ solve the coupled problem (11)-(13) with Dirichlet-Neumann iterations, the convergence is achieved when $\|\mathbf{u}_{\Gamma_1}^{k+1} - \mathbf{u}_{\Gamma_2}^{k+1}\|_2 <$tol
5:         $\mathbf{u}_{\Gamma_1} \leftarrow$ extract the slave interface solution;
6:         $\mathbf{z}_{N_2} \leftarrow$ extract the master primal residual.
7:         $\mathbf{S}_D = [\mathbf{S}_D, \mathbf{u}_{\Gamma_1}]$;
8:         $\mathbf{S}_N = [\mathbf{S}_N, \mathbf{z}_{N_2}]$;
9:     **end for**
10:    *DEIM reduced-order arrays:*
11:    $\boldsymbol{\Phi}_D \leftarrow \text{POD}(\mathbf{S}_D, \epsilon_{tol_D})$; $\mathcal{I}_{1,D} \leftarrow \text{DEIM-indices}(\boldsymbol{\Phi}_D)$;
12:    $\boldsymbol{\Phi}_N \leftarrow \text{POD}(\mathbf{S}_N, \epsilon_{tol_N})$; $\mathcal{I}_{2,N} \leftarrow \text{DEIM-indices}(\boldsymbol{\Phi}_N)$;
13:    *Dirichlet magic points:*
14:    **for** $i_{1,D} \in \mathcal{I}_{1,D}$ **do**
15:        $\mathbf{p}_1 \leftarrow$ get Cartesian coordinates of $i_{1,D}$ node;
16:        $\mathbf{p}_2 = \min_{\mathbf{p}_2^j \in node_{\Gamma_2}}(\text{dist}(\mathbf{p}_1 - \mathbf{p}_2^j)) \leftarrow$ search in Γ_2 the nearest node to $\mathbf{p}_1 \in \Gamma_1$;
17:        $i_{2,D} \leftarrow$ get the Dirichlet index for $\mathbf{p}_2$;
18:        $\mathcal{I}_{2,D} = [\mathcal{I}_{2,D}, i_{2,D}]$;
19:    **end for**
20:    *Neumann magic points:*
21:    **for** $i_{2,N} \in \mathcal{I}_{2,N}$ **do**
22:        $\mathbf{p}_2 \leftarrow$ get Cartesian coordinates of $i_{2,N}$ node;
23:        $\mathbf{p}_1 = \min_{\mathbf{p}_1^j \in node_{\Gamma_1}}(\text{dist}(\mathbf{p}_2 - \mathbf{p}_1^j)) \leftarrow$ search in Γ_1 the nearest node to $\mathbf{p}_2 \in \Gamma_2$;
24:        $i_{1,N} \leftarrow$ get the Neumann index for $\mathbf{p}_1$;
25:        $\mathcal{I}_{1,N} = [\mathcal{I}_{1,N}, i_{1,N}]$;
26:    **end for**
27: **end procedure**
28:

Algorithm 2 Interface DEIM query

1: **procedure** [$\mathbf{u}_1$,$\mathbf{u}_2$] = ONLINE QUERY(ROM arrays, FOM arrays, $\boldsymbol{\mu}$, tol)
2:    $\mathbf{u}_{2|\mathcal{I}_{2,D}}(\boldsymbol{\mu}) \leftarrow$ extract Dirichlet magic points on Γ_2;
3:    $\mathbf{u}_{\Gamma_1}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D_{|\mathcal{I}_{1,D}}}^{-1} \mathbf{u}_{2|\mathcal{I}_{2,D}}(\boldsymbol{\mu}) \leftarrow$ Dirichlet DEIM approximation;
4:    use $\mathbf{u}_{\Gamma_1}(\boldsymbol{\mu})$ to solve the slave problem (15);
5:    $\mathbf{r}_{N_{1|\mathcal{I}_{1,N}}}(\boldsymbol{\mu}) \leftarrow$ extract the slave interface residual at the magic points $\mathcal{I}_{1,N}$;
6:    $\mathbf{r}_{N_{2|\Gamma_2}}(\boldsymbol{\mu}) \approx \mathbb{M}_{\Gamma_2} \boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}^{-1} \mathbb{M}_{\Gamma_1}^{-1} \mathbf{r}_{N_{1|\mathcal{I}_{1,N}}}(\boldsymbol{\mu}) \leftarrow$ Neumann DEIM approximation ;
7:    $\mathbf{r}_{N_2}(\boldsymbol{\mu}) \leftarrow$ recover the interface residual, *i.e.* the Neumann term of the master problem;
8:    apply $\mathbf{r}_{N_2}(\boldsymbol{\mu})$ to solve the master problem (16);
9: **end procedure**

---

we aim to investigate the performances of the proposed algorithm by comparing FOM and ROM results in terms of both efficiency and accuracy.

    The mathematical models and numerical methods presented in this section have been implemented in C++ and Python languages and based on life^x (`https://lifex.gitlab.io`) [56], a in-house high-performance C++ FE library mainly focused on cardiac applications based on deal.II FE core [57] (`https://www.dealii.org`). Both online and offline stages of the simulations have been performed in serial on a notebook with Intel Core

---

Algorithm 3 ROM training procedure

1: **procedure** [ROM ARRAYS] = OFFLINE TRAINING(FOM arrays, $\mathscr{P}_{train}, \epsilon_{tol_1}, \epsilon_{tol_2}, \epsilon_{tol_D}, \epsilon_{tol_N}$, tol)

2:     *Solution, Dirichlet and Neumann data snapshots*

3:     **for** $\boldsymbol{\mu} \in \mathscr{P}_{train}$ **do**

4:         $\mathbf{u}_1$, $\mathbf{u}_2 \leftarrow$ solve the coupled problem (11)-(13) with Dirichlet-Neumann iterations, the convergence is achieved when $\|\mathbf{u}_{\Gamma_1}^{k+1} - \mathbf{u}_{\Gamma_2}^{k+1}\|_2 <$tol

5:         $\mathbf{u}_{\Gamma_1} \leftarrow$ extract the slave interface solution;

6:         $\mathbf{z}_{N_2} \leftarrow$ extract the master primal vector of the residual.

7:         $\mathbf{S}_1 = [\mathbf{S}_1, \mathbf{u}_1]$; $\mathbf{S}_2 = [\mathbf{S}_2, \mathbf{u}_2]$; $\mathbf{S}_D = [\mathbf{S}_D, \mathbf{u}_{\Gamma_1}]$; $\mathbf{S}_N = [\mathbf{S}_N, \mathbf{z}_2]$;

8:     **end for**

9:     *POD reduced-order arrays:*

10:     $\mathbb{V}_1 \leftarrow$ POD($\mathbf{S}_1, \epsilon_{tol_1}$);

11:     $\mathbb{V}_2 \leftarrow$ POD($\mathbf{S}_2, \epsilon_{tol_2}$);

12:     $\{\mathbb{A}_{n_1}, \mathbf{f}_{n_1}\} \leftarrow$ Galerkin projection of the FOM slave arrays onto $\mathbb{V}_1$;

13:     $\{\mathbb{A}_{n_2}, \mathbf{f}_{n_2}\} \leftarrow$ Galerkin projection of the FOM master arrays onto $\mathbb{V}_2$;

14:     *DEIM reduced-order arrays:*

15:     $\boldsymbol{\Phi}_D \leftarrow$ POD($\mathbf{S}_D, \epsilon_{tol_D}$); $\mathcal{I}_{1,D} \leftarrow$ DEIM-indices($\boldsymbol{\Phi}_D$);

16:     $\boldsymbol{\Phi}_N \leftarrow$ POD($\mathbf{S}_N, \epsilon_{tol_N}$); $\mathcal{I}_{2,N} \leftarrow$ DEIM-indices($\boldsymbol{\Phi}_N$);

17:     *Dirichlet magic points:*

18:     **for** $i_{1,D} \in \mathcal{I}_{1,D}$ **do**

19:         $\mathbf{p}_1 \leftarrow$ get Cartesian coordinates of $i_{1,D}$ node;

20:         $\mathbf{p}_2 = \min_{\mathbf{p}_2^j \in node_{\Gamma_2}}(\text{dist}(\mathbf{p}_1 - \mathbf{p}_2^j)) \leftarrow$ search in Γ_2 the nearest node to $\mathbf{p}_1 \in \Gamma_1$;

21:         $i_{2,D} \leftarrow$ get the Dirichlet index for $\mathbf{p}_2$;

22:         $\mathcal{I}_{2,D} = [\mathcal{I}_{2,D}, i_{2,D}]$;

23:     **end for**

24:     *Neumann magic points:*

25:     **for** $i_{2,N} \in \mathcal{I}_{2,N}$ **do**

26:         $\mathbf{p}_2 \leftarrow$ get Cartesian coordinates of $i_{2,N}$ node;

27:         $\mathbf{p}_1 = \min_{\mathbf{p}_1^j \in node_{\Gamma_1}}(\text{dist}(\mathbf{p}_2 - \mathbf{p}_1^j)) \leftarrow$ search in Γ_1 the nearest node to $\mathbf{p}_2 \in \Gamma_2$;

28:         $i_{1,N} \leftarrow$ get the Neumann index for $\mathbf{p}_1$;

29:         $\mathcal{I}_{1,N} = [\mathcal{I}_{1,N}, i_{1,N}]$;

30:     **end for**

31:     $\mathbb{V}_1^T \mathbb{A}_1 \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D|\mathcal{I}_{1,D}}^{-1} \mathbb{V}_{2|\mathcal{I}_{2,D}} \leftarrow$ save matrix product for the slave trace term;

32:     $\mathbb{V}_2^T \mathbb{M}_{\Gamma_2} \boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N|\mathcal{I}_{2,N}}^{-1} \mathbb{M}_{\Gamma_1|\mathcal{I}_{2,N}} \leftarrow$ save matrix product for the master residual term;

33: **end procedure**

---

i7-10710U processor and 16GB of RAM.

5.1. Steady case: diffusion-reaction equation

In this first test case we solve the following boundary value problem for a diffusion-reaction equation: find $u \in \Omega$ such that

$$\begin{cases} -\nabla \cdot (\alpha \nabla u) + \beta u = f & \text{in } \Omega \\ u = g_D & \text{on } \partial\Omega_D \end{cases} \tag{24}$$

where the domain Ω is a hollow spheroid (see Fig. 4) with inner and outer radius equal to $0.5m$ and $3.0m$, respectively, and parameters vector $\boldsymbol{\mu} = (\alpha, \beta)$. Moreover, $f = \frac{\pi}{4} yx^2 \sin\left(\frac{\pi}{2}y\right) \exp(z-1)$, while

$$g_D = \begin{cases} 0.01 & \text{on the external sphere} \\ 0 & \text{on the internal sphere.} \end{cases}$$

14

Algorithm 4 ROM query

1: **procedure** $[\mathbf{u}_1, \mathbf{u}_2] = $ ONLINE QUERY(ROM arrays, FOM arrays, $\boldsymbol{\mu}$, tol)
2:   given the index $k \geq 0$ and the initial guess $\mathbf{u}_2^0$
3:   **while** $\|\mathbf{u}_{1|\mathcal{I}_{1,D}}^{k+1} - \mathbf{u}_{2|\mathcal{I}_{2,D}}^{k+1}\|_{\ell_2} > $ tol **do**
4:     $\mathbf{u}_{2|\mathcal{I}_{2,D}}^{k+1}(\boldsymbol{\mu}) \leftarrow$ extract Dirichlet magic points on Γ_2;
5:     $\mathbb{V}_1^T \mathbb{A}_1 \boldsymbol{\Phi}_D \boldsymbol{\Phi}_{D_{\mathcal{I}_{1,D}}}^{-1} \mathbb{V}_{2|\mathcal{I}_{2,D}} \mathbf{u}_{n_2}^{k+1} \leftarrow$ assemble the trace term;
6:     $\tilde{\mathbf{u}}_{n_1}^{k+1}(\boldsymbol{\mu}) \leftarrow$ solve the slave problem (15);
7:     $\mathbf{r}_{N_{1|\mathcal{I}_{1,N}}}^{k+1}(\boldsymbol{\mu}) \leftarrow$ extract the slave interface residual on the magic points $\mathcal{I}_{1,N}$;
8:     $\mathbb{V}_2^T \mathbb{M}_{\Gamma_2} \boldsymbol{\Phi}_N \boldsymbol{\Phi}_{N_{|\mathcal{I}_{2,N}}}^{-1} \mathbb{M}_{\Gamma_{1|\mathcal{I}_{2,N}}} \mathbf{r}_{N_{1|\mathcal{I}_{1,N}}}^{k+1}(\boldsymbol{\mu}) \leftarrow$ assemble the interface residual term;
9:     $\mathbf{u}_{n_2}^{k+1}(\boldsymbol{\mu}) \leftarrow$ solve the master (16);
10:     relax the master trace on Γ_2
11:     $k \leftarrow k + 1$
12:   **end while**
13: **end procedure**

We split the domain in two hollow spheroids with a common interface (see Fig. 4), and apply the Dirichlet-Neumann iterative scheme so that, given λ_2^0, for each $k \geq 0$, we solve the two following sub-problems

$$\begin{cases} -\nabla \cdot (\alpha \nabla u_1^{k+1}) + \beta u_1^{k+1} = f & \text{in } \Omega_1 \\ u_1^{k+1} = 0.01 & \text{on } \partial\Omega_{1,D} \\ u_1^{k+1} = \lambda_2^k & \text{on } \Gamma_1 \end{cases}$$

and

$$\begin{cases} -\nabla \cdot (\alpha \nabla u_2^{k+1}) + \beta u_2^{k+1} = f & \text{in } \Omega_2 \\ u_2^{k+1} = 0 & \text{on } \partial\Omega_{2,D} \\ \alpha \nabla u_2^{k+1} \cdot \mathbf{n}_2 = \alpha \nabla u_1^{k+1} \cdot \mathbf{n}_2 & \text{on } \Gamma_2, \end{cases}$$

while

$$\lambda_2^{k+1} = \omega u_{2|\Gamma_2}^k + (1-\omega)\lambda_2^k.$$

The acceleration parameter ω is fixed among the iterations. In particular, we set $\omega = 0.25$ when solving both the FOM and the ROM.
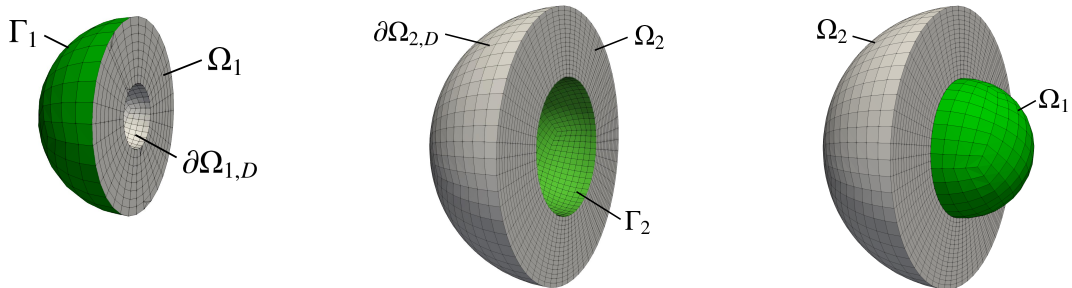

Figure 4: Slave (left) and master (center) domains. In green the interface Γ.

As FOM dimension, we consider $N = 3474$ in the slave domain Ω_1 and $N = 26146$ in the master domains Ω_2. We choose to vary the parameters α and β in $[1, 10]$ using a LHS to compute the solution snapshots. We select $N_{\text{train}} = 150$ to get a sufficiently rich snapshots set to build the reduced bases, while additional $N_{\text{test}} = 20$ values of the parameter vectors are chosen to test the method. Moreover, we prescribe for the stopping criteria of the Dirichlet-Neumann iteration a tolerance is 10^{-6}. On average, FOM solution is found after 23 iterations for the coarser discretization and 24 iterations for the finer one, while ROM solution is found after 32 iterations.
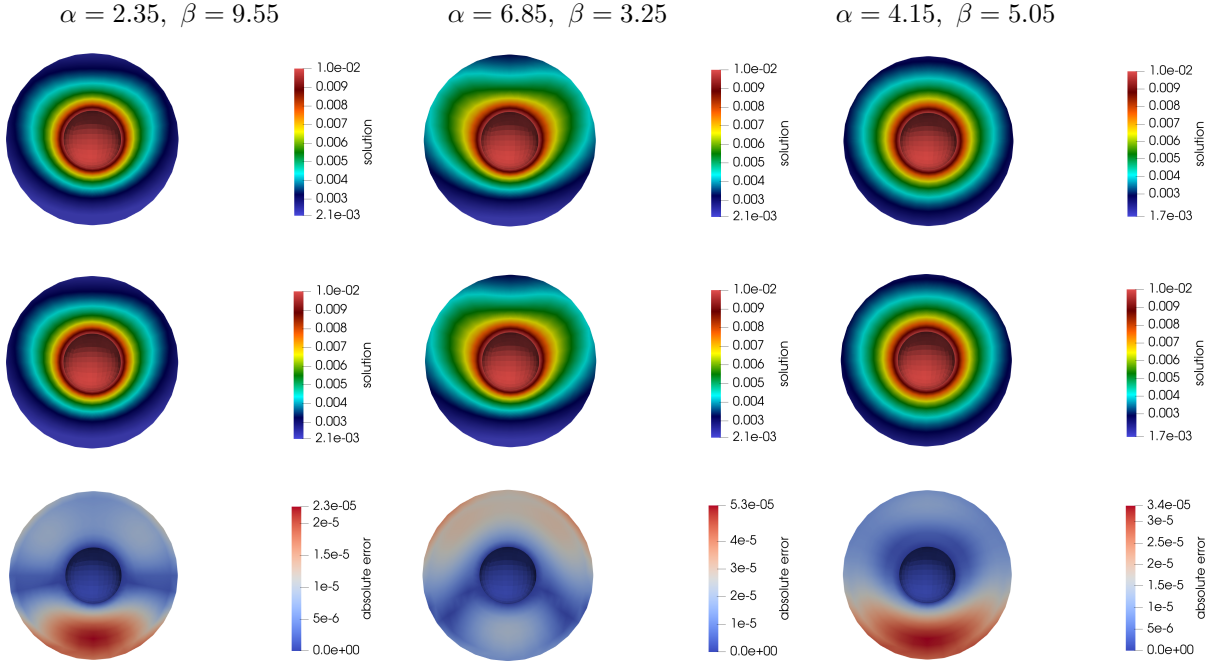
Figure 5: Slave solution FOM (top), ROM (center) solutions and absolute error (bottom) for three different parameter vectors.

As highlighted in Section 2, Remark 3, for each set of parameters we solve twice the high-fidelity FOM (once with a conforming mesh on Ω designed starting from the mesh size in Ω_1, and the other time with another conforming mesh on Ω designed starting from the mesh size in Ω_2) and once the ROM. Computational errors are evaluated considering the difference between the FOM and ROM solutions in the $H^1(\Omega_i)$ norm, *i.e.*

$$\|\mathbf{u}_{FOM}(\boldsymbol{\mu}) - \mathbf{u}_{ROM}(\boldsymbol{\mu})\|_{H^1(\Omega_i)}, \quad i = 1, 2. \tag{25}$$

Fig. 5 and 6 show three FOM and ROM solutions and the absolute error on the slave and the master domains, respectively, while in Fig. 7 we plot the 2-norm of the relative error for different dimensions of the ROM, *i.e.* for different values of the parameters n_1, n_2, M_1 and M_2 defined in Sections 3 and 4. After some tests, we chose to consider the same number of basis functions to get a fixed accuracy of the solution for the Dirichlet and Neumann data *i.e.* $M_1 = M_2$, while we treat independently the number of basis functions n_1 and n_2 used for the master and slave reduction. Then, Fig. 7 shows an increase, even if small, of the solution accuracy when the number of basis functions of one of the reduced quantities involved in the procedure is increased, as expected from RB theory.

Table 1 compares the computational costs of solving problem (24) with the proposed ROM rather than with the high-fidelity FOMs. In particular, the ROM is about 2.5 times faster than the FOM employing the coarser discretization, corresponding to a CPU time reduction of about 61%, while we are able to achieved a speed up of about 24 times compared to the FOM employing the finer discretization, corresponding to a reduction of 96% of the computational costs.
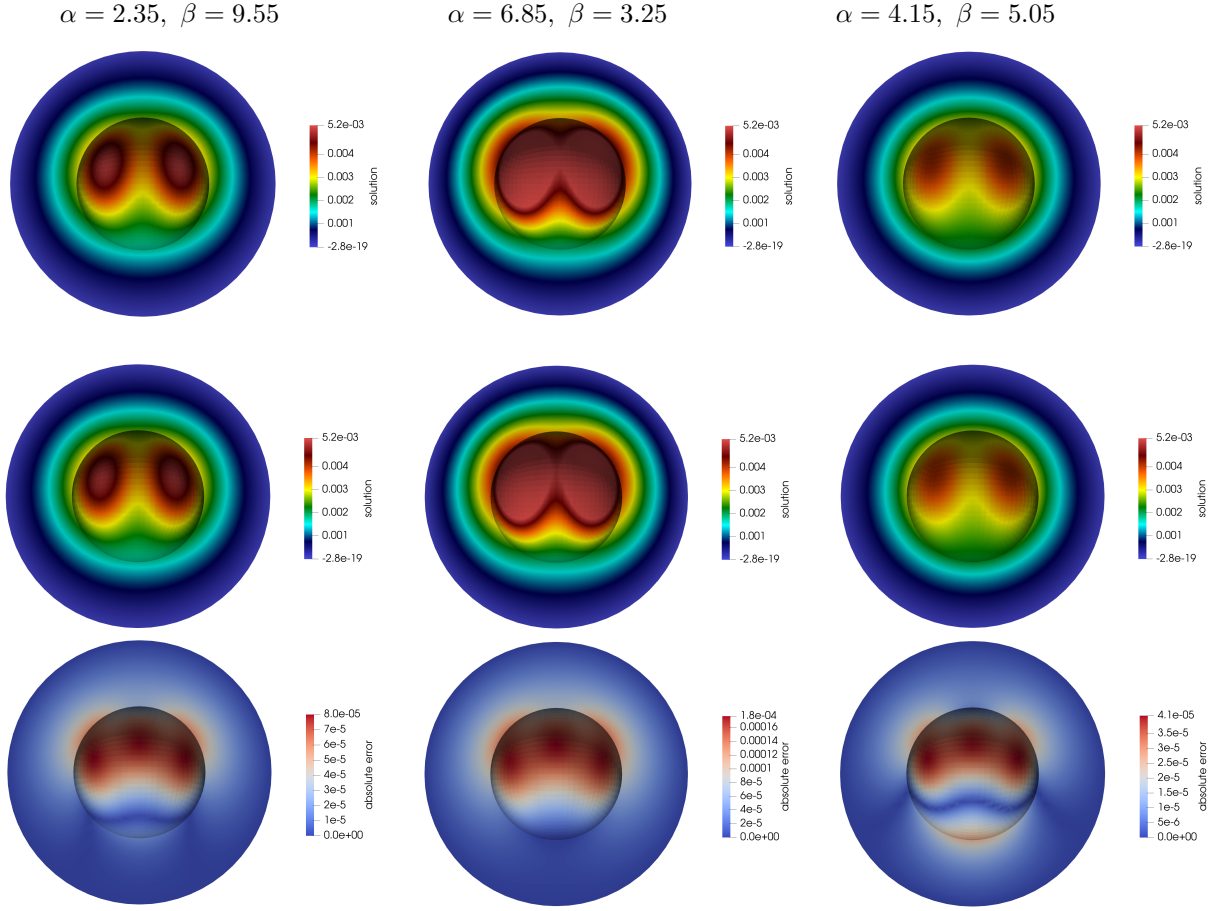
$\alpha = 2.35, \ \beta = 9.55$ $\alpha = 6.85, \ \beta = 3.25$ $\alpha = 4.15, \ \beta = 5.05$

Figure 6: Master solution FOM (top), ROM (center) solutions and absolute error (bottom) for three different parameter vectors.

| | # DoFs or # Basis | | | | CPU time | |
	Master solution	Slave solution	Master interface	Slave interface	Online	Offline
FOM	3474	3474	386	386	$\sim 3.93s$	
	26146	26146	1538	1538	$\sim 37.16s$	
ROM	7	5	6	6	$\sim 1.55s$	$\sim 1h45m$

Table 1: High fidelity and reduced order model dimensions and CPU times. We highlight the performances of the ROM model with respect to FOM solution using colors from red (worst) to green (best).

5.2. Unsteady case: time-dependent heat equation

We now apply the proposed technique to a time-dependent problem. In particular, we consider the following initial-boundary value problem for the heat equation

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nabla \cdot (\alpha \nabla u) = f & \text{in } \Omega \times (0, T) \\ \alpha \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \times (0, T) \\ u = 0 & \text{in } \Omega \times \{t = 0\} \end{cases} \qquad (26)$$
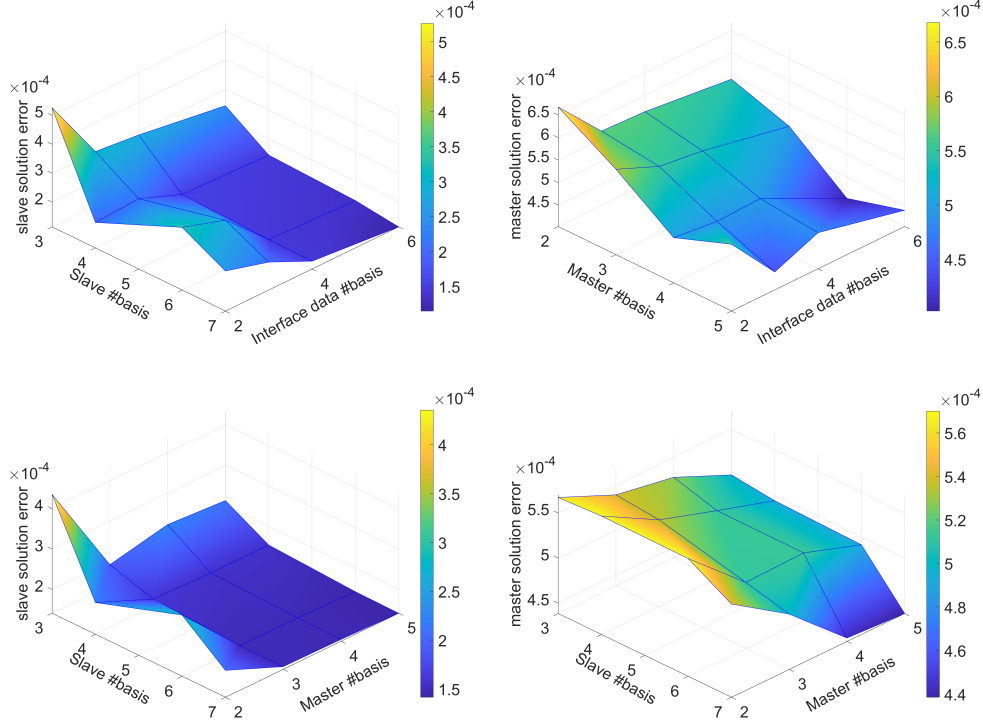
Figure 7: $H^1(\Omega_i)$ mean error (z-axis) over the solution for N_{test} different instances of the parameters between the FOM and ROM solutions varying the number of basis functions used to represent the slave and the master solution n_1 and n_2, and the interface data M_1 and M_2 (x- and y-axis). On the top row we fix the number of basis functions of the master problem to 5 (on the left) and to 7 for the slave problem (on the right), while on the bottom we fix to 6 the number of basis function for the interface data representation.

being $\Omega = (-0.5, 1.5) \times (-0.5, 0.5) \times (-0.5, 0.5)$ (see Figure 8) and

$$f = \begin{cases} 1 & \text{if } x < 0 \wedge 0.2 < t < 0.5 \\ 0 & \text{otherwise.} \end{cases}$$

The problem is parametrized through the coefficient α and – indirectly – through the time variable t.

To apply the reduced technique presented in this work, we split Ω in two cubes with a common plane $x = 0.5$ that represents the interface Γ (see Fig. 8). The application of Dirichlet-Neumann iterative scheme leads therefore to the two following sub-problems: given λ_2^0 on Γ_2, for each $k \geq 0$, solve until convergence of the interface solution

$$\begin{cases} \dfrac{\partial u_1^{k+1}}{\partial t} - \nabla \cdot (\alpha \nabla u_1^{k+1}) = f & \text{in } \Omega_1 \times (0, T) \\ \alpha \nabla u_1^{k+1} \cdot \mathbf{n}_1 = 0 & \text{on } \partial \Omega_{1,N} \times (0, T) \\ u_1^{k+1} = \lambda_2^k & \text{on } \Gamma_1 \times (0, T) \\ u_1^{k+1} = 0 & \text{in } \Omega_1 \times \{t = 0\} \end{cases} \tag{27}$$

and

$$\begin{cases} \dfrac{\partial u_2^{k+1}}{\partial t} - \nabla \cdot (\alpha \nabla u_2^{k+1}) = f & \text{in } \Omega_2 \times (0, T) \\ \alpha \nabla u_2^{k+1} \cdot \mathbf{n}_2 = 0 & \text{on } \partial \Omega_{2,N} \times (0, T) \\ \alpha \nabla u_2^{k+1} \cdot \mathbf{n}_2 = -\alpha \nabla u_1^{k+1} \cdot \mathbf{n}_1 & \text{on } \Gamma_2 \times (0, T) \\ u_2^{k+1} = 0 & \text{in } \Omega_2 \times \{t = 0\}, \end{cases} \tag{28}$$

18

while
$$\lambda_2^{k+1} = \omega u_{2_{\Gamma_2}}^k - (1 + \omega)\lambda_2^k.$$

As for the steady test case, a fixed point acceleration strategy with parameter $\omega = 0.25$ is applied to accelerate the convergence of the Dirichlet-Neumann scheme for both FOM and ROM computations.
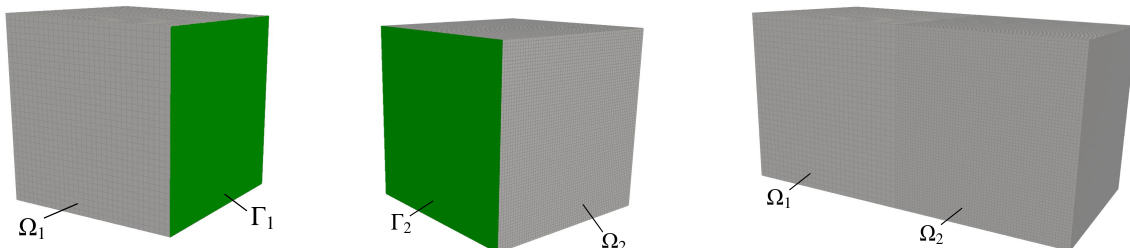

Figure 8: Ω_1 (left) and Ω_2 (center) and Ω (right). In green the interface Γ.

Two different discretizations corresponding to a FOM dimension equal to $N = 35937$ for the slave domain and $N = 274625$ DoFs for the master domain have been considered. The first order backward Euler scheme with $\Delta t = 10^{-2}s$ has been used to handle time discretization. Moreover, the diffusive coefficients α is varied between 0.5 and 5, employing also in this case a LHS strategy to select the snapshots to be computed for the ROM training. In particular, we select $N_S = N_t N_{\text{train}}$ snapshots to train the ROM, corresponding to $N_{\text{train}} = 10$ complete simulations in time, being $N_t = 100$ the number of time steps in a model simulation. Then, $N_{\text{test}} = 2$ simulations of 100 time steps each are considered to test the ROM. The tolerance to stop the Dirichlet-Neumann iteration is 10^{-6}, which correspond to 18 iterations to find the FOM solution with the coarser discretization, 19 iterations for the FOM solution with the finer discretizations and 17 iterations to find the ROM solution for each time step.

We compute the $H^1(\Omega_i)$ error (25) between the ROM and the coarse FOM solutions for the slave problem, and between the ROM and the fine FOM solutions for the master problem. In Figs. 9 and 10 we draw the ROM and FOM solutions and the absolute errors for three different time instants and some selected values of α, while in Fig. 11 we plot the $H^1(\Omega_i)$ error for different ROM dimensions. Note that fixing the number of basis functions used for both the slave solution, the master solution, and the interface Dirichlet-Neumann data, we show that the slave and master computational error decrease when increasing the number of basis functions, as expected.

Finally, in Table 2 we compare the computational costs of solving the problem with the FOM and the ROM. As before, our reduced technique shows high CPU performances, effectively achieving a speed up of 6.5 times with respect to the FOM solution with the coarsest discretization and of about 55.7 times compared to the finest discretization, corresponding a reduction of about 85% and 98% of the computational costs, respectively. These results refer to a complete simulation of 100 time steps, including some repeated tasks such as the assembling of the right-hand side, according to the time discretization scheme implemented. A hyper-reduction technique can be eventually included to enhance the online performance of the ROM even more.

| | # DoFs or # Basis | | | | CPU time | |
	Master solution	Slave solution	Master interface	Slave interface	Online	Offline
FOM	35937	35937	1089	1089	$\sim$36m 6s	
	274625	274625	4225	4225	$\sim$5h 8m	
ROM	10	10	10	10	$\sim$5m 32s	$\sim$8h

Table 2: FOM and ROM dimensions and CPU times. We highlight the ROM performances with respect to the ones of the FOM using colors from red (worst) to green (best).
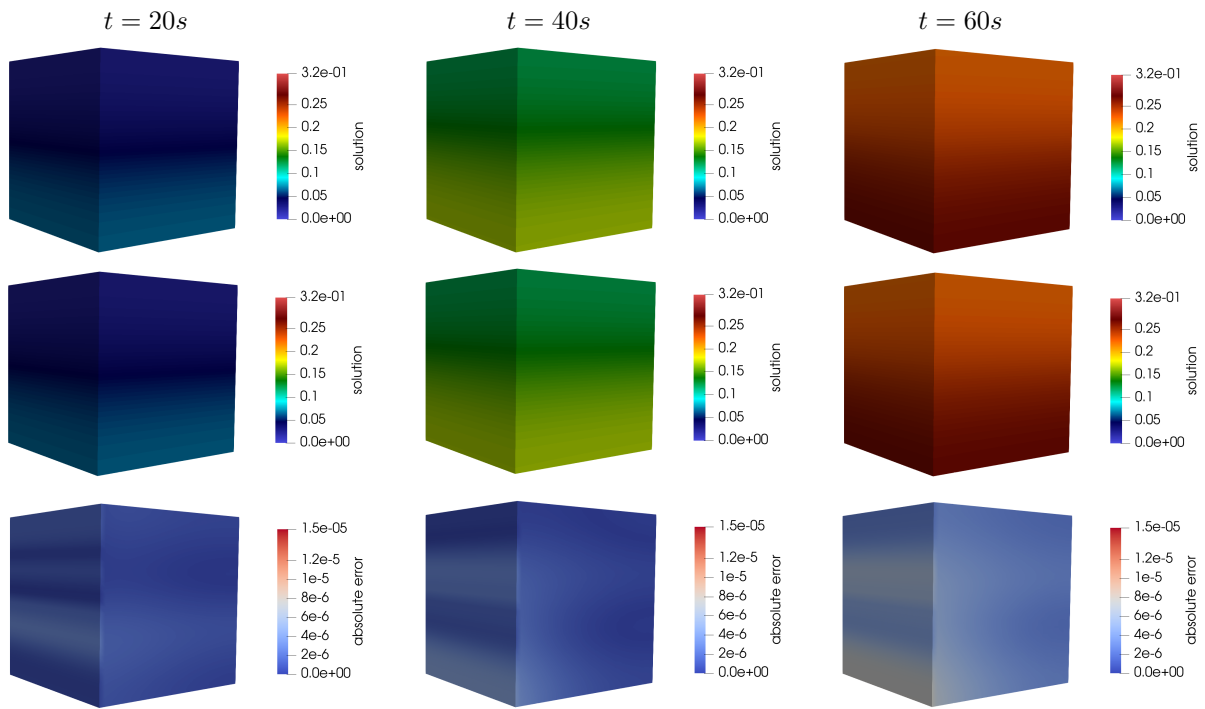
Figure 9: Slave solution FOM (top), ROM (center) solutions and absolute error (bottom) for $\alpha = 2.75$ and three different time instants.
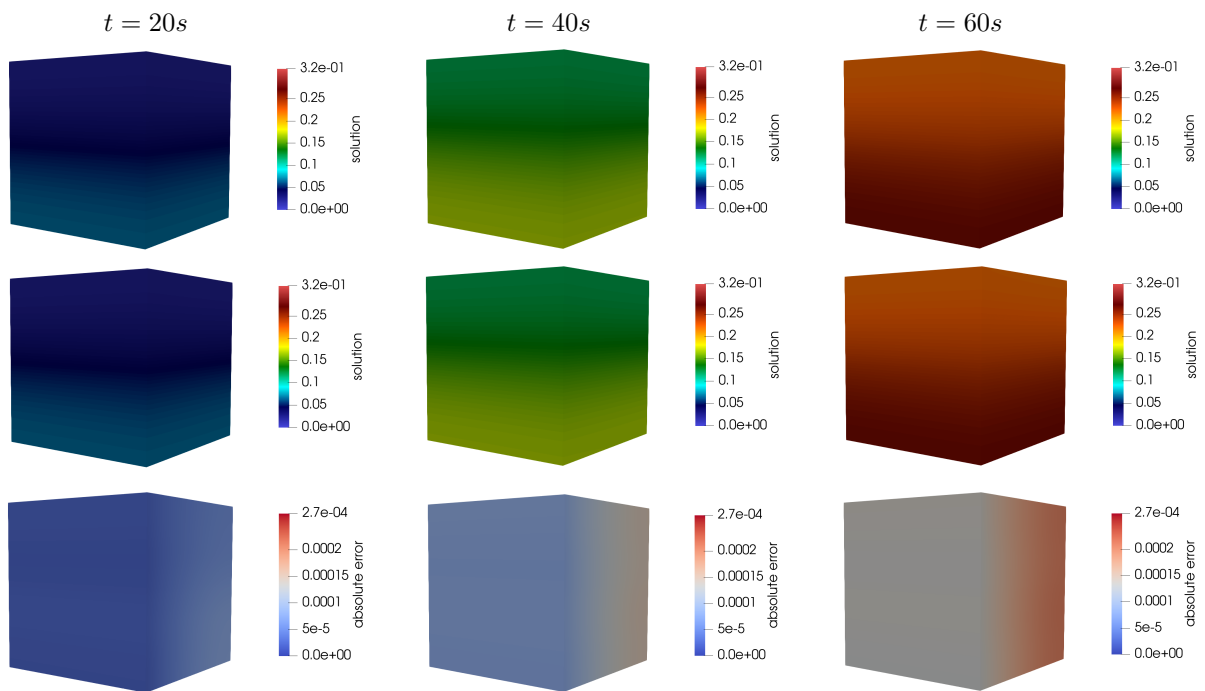

Figure 10: Master solution FOM (top), ROM (center) solutions and absolute error (bottom) for $\alpha = 2.75$ and three different time instants.
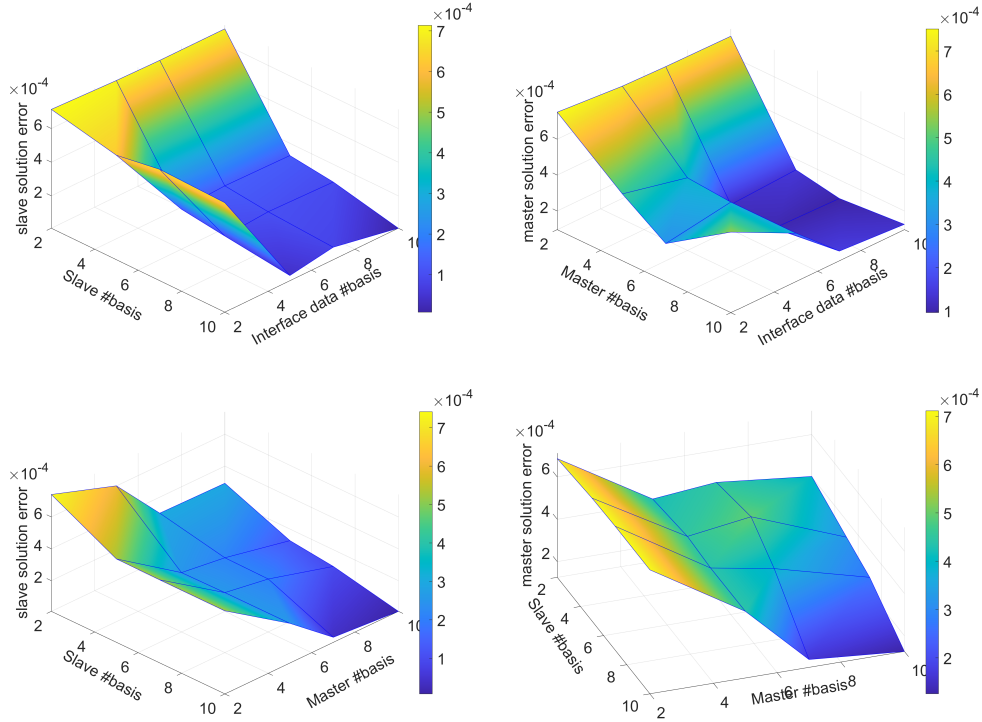
Figure 11: $H^1(\Omega_i)$ mean error (z-axis) over the solution for $N_t N_{\text{test}}$ different instances of the parameters between the FOM and ROM solutions varying the number of basis functions used to represent the slave and the master solution, and the interface data (x- and y-axis). On the top row we fix the number of basis functions of the master problem to 10 (on the left) and to 10 for the slave problem (on the right), while on the bottom we fix to 10 the number of basis function for the interface data representation.

6. Conclusion

In this work we have introduced a reduced order modeling technique based on RB methods to decrease the computational costs entailed by the solution of two-way coupled problems employing Dirichlet-Neumann iterative schemes. The modularity of the procedure ensures the efficiency of the ROM through different treatments of the master, slave and interface data reduction. Indeed, the master and the slave modes can be reduced using appropriate RB strategies, while the Dirichlet and the Neumann interface data can be handled through the DEIM inside the INTERNODES method, highlighting the possibility to use such reduced scheme to transmit the interface data between the conforming and non-conforming interface grids. The proposed algorithm can also be applied, in principle, to more general multi-physics problems that are solved through Dirichlet-Neumann domain-decomposition iterative schemes.

The numerical test cases show that our ROM is very cheap in the online phase, outperforming the online CPU time of the FOM when either fine or coarse conforming meshes in the two domains are considered. Indeed, a decrease of the 98% of the CPU time can be achieved in the case of a time-dependent parabolic PDE problem (see Subsection 5.2) with respect to a finite element FOM built over a fine discretization, and of the 85% with respect to a finite element FOM built over a coarser one.

This paper represents (at our knowledge) the first attempt towards the achievement of a fully-reduced RB interpolation based numerical scheme for two-way coupled model.

Future developments concern the use of the INTERNODES method on the FOM offline phase and the use of interpolation operators in the ROM online phase (see Remark 8) that are more accurate than those used in this paper. We are confident in this way to improve the accuracy as well as the efficiency (of the offline phase) of our approach. Finally, a future work will be the derivation of an error estimate for the proposed method.

21

Considering the results proposed in the last Section, we expect to be able to apply the presented method to more complex and relevant physics-based coupled problems, *e.g.* cardiac electrophysiological and fluid-structure interaction problems. Moreover, having been able to correctly reduce Dirichlet and Neumann interface data, the final extension of this work would be the application of our technique also to Robin-like interface conditions. All of these aspects will be the focus of future papers.

Statements and Declarations

References

[1] D. Forti, G. Rozza, Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid-structure interaction coupling problems, International Journal of Computational Fluid Dynamics 28 (2014) 158–169.

[2] D. Amsallem, J. Cortial, C. Farhat, Toward real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information, AIAA Journal 48 (2010) 2029–2037.

[3] F. Ballarin, G. Rozza, Y. Maday, Reduced-order semi-implicit schemes for fluid-structure interaction problems, in: Model Reduction of Parametrized Systems, Springer, 2017, pp. 149–167.

[4] F. Ballarin, G. Rozza, Pod-galerkin monolithic reduced order models for parametrized fluid-structure interaction problems: Pod-galerkin monolithic rom for parametrized fsi problems, International Journal for Numerical Methods in Fluids 82 (2016).

[5] S. Lorenzi, A. Cammi, L. Luzzi, G. Rozza, Pod-galerkin method for finite volume approximation of navier-stokes and rans equations, Computer Methods in Applied Mechanics and Engineering 311 (2016).

[6] B. Noack, W. Stankiewicz, M. Morzyński, P. Schmid, Recursive dynamic mode decomposition of transient and post-transient wake flows, Journal of Fluid Mechanics 809 (2016) 843–872.

[7] F. Ballarin, E. Faggiano, S. Ippolito, A. Manzoni, A. Quarteroni, G. Rozza, R. Scrofani, Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD-Galerkin method and a vascular shape parametrization, Journal of Computational Physics 315 (2016) 609–628.

[8] S. Fresca, A. Manzoni, L. Dede, A. Quarteroni, Pod-enhanced deep learning-based reduced order models for the real-time simulation of cardiac electrophysiology in the left atrium, Frontiers in Physiology 12 (2021).

[9] S. Pagani, A. Manzoni, A. Quarteroni, Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method, Computer Methods in Applied Mechanics and Engineering 340 (2018).

[10] H. Schwarz, Ueber einige abbildungsaufgaben, für die reine und angewandte Mathematik (1869) 105–120.

[11] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Numerical Mathematics and Scientific Computation, Oxford University Press, 1999.

[12] R. Glowinski, Q. Dinh, J. Periaux, Domain decomposition methods for nonlinear problems in fluid dynamics, Computer Methods in Applied Mechanics and Engineering 40 (1983) 27–109.

[13] R. Paz, M. Storti, L. Dalcin, G. Rios, Domain decomposition methods in computational fluid dynamics, Computational Mechanics Research Trends (2010) 495–555.

[14] F. Confalonieri, A. Corigliano, M. Dossi, M. Gornati, A domain decomposition technique applied to the solution of the coupled electro-mechanical problem, International Journal for Numerical Methods in Engineering 93 (2013) 137–159.

[15] Q. Weilun, S. Evans, H. Hastings, Efficient integration of a realistic two-dimensional cardiac tissue model by domain decomposition, IEEE Transactions on Biomedical Engineering 45 (1998) 372–385.

[16] W. McGee, S. Padmanabhan, Non-Conforming Finite Element Methods for Nonmatching Grids in Three Dimensions, volume 40, 2005, pp. 327–334. doi:10.1007/3-540-26825-1_32.

[17] J. Oliver, S. Hartmann, J. Cante, R. Weyler, J. Hernández, A contact domain method for large deformation frictional contact problems. part 1: Theoretical basis, Computer methods in applied mechanics and engineering 198 (2009).

[18] S. Hartmann, J. Oliver, R. Weyler, J. Cante, J. Hernández, A contact domain method for large deformation frictional contact problems. part 2: Numerical aspects, Computer Methods in Applied Mechanics and Engineering 198 (2009) 2607–2631.

[19] A. de Boer, A. van Zuijlen, H. Bijl, Review of coupling methods for non-matching meshes, Computer Methods in Applied Mechanics and Engineering 196 (2007) 1515–1525.

[20] T. Chan, B. Smith, J. Zou, Overlapping schwarz methods on unstructured meshes usingnon-matching coarse grids, Numerische Mathematik 73 (1996) 149–167.

[21] C. Bernardi, Y. Maday, F. Rapetti, Basics and some applications of the mortar element method, GAMM-Mitteilungen 28 (2005).

[22] C. Hesch, A. Gil, A. Arranz Carreño, J. Bonet, P. Betsch, A mortar approach for fluid-structure interaction problems: Immersed strategies for deformable and rigid bodies, Computer Methods in Applied Mechanics and Engineering 278 (2014).

[23] S. Deparis, D. Forti, P. Gervasio, A. Quarteroni, Internodes: an accurate interpolation-based method for coupling the galerkin solutions of pdes on subdomains featuring non-conforming interfaces, Computers & Fluids 141 (2016).

[24] P. Gervasio, A. Quarteroni, Analysis of the internodes method for non-conforming discretizations of elliptic equations, Computer Methods in Applied Mechanics and Engineering 334 (2018) 138–166.

[25] P. Gervasio, A. Quarteroni, The internodes method for non-conforming discretizations of pdes, Communications on Applied Mathematics and Computation 1 (2019) 361–401.

[26] A. Quarteroni, T. Lassila, S. Rossi, R. Ruiz Baier, Integrated heart - coupling multiscale and multiphysics models for the simulation of the cardiac function, Computer Methods in Applied Mechanics and Engineering 314 (2016).

[27] J. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, 2016. doi:10.1007/978-3-319-22470-1.

[28] P. Benner, M. Ohlberger, A. Cohen, W. Karen, Model reduction and approximation: Theory and algorithms., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi:10.1137/1.9781611974829.

[29] D. Lucia, P. King, P. Beran, Reduced order modeling of a two-dimensional flow with moving shocks, Computers & Fluids 32 (2003) 917–938.

[30] D. Xiao, C. Heaney, L. Mottet, R. Hu, D. Bistrian, E. Aristodemou, I. Navon, C. Pain, A domain decomposition non-intrusive reduced order model for turbulent flows, Computers & Fluids 182 (2019).

[31] M. Wicke, M. Stanton, A. Treuille, Modular bases for fluid dynamics, ACM Trans. Graph. 28 (2009).

[32] J. Baiges, R. Codina, S. Idelsohn, A domain decomposition strategy for reduced order models. application to the incompressible navier–stokes equations, Computer Methods in Applied Mechanics and Engineering 267 (2013) 23–42.

[33] K. Washabaugh, D. Amsallem, M. Zahr, C. Farhat, Nonlinear model reduction for cfd problems using local reduced order bases, AIAA-2012-2686, AIAA Fluid Dynamics and Co-located Conferences and Exhibit (2012).

[34] L. Iapichino, A. Quarteroni, G. Rozza, A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks, Computer Methods in Applied Mechanics and Engineering 221-222 (2012) 63–82.

[35] I. Martini, G. Rozza, B. Haasdonk, Reduced basis approximation and a-posteriori error estimation for the coupled stokes-darcy system, Advances in Computational Mathematics 41 (2015) 1131–1157.

[36] L. Pegolotti, M. Pfaller, A. Marsden, S. Deparis, Model order reduction of flow based on a modular geometrical approximation of blood vessels, Computer Methods in Applied Mechanics and Engineering 380 (2021) 113762.

[37] P. Legresley, J. Alonso, Application of proper orthogonal decomposition (pod) to design decomposition methods /, 2006.

[38] P. Kerfriden, O. Goury, T. Rabczuk, S. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, Computer Methods in Applied Mechanics and Engineering 256 (2013) 169–188.

[39] A. Corigliano, M. Dossi, S. Mariani, Model order reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem, Computer Methods in Applied Mechanics and Engineering 290 (2015) 127–155.

[40] H. Antil, M. Heinkenschloss, R. Hoppe, D. Sorensen, Domain decomposition and model reduction for the numerical solution of pde constrained optimization problems with localized optimization variables, Computing and Visualization in Science 13 (2010) 249–264.

[41] H. Antil, M. Heinkenschloss, R. Hoppe, Domain decomposition and balanced truncation model reduction for shape optimization of the stokes system, Optimization Methods and Software 26 (2011) 643–669.

[42] E. Zappon, A. Manzoni, A. Quarteroni, Efficient and certified solution of parametrized coupled problems through deim-based data projection across non-conforming interfaces, 2022. arXiv:abs/2203.09226.

[43] P. E. Bjørstad, S. C. Brenner, L. Halpern, H. H. Kim, R. Kornhuber, T. Rahman, O. B. Widlund, Domain Decomposition Methods in Science and Engineering XXIV, Lecture Notes in Computational Science and Engineering, Springer, Cham, 2018.

[44] P. Gervasio, A. Quarteroni, Internodes for heterogeneous couplings, in: P. E. Bjørstad, S. C. Brenner, L. Halpern, H. H. Kim, R. Kornhuber, T. Rahman, O. B. Widlund (Eds.), Domain Decomposition Methods in Science and Engineering XXIV, Springer International Publishing, Cham, 2018, pp. 59–71.

[45] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations. An Introduction, Springer International Publishing, 2016. doi:10.1007/978-3-319-15431-2.

[46] H. Brauchli, J. Oden, Conjugate approximation functions in finite-element analysis, Quart. Appl. Math. 29 (1971) 65–90.

[47] M. Mckay, R. Beckman, W. Conover, A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.

[48] R. Iman, J. Helton, An investigation of uncertainty and sensitivity analysis techniques for computer-models, Risk Analysis 8 (2006) 71 – 90.

[49] M. Barrault, Y. Maday, N. C. Nguyen, A. T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. Acad. Sci. Paris 339 (2004) 667–672.

[50] S. Chaturantabut, D. C. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 2009, pp. 4316–4321. doi:10.1109/CDC.2009.5400045.

[51] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (2010) 2737–2764.

[52] M. Grepl, Y. Maday, N. Nguyen, A. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM: Mathematical Modelling and Numerical Analysis 41 (2007).

[53] Y. Maday, N. Nguyen, A. Patera, G. S. H. Pau, A general multipurpose interpolation procedure: The magic points, Communications on Pure and Applied Analysis 8 (2008).

[54] F. Negri, A. Manzoni, D. Amsallem, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation, Journal of Computational Physics 303 (2015) 431–454.

[55] C. Farhat, S. Grimberg, A. Manzoni, A. Quarteroni, Computational bottlenecks for proms: precomputation and hyperreduction, in: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. Silveira (Eds.), Snapshot-Based Methods and Algorithms, De Gruyter, 2020, pp. 181–244.

[56] P. C. Africa, R. Piersanti, M. Fedele, L. Dede, A. Quarteroni, Lifex–heart module: a high-performance simulator for the cardiac function Package 1: Fiber generation, arXiv preprint arXiv:2201.03303 (2022).

[57] D. Arndt, W. Bangerth, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, U. Köcher, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Proell, K. Simon, B. Turcksin, D. Wells, J. Zhang, The `deal.II` library, version 9.3, Journal of Numerical Mathematics (2021).

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

41/2022   Arnone, A.; Ferraccioli, F.; Pigolotti, C.; Sangalli, L.M.
A roughness penalty approach to estimate densities over two-dimensional manifolds

42/2022   Gatti, F.; Fois, M.; de Falco, C.; Perotto, S.; Formaggia, L.
Parallel simulations for fast-moving landslides: space-time mesh adaptation and sharp tracking of the wetting front

40/2022   Fumagalli, A.; Patacchini, F. S.
Well-posedness and variational numerical scheme for an adaptive model in highly heterogeneous porous media

39/2022   Ferro, N.; Perotto, S.; Gavazzoni, M.
A new fluid-based strategy for the connection of non-matching lattice materials

37/2022   Boon, W. M.; Fumagalli, A.
A Reduced Basis Method for Darcy flow systems that ensures local mass conservation by using exact discrete complexes

38/2022   Burzacchi, A.; Landrò, M.; Vantini, S.
Object-oriented Classification of Road Pavement Type in Greater Maputo from Satellite Images

35/2022   Perotto, S.; Bellini, G.; Ballarin, F.; Calò, K.; Mazzi, V.; Morbiducci, U.
Isogeometric hierarchical model reduction for advection-diffusion process simulation in microchannels

36/2022   Vaccaro, F.; Brivio, S.; Perotto, S.; Mauri, A.G.; Spiga, S.
Physics-based Compact Modelling of the Analog Dynamics of HfOx Resistive Memories

34/2022   Antonietti, P.F.; Vacca, G.; Verani, M.
Virtual Element method for the Navier-Stokes equation coupled with the heat equation

33/2022   Africa, P.C.; Salvador, M.; Gervasio, P.; Dede', L.; Quarteroni, A.
A matrix-free high-order solver for the numerical solution of cardiac electrophysiology