Noname manuscript No.
(will be inserted by the editor)

Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries

Laura Iapichino · Alfio Quarteroni · Gianluigi Rozza

Received: date / Revised version: date

Abstract

The aim of this work is to solve parametrized partial differential equations in computational domains represented by networks of repetitive geometries by combining reduced basis and domain decomposition techniques. The main idea behind this approach is to compute once, locally and for few reference shapes, some representative finite element solutions for different values of the parameters and with a set of different suitable boundary conditions on the boundaries: these functions will represent the basis of a reduced space where the global solution is sought for. The continuity of the latter is assured by a classical domain decomposition approach. Test results on Poisson problem show the flexibility of the proposed method in which accuracy and computational time may be tuned by varying the number of reduced basis functions employed, or the set of boundary conditions used for defining locally the basis functions. The proposed approach simplifies the pre-computation of the reduced basis space by splitting the global problem into smaller local subproblems. Thanks to this feature, it allows dealing with arbitrarily complex network and features more flexibility than a classical global reduced basis approximation where the topology of the geometry is fixed.

1 Introduction

The numerical solution of parametrized partial differential equations ($\boldsymbol{\mu}$PDEs) in large computational domains, representing networks of repetitive geometries characterized by different parameters, is a problem arising in many applications. Notable examples are the auto-similarities of the vessels constituting the cardiovascular system or of the airways constituting the respiratory system, among many others.
This work introduces a flexible and versatile strategy that combines reduced basis (RB) method and domain decomposition (DD) techniques by offering competitive

Laura Iapichino
Delft University of Technology, Precision and Microsystems Engineering Department,
Mekelweg 2, 2628 CD Delft, The Netherlands.
E-mail: l.iapichino@tudelft.nl

Alfio Quarteroni
École Polytechnique Fédérale de Lausanne (EPFL),
Mathematics Institute of Computational Science and Engineering (MATHICSE),
Station 8, CH-1015 Lausanne, Switzerland,
Politecnico di Milano, Modelling and Scientific Computing (MOX)
Piazza Leonardo da Vinci 32, 20133 Milan, Italy (on leave).
E-mail: alfio.quarteroni@epfl.ch

Gianluigi Rozza
International School for Advanced Studies (SISSA), Mathematics Area, mathLab,
Via Bonomea 265, 34136 Trieste, Italy
E-mail: gianluigi.rozza@sissa.it

performances both in terms of solution accuracy and related computational time. Reduced order methods, and, in particular, the RB method, have been developed for elliptic $\boldsymbol{\mu}$PDEs [31] and successfully applied to Stokes [26,29,30,33] and Navier-Stokes equations in [4,5,6,13,25,34]. This technique is particularly suitable in a many query context where the solution has to be computed for many different values of the parameters. In fact it is able to drastically reduce the computational time compared to other classical numerical technique (i.e. Finite Elements (FE)) for any additional solution (during the so-called *online* stage) once an initial small set of basis functions has been computed (during the so-called *offline* stage) still retaining a certified level of accuracy. The domain decomposition method [27] is a viable procedure to further reduce the complexity of parametrized problems by decomposing the global computational domain into smaller subdomains characterized by an independent sets of parameters. In this sense, preliminary efforts to combine reduced basis and domain decomposition, in order to reduce the global parametrized problem to smaller local ones, have led to the introduction of the so-called reduced basis element method (RBEM) in the context of the Laplace problem [19,20] and of the Stokes problem [16,17], and more recently to the reduced basis hybrid (RBHM) method [11], to the static condensation method (SCRBEM) [7,8,9,23], to the reduced basis methods used for heterogeneous domain decomposition (RBHDD) [22], and to discontinuous Galerkin reduced basis element method (DGRBE) [1]. The common idea behind these methods is the definition of local small sets of basis functions, as solutions of proper $\boldsymbol{\mu}$PDEs in the subdomains composing the computational domain. Since each subdomain is parametrized by a smaller number of parameters with respect to those of the global computational domain, the effort for the offline basis computation is considerably smaller than that of a straightforward RB approach. Nevertheless, these basis functions have to be suitably chosen for defining a space for the solution of the original $\boldsymbol{\mu}$PDEs in the global domain where continuity of the solution along the internal subdomain interfaces is required. The different treatments of this critical issue represents the ingredient characterizing and differentiating the listed methods.

In this work we introduce the RDF method, whose name stems from RB, DD and FE and whose preliminary version was proposed in [10], for the solution of linear $\boldsymbol{\mu}$PDEs . The novel ingredient is on the definition of the local basis functions, that are computed as solution of subproblems defined in some reference shapes by using a small set of interface basis functions (Lagrangian or Fourier bases) as boundary conditions (BCs). Each interface basis is identified through an artificial parameter and, for this reason, we will refer to them as *parametric boundary conditions* (BCs). Such set of BCs will lead to local solutions used as a reduced basis space suitable to represent, in the online phase, the traces of the global solution along the internal interfaces. The parametric BCs allow to define flexible basis functions, with ad hoc interface profiles able to accurately recover the final solution of the problem. Moreover, a small set of local finite element (FE) basis functions is included in the reduced basis spaces in order to improve the final solution in critical parts of the domains (tipically along the coupling of the subdomains). By summarizing, the basic idea of the RDF method is to compute the global problem solution by a Galerkin projection on a small dimensional space composed by N_{FE} finite element basis functions and N_{RB} reduced basis functions, computed by using locally a set of M parametric boundary conditions. Through an a-priori error estimate and depending on the solution accuracy and the requested computational efficiency, the method is tunable by varying its three main ingredients represented by the values of N_{RB}, N_{FE} and M.

This paper is organized as follows. After this introduction, in Section 2 we provide a detailed description of the linear elliptic problem including its parametrized formulation. Section 3 is devoted to the finite element approximation of the problem,

Section 4 to a detailed description of the RDF method with the description of the local problems. In Section 5 we present an a-priori error estimate for the proposed method. In Section 6 we compare the RDF method with respect to the already existing methods combining RB and DD. Several numerical results are presented regarding the RDF accuracy and computational complexity and tests for large networks in Section 7. In Section 8 a discussion on the performance of the RDF method is carried out in light of the numerical results obtained and the theoretical analysis developed. Finally Section 9 is devoted to drawing conclusions and perspectives.

2 Problem Setting

We consider the RFD method applied to elliptic parametric PDEs in domains composed by the union of non-overlapping subdomains that can be regarded as a geometrical deformations of some reference shapes.
The aim of the method is, for a given a $\boldsymbol{\mu}$PDE, to approximate rapidly and reliably its solution, not only for different parameter values, but also for different geometrical configurations, defined as every desired combinations (e.g. network) of deformed reference shapes.

2.1 The parametric elliptic boundary value problem

Suppose that Ω is an open and bounded domain in $\mathbb{R}^2$, with Lipschitz-continuous boundary $\Gamma = \partial\Omega$. Let V_Ω and H_Ω be real, separable Hilbert spaces defined on the spatial domain Ω. We identify H_Ω with its dual space denoted by H'_Ω. Furthermore, we suppose that V_Ω is dense in H_Ω with compact embedding, i.e. $V_\Omega \hookrightarrow H_\Omega \equiv H'_\Omega \hookrightarrow V'_\Omega$. Analogously, we consider the Hilbert space V_Γ of the traces of V_Ω on the boundary Γ and its dual V'_Γ. By $\langle\cdot\,,\cdot\rangle_H$ and $\langle\cdot\,,\cdot\rangle_V$ we denote the inner products in H_Ω and V_Ω, respectively.
We assume that the symbol $\boldsymbol{\mu}_o$ represents a set of scalar parameters addressing physical features of the problem, $\boldsymbol{\mu}_o = (\mu_{o1}, \ldots, \mu_{op}) \in \mathcal{D}_o$, where $\mathcal{D}_o \subset \mathbb{R}^p$.
Let us focus on linear parametric partial differential equations ($\boldsymbol{\mu}$PDEs) written in operator form:

$$\mathscr{A}(\tilde{y}(\boldsymbol{\mu}_o); \boldsymbol{\mu}_o) = \mathscr{F}(\boldsymbol{\mu}_o) \text{ in } V'_\Omega, \tag{2.1}$$

with Dirichlet boundary conditions $\mathscr{B}_D(\tilde{y}(\boldsymbol{\mu}_o); \boldsymbol{\mu}_o) = \mathscr{G}_D(\boldsymbol{\mu}_o)$ in V_{Γ_D} and/or Neumann boundary conditions $\mathscr{B}_N(\tilde{y}(\boldsymbol{\mu}_o); \boldsymbol{\mu}_o) = \mathscr{G}_N(\boldsymbol{\mu}_o)$ in V'_{Γ_N} where Γ_D and Γ_N provide a disjoint partition of the domain boundary Γ. By $\mathscr{A}(\cdot\,; \boldsymbol{\mu}_o) : V_\Omega \to V'_\Omega$ we denote the parametric linear second-order differential operator, by $\mathscr{B}_D(\cdot\,; \boldsymbol{\mu}_o) : V_\Omega \to V_{\Gamma_D}$ and $\mathscr{B}_N(\cdot\,; \boldsymbol{\mu}_o) : V_\Omega \to V'_{\Gamma_N}$ respectively the Dirichlet and Neumann boundary conditions operator, by $\mathscr{F}(\boldsymbol{\mu}_o)$ the parametric source term and by $\mathscr{G}_D(\boldsymbol{\mu}_o)$ and $\mathscr{G}_N(\boldsymbol{\mu}_o)$ the parametric boundary data.
In order to assure the well posedness of (2.1) additional problem specific assumptions may be needed. For second-order elliptic equations we often choose the functions spaces $V_\Omega = H^1(\Omega)$, $H_\Omega = L^2(\Omega)$, $V_\Gamma = H^{1/2}(\Gamma)$.
For any $\boldsymbol{\mu}_o \in \mathcal{D}_o$ let us introduce the parameter-dependent bilinear form:

$$a(u, v; \boldsymbol{\mu}_o) = \langle \mathscr{A}(u; \boldsymbol{\mu}_o), v\rangle_{V',V} \quad \text{for } u, v \in V_\Omega \text{ and } \boldsymbol{\mu}_o \in \mathcal{D}_o,$$

which satisfies the following assumptions. There exist two constants $\beta \geq 0$ and $\alpha > 0$ independent of $\boldsymbol{\mu}_o \in \mathcal{D}_o$ such that

$$\begin{aligned} \left|a(u, v; \boldsymbol{\mu}_o)\right| &\leq \beta \left\|u\right\|_V \left\|v\right\|_V && \text{for all } u, v \in V_\Omega \text{ and } \boldsymbol{\mu}_o \in \mathcal{D}_o, \\ a(u, u; \boldsymbol{\mu}_o) &\geq \alpha \left\|u\right\|_V^2 && \text{for all } u \in V_\Omega \text{ and } \boldsymbol{\mu}_o \in \mathcal{D}_o. \end{aligned}$$

Under these assumptions there is a unique solution given by $\tilde{y}(\boldsymbol{\mu}_o)$ to (2.1). Introducing a lifting function $y_\Gamma(\boldsymbol{\mu}_o) \in V_\Omega$ of the Dirichlet boundary condition we may decompose $\tilde{y}(\boldsymbol{\mu}_o)$ as:

$$\tilde{y}(\boldsymbol{\mu}_o) = y_o(\boldsymbol{\mu}_o) + y_\Gamma(\boldsymbol{\mu}_o),$$

where $y_o(\boldsymbol{\mu}_o) \in V_\Omega^0 = \{v \in V_\Omega \,|\, \mathscr{B}_D(v; \boldsymbol{\mu}_o) = 0 \text{ in } V_{\Gamma_D}\}$.
In variational form, the problem of finding $\tilde{y}(\boldsymbol{\mu}_o)$, solution of (2.1), becomes: for $\boldsymbol{\mu}_o \in \mathcal{D}_o$, find $y(\boldsymbol{\mu}_o) \in V_\Omega^0$ such that

$$a(y_o(\boldsymbol{\mu}_o), v; \boldsymbol{\mu}_o) = f(v; \boldsymbol{\mu}_o) \quad \forall v \in V_\Omega^0, \tag{2.2}$$

with $f(v; \boldsymbol{\mu}_o) = \langle \mathscr{F}(\boldsymbol{\mu}_o) - \mathcal{A}(y_\Gamma(\boldsymbol{\mu}_o); \boldsymbol{\mu}_o), v \rangle_{V',V}$ for $v \in V_\Omega^0$ and $\boldsymbol{\mu}_o \in \mathcal{D}_o$. Note that the formulation (2.2) is general enough to embrace second order elliptic problem with Dirichlet boundary conditions or, in addition, homogeneous Neumann boundary conditions; see, e.g., in [3,14,35].

2.2 Parametrized and decomposable computational domain

We define a small number of reference shapes $\hat{\Omega}_k, k = 1, \ldots, K$ and we consider the computational domain defined as a non-overlapping union of deformed reference shapes: $\bar{\Omega} = \cup_{r=1}^{R} \bar{\Omega}_r$ where $\Omega_r = T^k(\hat{\Omega}_k, \boldsymbol{\mu}_r^k)$ and T^k is the parametric map associated to the domain $\hat{\Omega}_k$ able to deform the geometry through the parameter $\boldsymbol{\mu}^k$. Let us consider, for instance, that we want to solve problem (2.2) in a domain defined by the union of geometrical deformation of three shapes, see Figure 1.

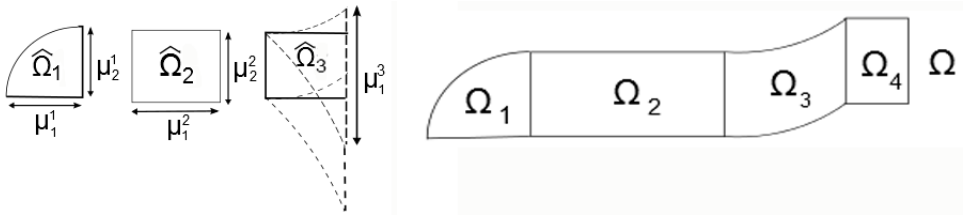

Fig. 1 Example of reference shapes (left) and computational domain Ω (right).

In general, the definition of the reference shapes, as well as their parametric maps, is arbitrary and strictly dependent on the geometric features of the desired computational domain and on the feasibility of defining suitable local maps. For instance, in the shown example, $\hat{\Omega}_2$ and $\hat{\Omega}_3$ would be represented as a single reference shape by using a more complex parametric map. Since every reference shape $\hat{\Omega}_k$ defines, through its map, a subdomain in the computational domain, the parameter vectors $\boldsymbol{\mu}^k$ (besides the maps T^k) may refer to different geometrical features and they can include a different number of parameters, such that $\boldsymbol{\mu}^k \in \mathcal{D}^k \subset \mathbb{R}^{p(k)}$ i.e. $\boldsymbol{\mu}^1 = (\mu_1^1, \mu_2^1) \in \mathcal{D}^1 \subset \mathbb{R}^2$ $(p(1) = 2)$ and $\boldsymbol{\mu}^3 = \mu_1^3 \in \mathcal{D}^3 \subset \mathbb{R}^1$ $(p(3) = 1)$. In Figure 1 a potential representation of the computational domain is shown, where $\Omega_1 = T^1(\hat{\Omega}_1, \boldsymbol{\mu}_1^1)$, $\Omega_2 = T^2(\hat{\Omega}_2, \boldsymbol{\mu}_1^2)$, $\Omega_3 = T^3(\hat{\Omega}_3, \boldsymbol{\mu}_1^3)$ and $\Omega_4 = T^2(\hat{\Omega}_2, \boldsymbol{\mu}_2^2)$.

We denote with μ_j^i a component of the vector $\boldsymbol{\mu}^i$, while with $\boldsymbol{\mu}_k^i$ a value of the vector $\boldsymbol{\mu}^i$ defining a possible deformation of the geometry. Moreover we introduce a correspondence $t(r) : \{1, \ldots, R\} \to \{1, \ldots, K\}$ able to associate for every subdomain of Ω its reference shapes; in the given example we have $t(1) = 1$, $t(2) = 2$, $t(3) = 3$ and $t(4) = 2$. With respect to the parametric maps T^k, many possible options are available, for instance simple affine maps can be defined by hand (suitable for stretching domains like $\hat{\Omega}_2$), non-affine maps can be suitable for more complex deformations, i.e. free form deformations [15], transfinite maps [11,12,18], etc. The

RB methods require to deal with affine parametric dependent problem, so that if we consider a non-affine map, the empirical interpolation method (EIM) [2] is necessary to recover the affinity of the linear and bilinear forms of the problem.
Once we introduce the reference shapes, together with their geometrical deformations and the desired computational domain Ω, we define the corresponding reference domain $\hat{\Omega} = \cup_{r=1}^{R} \hat{\Omega}_{t(r)}$ by replacing its original subdomains by their reference shapes, see Figure 2. Moreover, we define further subdomains $\hat{\Omega}_{r\Gamma}, r = 1, \dots, 4$ in every reference subdomains such that $\hat{\Gamma}_r = \partial\hat{\Omega}_{t(r)} \backslash \partial\hat{\Omega} \subset \hat{\Omega}_{r\Gamma} \subset \hat{\Omega}_{t(r)}$ and we denote with $\hat{\Omega}_\Gamma \subset \hat{\Omega}$ the union of the subdomains $\hat{\Omega}_{r\Gamma}$, $\hat{\Omega}_\Gamma = \cup_{r=1}^{R} \hat{\Omega}_{r\Gamma}$.

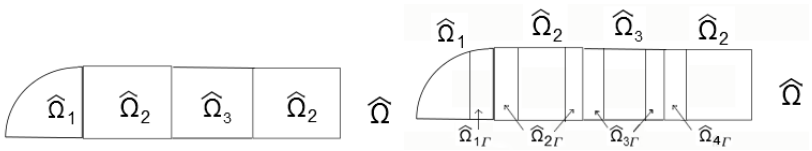

Fig. 2 Reference domain $\hat{\Omega}$ and subdomains associated to the example in Figure 1 (left). Graphical representation of the subdomains $\hat{\Omega}_{r\Gamma}, r = 1, \dots, 4$, in $\hat{\Omega}$ (right).

We recast problem (2.2) in the defined reference domain $\hat{\Omega}$. We introduce the parameter vector $\boldsymbol{\mu} = (\boldsymbol{\mu}_o, \boldsymbol{\mu}^{t(1)}, \dots, \boldsymbol{\mu}^{t(R)}) \in \mathcal{D} = \mathcal{D}_o \times \mathcal{D}^{t(1)} \times \cdots \times \mathcal{D}^{t(R)}$ and by the use of the geometrical tensors coming from the parametric maps, problem (2.2) can be defined in the reference domain as follows: for $\boldsymbol{\mu} \in \mathcal{D}$ find $y(\boldsymbol{\mu}) \in V = V^0_{\hat{\Omega}} = \{v \in V_{\hat{\Omega}} \,|\, \mathscr{B}_D(v; \boldsymbol{\mu}_o) = 0 \text{ in } V_{\hat{\Gamma}_D}\}$ such that

$$a(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V, \tag{2.3}$$

with $f(v; \boldsymbol{\mu}) = \langle \mathscr{F}(\boldsymbol{\mu}) - \mathcal{A}(y_\Gamma(\boldsymbol{\mu}); \boldsymbol{\mu}), v \rangle_{V',V}$ for $v \in V$ and $\boldsymbol{\mu} \in \mathcal{D}$. We choose the functions spaces $V_{\hat{\Omega}} = H^1(\hat{\Omega})$, $V_{\hat{\Gamma}} = H^{1/2}(\hat{\Gamma})$. Finally, by assuming that the linear and bilinear forms are affine in the parameter dependence, or otherwise can be approximated by an affine decomposition, we can write the forms of equation (2.3) as follows:

$$a(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \sum_{q=1}^{M_a} \Theta^a_q(\boldsymbol{\mu}) a_q(y(\boldsymbol{\mu}), v), \quad f(v; \boldsymbol{\mu}) = \sum_{q=1}^{M_f} \Theta^f_q(\boldsymbol{\mu}) f_q(v). \tag{2.4}$$

3 Finite element approximation

The Galerkin method to numerically solve problem (2.3) consists in finding an approximate solution $y^{\mathcal{N}}(\boldsymbol{\mu}) \in V^{\mathcal{N}}$, where $V^{\mathcal{N}}$ is a family of subspaces of V, with finite dimension $\mathcal{N}$ (typically large). The final solution $\tilde{y}^{\mathcal{N}}(\boldsymbol{\mu})$ is then recovered by adding the projection of the lifting function $y_\Gamma(\boldsymbol{\mu})$ on $V^{\mathcal{N}}$, such that $\tilde{y}^{\mathcal{N}}(\boldsymbol{\mu}) = y^{\mathcal{N}}(\boldsymbol{\mu}) + y^{\mathcal{N}}_\Gamma(\boldsymbol{\mu})$. Therefore the approximate problem becomes: find $y^{\mathcal{N}}(\boldsymbol{\mu}) \in V^{\mathcal{N}}$ such that

$$a(y^{\mathcal{N}}(\boldsymbol{\mu}), v^{\mathcal{N}}; \boldsymbol{\mu}) = f(v^{\mathcal{N}}; \boldsymbol{\mu}), \qquad \forall v^{\mathcal{N}} \in V^{\mathcal{N}}. \tag{3.1}$$

Problem (3.1) is usually called the Galerkin approximation of problem (2.3).
Using finite elements (FE) reduces to a particular choice for the subspace $V^{\mathcal{N}}$. We consider a triangulation $\mathcal{T}^{\mathcal{N}}$ of $\hat{\Omega}$ and let K be the generic element of $\mathcal{T}^{\mathcal{N}}$. In particular we use piecewise linear functions and define:

$$V^{\mathcal{N}} = X^1_{\mathcal{N}} \equiv \{v^{\mathcal{N}} \in C^0(\hat{\Omega}) : v^{\mathcal{N}}|_K \in \mathbb{P}_1 \ \ \forall K \in \mathcal{T}^{\mathcal{N}}\} \cap H^1_{0,\Gamma_D}(\hat{\Omega}). \tag{3.2}$$

We consider $\mathcal{T}^{\mathcal{N}}_r \subset \mathcal{T}^{\mathcal{N}}, r = 1, \dots, R$, the part of the triangulation $\mathcal{T}^{\mathcal{N}}$ associated to each subdomain $\hat{\Omega}_{t(r)} \backslash \hat{\Omega}_\Gamma \subset \hat{\Omega}$ and the corresponding space of functions $V^{\mathcal{N}}_r \subset V^{\mathcal{N}}$.

Moreover we denote with $\mathcal{T}_\Gamma^{\mathcal{N}} \subset \mathcal{T}^{\mathcal{N}}$ the part of the triangulation $\mathcal{T}^{\mathcal{N}}$ associated to the subdomain $\hat{\Omega}_\Gamma \subset \hat{\Omega}$ and the corresponding space $V_\Gamma^{\mathcal{N}} \subset V^{\mathcal{N}}$. To avoid further indices, we use $\mathcal{N}$ as a common index representing the finite element spaces, while each space may have a different dimensions, i.e. $\mathcal{N}_i = \dim(V_i^{\mathcal{N}}), i = 1, \dots, R$, and $\mathcal{N}_\Gamma = \dim(V_\Gamma^{\mathcal{N}})$.
Since the bilinear and linear forms in (3.1) are defined by integrals on $\hat{\Omega}$ we can split them according to the introduced triangulations, such that:

$$a(u, v, \boldsymbol{\mu}) = \sum_{r=1}^{R} a^r(u, v; \boldsymbol{\mu}) + a^\Gamma(u, v; \boldsymbol{\mu}),\ f(v; \boldsymbol{\mu}) = \sum_{r=1}^{R} f^r(v; \boldsymbol{\mu}) + f^\Gamma(v; \boldsymbol{\mu}), \quad (3.3)$$

where $a^r(\cdot, \cdot; \boldsymbol{\mu})$ and $f^r(\cdot; \boldsymbol{\mu})$ are defined in $\hat{\Omega}_{t(r)} \backslash \hat{\Omega}_\Gamma$ and $a^\Gamma(\cdot, \cdot; \boldsymbol{\mu})$ and $f^\Gamma(\cdot; \boldsymbol{\mu})$ in $\hat{\Omega}_\Gamma$ and they can be reformulated in their affine decompositions:

$$a^r(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \sum_{q=1}^{M_a^r} \Theta_q^{ra}(\boldsymbol{\mu}) a_q^r(y(\boldsymbol{\mu}), v);\ f^r(v; \boldsymbol{\mu}) = \sum_{q=1}^{M_f^r} \Theta_q^{rf}(\boldsymbol{\mu}) f_q^r(v), r = 1, \dots, R, \quad (3.4)$$

$$a^\Gamma(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \sum_{q=1}^{M_a^\Gamma} \Theta_q^{\Gamma a}(\boldsymbol{\mu}) a_q^\Gamma(y(\boldsymbol{\mu}), v);\ f^\Gamma(v; \boldsymbol{\mu}) = \sum_{q=1}^{M_f^\Gamma} \Theta_q^{\Gamma f}(\boldsymbol{\mu}) f_q^\Gamma(v). \quad (3.5)$$

The solution of problem (3.1) can be seen as:

$$y^{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{r=1}^{R} \sum_{i=1}^{\mathcal{N}_r} y_i^r \phi_i^r + \sum_{k=1}^{\mathcal{N}_\Gamma} y_k^\Gamma \phi_k^\Gamma, \quad (3.6)$$

where we denote with $\{\phi_i^r, i = 1, \dots, \mathcal{N}_r\}, r = 1, \dots, R$, a basis of $V_r^{\mathcal{N}}$ and $\{\phi_k^\Gamma, k = 1, \dots, \mathcal{N}_\Gamma\}$ a basis of $V_\Gamma^{\mathcal{N}}$. Moreover, problem (3.1) can be rewritten as follows:

$$\begin{cases} \sum_{j=1}^{\mathcal{N}_r} y_j^r a^r(\phi_j^r, \phi_i^r; \boldsymbol{\mu}) + \sum_{k=1}^{\mathcal{N}_\Gamma} y_k^\Gamma a^\Gamma(\phi_k^\Gamma, \phi_i^r; \boldsymbol{\mu}) = f^r(\phi_i^r; \boldsymbol{\mu}),\ r = 1, \dots, R,\ i = 1, \dots, \mathcal{N}_r, \\ \sum_{r=1}^{R} \sum_{i=1}^{\mathcal{N}_r} y_i^r a^r(\phi_i^r, \phi_k^\Gamma; \boldsymbol{\mu}) + \sum_{k=1}^{\mathcal{N}_\Gamma} y_k^\Gamma a^\Gamma(\phi_k^\Gamma, \phi_k^\Gamma; \boldsymbol{\mu}) = f^\Gamma(\phi_k^\Gamma; \boldsymbol{\mu}),\ k = 1, \dots, \mathcal{N}_\Gamma. \end{cases} \quad (3.7)$$

The algebraic linear system associated to (3.7) is:

$$\begin{bmatrix} \mathbb{A}_1 & & & \mathbb{A}_{1\Gamma} \\ & \ddots & & \vdots \\ & & \mathbb{A}_R & \mathbb{A}_{R\Gamma} \\ \mathbb{A}_{1\Gamma}^T & \cdots & \mathbb{A}_{R\Gamma}^T & \mathbb{A}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{Y_1} \\ \vdots \\ \mathbf{Y_R} \\ \mathbf{Y_\Gamma} \end{bmatrix} = \begin{bmatrix} \mathbb{F}_1 \\ \vdots \\ \mathbb{F}_R \\ \mathbb{F}_\Gamma \end{bmatrix}. \quad (3.8)$$

By recalling the affine decompositions (3.4) and (3.5), we define for $r = 1, \dots, R$:

$$[\boldsymbol{Y_r}]_j = y_j^r, \quad [\mathbb{A}_r]_{i,j} = \sum_{q=1}^{M_a^r} \Theta_q^{ra}(\boldsymbol{\mu}) a_q^r(\phi_j^r, \phi_i^r), \quad [\mathbb{F}_r]_j = \sum_{q=1}^{M_f^r} \Theta_q^{rf}(\boldsymbol{\mu}) f_q^r(\phi_j^r),\ \ i, j = 1, \dots, \mathcal{N}_r,$$

$$[\mathbb{A}_{r\Gamma}]_{i,j} = \sum_{q=1}^{M_a^\Gamma} \Theta_q^{\Gamma a}(\boldsymbol{\mu}) a_q^\Gamma(\phi_j^r, \phi_i^\Gamma), j = 1, \dots, \mathcal{N}_r, i = 1, \dots, \mathcal{N}_\Gamma,$$

$$[\mathbb{A}_{\Gamma\Gamma}]_{i,j} = \sum_{q=1}^{M_a^\Gamma} \Theta_q^{\Gamma a}(\boldsymbol{\mu}) a_q^\Gamma(\phi_j^\Gamma, \phi_i^\Gamma), [\boldsymbol{Y_\Gamma}]_j = y_j^\Gamma, [\mathbb{F}_\Gamma]_j = \sum_{q=1}^{M_f^\Gamma} \Theta_q^{\Gamma f}(\boldsymbol{\mu}) f_q^\Gamma(\phi_j^\Gamma), i, j = 1, \dots, \mathcal{N}_\Gamma,$$

where, $\mathcal{N}_r$ is the number of Lagrangian bases of $V_r^{\mathcal{N}}$ and $\mathcal{N}_\Gamma$ the number of Lagrangian bases on $V_\Gamma^{\mathcal{N}}$.

4 The RDF method

We propose now a new strategy to couple the finite element method and the reduced basis method in the framework of non-overlapping domain decomposition. The described approach, can be seen as *full order model* in DD framework and the solution of the problem as a Galerkin projection of problem (3.1) onto a finite element subspace $V^{\mathcal{N}}$ defined as follows:

$$V^{\mathcal{N}} = V_1^{\mathcal{N}} \oplus \cdots \oplus V_R^{\mathcal{N}} \oplus V_\Gamma^{\mathcal{N}}, \tag{4.1}$$

where $V_j^{\mathcal{N}} = span\{\phi_i^j, i = 1, \ldots, \mathcal{N}_j\}, j = 1, \ldots, R, V_\Gamma^{\mathcal{N}} = span\{\phi_k^\Gamma, k = 1, \ldots, \mathcal{N}_\Gamma\}$. The idea of the RDF method is to consider a reduced order model to solve problem (3.1), more precisely the solution is found through a Galerkin projection on the subspace $V^N \subset V^{\mathcal{N}}$ composed by the subspaces $V_{t(j)}^N \subset V_j^{\mathcal{N}}$ built by the RB functions precomputed in the reference domains $\hat{\Omega}_{t(j)}$ and the subspace $V_\Gamma^{\mathcal{N}}$ that remains the same of the FE approximation, i.e. the subspace built by the FE functions associated to the nodes belonging to the subdomain $\hat{\Omega}_\Gamma$. In other words, we replace $V^{\mathcal{N}}$ defined in (4.1) by the following composed space:

$$V^N = V_{t(1)}^N \oplus \cdots \oplus V_{t(R)}^N \oplus V_\Gamma^{\mathcal{N}}. \tag{4.2}$$

As already done in the existing RB approaches coupled with DD techniques (i.e. RBEM [19], RBHM [11] and [21]), in the RDF method the local RB functions defining the spaces V_i^N are computed through local parametric problems using a greedy algorithm. Here, a new feature of the RDF approach is represented by the use of parametric BCs in the local problems that allows the definition of effective and versatile local reduced basis functions and spaces. Moreover, we exploit the repetitiveness of the computational domain in order to deal with a low dimensionality of the parameter space. The idea is to build few local RB spaces V_k^{RB} only for the reference shapes $\hat{\Omega}_k, k = 1, \ldots, \mathrm{K}$. The bases of these RB spaces are used for defining the spaces $V_{t(j)}^N, j = 1, \ldots, R$, in $\hat{\Omega}$. Since the subdomains in $\hat{\Omega}$ are obtained by translations of $\hat{\Omega}_k$, the mapping of the corresponding functions in $\hat{\Omega}_{t(j)}$ is straightforward. In order to define these functions in the whole domain $\hat{\Omega}$, we define them equal to zero in the remaining part of the domain $\hat{\Omega} \backslash \hat{\Omega}_{t(j)}$ and in $\hat{\Omega}_\Gamma$ where we use the FE basis functions. Then we can apply the reduced order strategy.

4.1 Local parametrized problems for computing the RB functions

The computation of the RB functions is performed locally in the reference shapes and during the offline step [28]. The idea is that the RB spaces are independent on the number of subdomains that will compose the whole computational domain. The way to define these basis functions is, in principle, arbitrary and can depend on the physical characteristics of the problem. The behavior of these functions affects strongly the numerical solution of the final system we want to solve and its accuracy with respect to the exact solution of the problem.
The goal at this stage is to construct local sets of basis functions that are rich enough to define the online global space where we find an accurate and continuous final solution of the problem. To perform this task, we have to take into account two main aspects; first, these functions, defined locally, will be used as bases to

recover the solution on the whole domain Ω. In particular, this means that the BCs used on the local problems will define an interface basis space along the internal interfaces of the decomposed computational domain that has to recover the trace of the final solution. On the other hand, we have to take into account that each local problem depends on local parameters coming from the original problem and their values are unknown at this stage.
For each reference shape $\hat{\Omega}_i$ we define a local parametric problem coming from the operators of the original problem and using different local boundary conditions (BCs) on the boundaries $\hat{\Gamma}_\lambda$ identified through an additional artificial parameter μ_λ. The latter will be used for connecting the different shapes and will represent the internal interfaces of the global computational domain. By recalling the original linear parametric problem (2.1) in Ω, we define the following parametric local problems in each reference shape $\hat{\Omega}_i$, for $i = 1, \dots, K$:

$$\begin{cases} \mathscr{A}_i(\tilde{y}(\tilde{\boldsymbol{\mu}}^i); \boldsymbol{\mu}_{oi}) = \mathscr{F}_i(\boldsymbol{\mu}_{oi}) \text{ in } V'_{\hat{\Omega}_i}, \\ \mathscr{B}_{Di}(\tilde{y}(\tilde{\boldsymbol{\mu}}^i); \boldsymbol{\mu}_{oi}) = \mathscr{G}_{Di}(\boldsymbol{\mu}_{oi}) \text{ in } V_{\hat{\Gamma}_{Di}}, \\ \mathscr{B}_{Ni}(\tilde{y}(\tilde{\boldsymbol{\mu}}^i); \boldsymbol{\mu}_{oi}) = \mathscr{G}_{Ni}(\boldsymbol{\mu}_{oi}) \text{ in } V'_{\hat{\Gamma}_{Ni}}, \\ \mathscr{B}_{Di}(\tilde{y}(\tilde{\boldsymbol{\mu}}^i); \boldsymbol{\mu}_{oi}) = \lambda(\mu_\lambda) \text{ in } V_{\hat{\Gamma}_\lambda}, \end{cases} \tag{4.3}$$

where $\hat{\Gamma}_{Di}$ and $\hat{\Gamma}_{Ni}$ are parts of the external boundaries of the final computational domain. The parametric linear differential operators $\mathscr{A}_i(\cdot; \boldsymbol{\mu}_{oi})$, the BC operators $\mathscr{B}_{Di}(\cdot; \boldsymbol{\mu}_{oi})$, $\mathscr{B}_{Ni}(\cdot; \boldsymbol{\mu}_{oi})$, the source term $\mathscr{F}_i(\boldsymbol{\mu}_{oi})$ and boundary data $\mathscr{G}_{Di}(\boldsymbol{\mu}_{oi})$ and $\mathscr{G}_{Ni}(\boldsymbol{\mu}_{oi})$ come form the original problem in Ω and are defined in different domains (the reference shapes). We define $\tilde{\boldsymbol{\mu}}^i = (\boldsymbol{\mu}_{oi}, \mu_\lambda)$ where $\boldsymbol{\mu}_{oi} \in \mathcal{D}_o \times \mathcal{D}^i$ represents a set of scalar parameters addressing physical and geometrical features of the original problem in $\hat{\Omega}_i$, the parameter $\mu_\lambda \in \mathcal{D}^\lambda$ characterizes the profile of the function $\lambda(\mu_\lambda)$ on the boundaries that will represent the internal interfaces of the computational domain. By varying the parameter μ_λ, $\lambda(\mu_\lambda)$ represents a basis function of a local space function W defined along $\hat{\Gamma}_\lambda$. If the space W is well chosen and the RB functions are defined through the solution of these local problems (for a set of parameter values) the trace of the final solution along the internal interfaces will be well interpolated and, because of the linearity of the problem, the reduced solution suitably accurate. The more parametric BCs we use locally, that is the larger is the space W considered, the better the final global solution will be recovered. On the other hand, by introducing local parametric BCs, we increase the complexity of the local problems that lead to a larger parameter vector and consequently more expensive computational times of the RB basis selection.
We now describe the reduced basis spaces $V_i^{RB}, i = 1, \dots, K$, used for defining the space V^N introduced in (4.2). By considering, for $i = 1, \dots, K$, the weak formulation of problems (4.3) and by introducing the corresponding lifting functions $y_\Gamma(\tilde{\boldsymbol{\mu}}^i) \in V_{\hat{\Omega}_i}$ of the Dirichlet BCs, we may decompose $\tilde{y}(\tilde{\boldsymbol{\mu}}^i)$ as $\tilde{y}(\tilde{\boldsymbol{\mu}}^i) = y_o(\tilde{\boldsymbol{\mu}}^i) + y_\Gamma(\tilde{\boldsymbol{\mu}}^i)$, where $y_o(\tilde{\boldsymbol{\mu}}^i) \in V^0_{\hat{\Omega}_i} = \{v \in V_{\hat{\Omega}} \,|\, \mathscr{B}_{Di}(v; \boldsymbol{\mu}_{oi}) = 0 \text{ in } V_{\hat{\Gamma}_i} \cup V_{\hat{\Gamma}_\lambda}\}$. We define the following problems for $i = 1, \dots, K$: for $\tilde{\boldsymbol{\mu}}^i \in \mathcal{D}^i \times \mathcal{D}_o \times \mathcal{D}^\lambda$ find $y(\tilde{\boldsymbol{\mu}}^i) \in V^0_{\hat{\Omega}_i}$ such that

$$a_i(y_o(\tilde{\boldsymbol{\mu}}^i), v; \tilde{\boldsymbol{\mu}}^i) = f_i(v; \tilde{\boldsymbol{\mu}}^i) \quad \forall v \in V^0_{\hat{\Omega}_i} \tag{4.4}$$

with $f_i(v; \tilde{\boldsymbol{\mu}}^i) = \langle \mathscr{F}_i(\boldsymbol{\mu}_o) - \mathcal{A}_i(y_\Gamma(\tilde{\boldsymbol{\mu}}^i); \boldsymbol{\mu}_o), v \rangle_{V',V}$ for $v \in V^0_{\hat{\Omega}_i}$. By using the affine decomposition of the problems (with the empirical interpolation if necessary) we apply the offline stage of the classical reduced basis method for the parametrized problems (4.4) in $\hat{\Omega}_i$ in order to select a set of suitable parameter values $\{\tilde{\boldsymbol{\mu}}^i_j = (\boldsymbol{\mu}^j_{oi}, \mu^j_\lambda), j = 1, \dots, N_i\}$ that defines the local reduced basis functions, in particular we use the greedy algorithm [31].
Finally the local RB spaces are defined as the spaces of the FE solutions $y^{\mathcal{N}}(\tilde{\boldsymbol{\mu}}^i_j)$

of (4.4) corresponding to the parameters values $\tilde{\boldsymbol{\mu}}_j^i, j = 1, \dots, N_i$, selected by the greedy algorithm and properly orthonormalized in order to define the RB spaces:

$$V_i^{RB} = \{\xi_j^i, j = 1, \dots, N_i\}, i = 1, \dots, K. \tag{4.5}$$

In general, for each reference domain, we define an independent parametric problem and we perform independent greedy algorithms (through parallel computations) and we define K reduced basis spaces with their proper number N_i of basis functions. The choice of the space W and the function $\lambda(\mu_\lambda)$ used as local BCs strongly affects the accuracy of the final solution of the problem as well as the computational complexity of the greedy algorithm. The relationship between the choice of W and *(i)* the accuracy, *(ii)* the computational complexity and *(iii)* the performances of the method, will be more clear when addressing specific model problems. We refer to Section 7.4 for a detailed explanation of the local problems together with several possible options for defining W, in particular by using either Lagrange bases or Fourier bases. Related results regarding the offline complexity, the solution accuracy reached during the online stage and the corresponding computational times will be deeply investigated and commented.

4.2 Galerkin projection on the reduced space

The RDF method consists in approximating (3.1) using a subspace generated by the local reduced basis functions, that is: find $y^N(\boldsymbol{\mu}) \in V^N$ such that:

$$a(y^N(\boldsymbol{\mu}), v^N; \boldsymbol{\mu}) = f(v^N; \boldsymbol{\mu}),\ \forall v^N \in V^N. \tag{4.6}$$

The solution of problem (4.6) can be split as:

$$y^N(\boldsymbol{\mu}) = \sum_{r=1}^{R} \sum_{i=1}^{N_{t(r)}} \alpha_i^r \xi_i^{t(r)} + \sum_{k=1}^{\mathcal{N}_\Gamma} \alpha_k^\Gamma \phi_k^\Gamma, \tag{4.7}$$

where ξ_i^j and ϕ_k^Γ are defined in (4.5) and (3.6), respectively and the coefficients α_i^r and α_k^Γ depend on $\boldsymbol{\mu}$. Problem (4.6) can therefore be written as:

$$\begin{cases} \displaystyle\sum_{i=1}^{N_{t(r)}} \alpha_i^r a^r(\xi_j^{t(r)}, \xi_i^{t(r)}; \boldsymbol{\mu}) + \sum_{k=1}^{\mathcal{N}_\Gamma} \alpha_k^\Gamma a^\Gamma(\phi_k^\Gamma, \phi_i^{t(r)}; \boldsymbol{\mu}) = f^r(\xi_i^r; \boldsymbol{\mu}),\ r = 1, \dots, R,\ i = 1, \dots, N_{t(r)}, \\ \displaystyle\sum_{r=1}^{R} \sum_{i=1}^{N_{t(r)}} \alpha_i^r a^r(\xi_i^{t(r)}, \phi_k^\Gamma; \boldsymbol{\mu}) + \sum_{k=1}^{\mathcal{N}_\Gamma} \alpha_k^\Gamma a^\Gamma(\phi_k^\Gamma, \phi_k^\Gamma; \boldsymbol{\mu}) = f^\Gamma(\phi_k^\Gamma; \boldsymbol{\mu}),\ k = 1, \dots, \mathcal{N}_\Gamma \end{cases} \tag{4.8}$$

and, in matrix form, as:

$$\begin{bmatrix} \Phi_1^T \mathbb{A}_1 \Phi_1 & & & \Phi_1^T A_{1\Gamma} \\ & \ddots & & \vdots \\ & & \Phi_R^T A_R \Phi_R & \Phi_R^T \mathbb{A}_{R\Gamma} \\ \mathbb{A}_{1\Gamma}^T \Phi_1 & \cdots & \mathbb{A}_{R\Gamma}^T \Phi_R & \mathbb{A}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{Y_1} \\ \vdots \\ \mathbf{Y_R} \\ \mathbf{Y_\Gamma} \end{bmatrix} = \begin{bmatrix} \Phi_1^T \mathbb{F}_1 \\ \vdots \\ \Phi_R^T \mathbb{F}_R \\ \mathbb{F}_\Gamma \end{bmatrix}, \tag{4.9}$$

where for $r = 1, \dots, R$:

$$[\boldsymbol{Y_r}]_j = \alpha_j^r, j = 1, \dots, N_{t(r)}, \quad \Phi_r^T = [\xi_1^{t(r)}, \dots, \xi_{N_{t(r)}}^{t(r)}], [\boldsymbol{Y_\Gamma}]_j = \alpha_j^\Gamma, j = 1, \dots, \mathcal{N}_\Gamma;$$

we recall that $N_{t(r)}$ is the number of RB basis functions of $V_{t(r)}^{RB}$ and $\mathcal{N}_\Gamma$ the number of Lagrangian bases on $V_\Gamma^{\mathcal{N}}$.

4.3 Online/Offline computational decoupling

As for all RB methods, the Online/Offline computational decoupling is crucial for the effectiveness of the proposed RDF method. All the computations involving the construction of the local reduced basis functions are performed in an offline stage; this corresponds to the classical offline stage of the reduced basis method applied to the local problems. The offline/online computational decomposition also applies to the matrix assembling of the linear system (4.9). Thanks to the affine decomposition of the problem, the parameter independent parts of the required matrices can be assembled starting from the FE ones of the FE linear system (3.8) (of dimension $\mathcal{N}_1 + \cdots + \mathcal{N}_R + \mathcal{N}_\Gamma$) by pre- and post-multiplication by the matrices Φ_r defined offline. Finally, the RDF system (4.9) can be easily assembled online by assembling the precomputed local matrices and the parametric coefficients of the linear and bilinear forms ($\Theta_q^{ra}(\boldsymbol{\mu})$ and $\Theta_q^{rb}(\boldsymbol{\mu})$ defined in (3.4) and (3.5)). The global dimension of the final problem will be $N_{t(1)} + \cdots + N_{t(R)} + \mathcal{N}_\Gamma$ (with $N_{t(i)} \ll \mathcal{N}_i$).

5 *A priori* error estimates

We are interested in providing some *a priori* error estimates between the exact solution $y(\boldsymbol{\mu})$ of problem (2.3) and the RDF one $y^N(\boldsymbol{\mu})$ of problem (4.6). We will suppose that an a-priori inequality:

$$\|y(\boldsymbol{\mu})\|_V \leq C\|f(\cdot,\boldsymbol{\mu})\|_{V'},$$

where C is a proper constant, holds for the solution of problem (2.3).
We start by noting that, thanks to Galerkin projection and Céa's Lemma (see [24]),

$$\|y(\boldsymbol{\mu}) - y^N(\boldsymbol{\mu})\|_V \leq \frac{M}{\alpha} \inf_{w^N(\boldsymbol{\mu})\in V^N} \|y(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_V, \tag{5.1}$$

where M and α are, respectively, the continuity and coercivity constants. Consider now the following inequality:

$$\begin{aligned}\inf_{w^N(\boldsymbol{\mu})\in V^N} \|y(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_V \leq \|y(\boldsymbol{\mu}) - y^{\mathcal{N}}(\boldsymbol{\mu})\|_V + \|y^{\mathcal{N}}(\boldsymbol{\mu}) - p^{\mathcal{N}}(\boldsymbol{\mu})\|_V + \\ + \inf_{w^N(\boldsymbol{\mu})\in V^N} \|p^{\mathcal{N}}(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_V,\end{aligned} \tag{5.2}$$

where:

$$p^{\mathcal{N}}(\boldsymbol{\mu}) = \underset{\substack{q^{\mathcal{N}}(\boldsymbol{\mu})\in V^{\mathcal{N}} \\ q^{\mathcal{N}}(\boldsymbol{\mu})|_\Gamma \in W \\ q^{\mathcal{N}}(\boldsymbol{\mu}) \text{ solves (2.3)}}}{\operatorname{argmin}} \|y^{\mathcal{N}}(\boldsymbol{\mu}) - q^{\mathcal{N}}(\boldsymbol{\mu})\|_V.$$

The first term in estimate (5.2) represents the FE error and decays to zero as h^k, k being the piecewise polynomial degree (see [24]).
The second term in estimate (5.2) represents the difference between the FE solution of problem (3.1) and the FE solution which satisfies further conditions on the internal interfaces $\hat{\Gamma} = \cup_{r=1}^R \hat{\Gamma}_r$: its trace on $\hat{\Gamma}$ belongs to the discrete space W. Decomposing the term on the different subdomains,

$$\|y^{\mathcal{N}}(\boldsymbol{\mu}) - p^{\mathcal{N}}(\boldsymbol{\mu})\|_V = \left(\sum_{r=1}^{\mathrm{R}} \|y^{\mathcal{N}}(\boldsymbol{\mu}) - p^{\mathcal{N}}(\boldsymbol{\mu})\|^2_{H^1(\hat{\Omega}_r)}\right)^{1/2},$$

the functions $y^{\mathcal{N}}(\boldsymbol{\mu})$ and $p^{\mathcal{N}}(\boldsymbol{\mu})$ can be seen in each subdomain Ω_r as solutions of the local problems (4.3) with different Dirichlet BCs, corresponding to their traces along $\hat{\Gamma}_r$ respectively. Thanks to the trace inequality:

$$\|v^{\mathcal{N}}\|_{H^1(\hat{\Omega}_r)} \leq C_2\|v^{\mathcal{N}}\|_{H^{1/2}(\hat{\Gamma}_r)},\ \forall v^{\mathcal{N}} \in H^1(\hat{\Omega}_r),$$

where C_2 is another constant depending on the measure of $\hat{\Omega}_r$, we have:

$$\|y^{\mathcal{N}}(\boldsymbol{\mu}) - p^{\mathcal{N}}(\boldsymbol{\mu})\|_{H^1(\hat{\Omega}_r)} \leq \|y^{\mathcal{N}}(\boldsymbol{\mu}) - d^{\mathcal{N}}(\boldsymbol{\mu})\|_{H^1(\hat{\Omega}_r)} \leq C_2 \|y^{\mathcal{N}}(\boldsymbol{\mu}) - d^{\mathcal{N}}(\boldsymbol{\mu})\|_{H^{1/2}(\hat{\Gamma}_r)},$$

where

$$d^{\mathcal{N}}(\boldsymbol{\mu}) = \underset{\substack{q^{\mathcal{N}}(\boldsymbol{\mu}) \in V^{\mathcal{N}} \\ q^{\mathcal{N}}(\boldsymbol{\mu})|_{\Gamma} \in W \\ q^{\mathcal{N}}(\boldsymbol{\mu}) \text{ solves (2.3)}}}{\operatorname{argmin}} \|y^{\mathcal{N}}(\boldsymbol{\mu}) - q^{\mathcal{N}}(\boldsymbol{\mu})\|_{H^{1/2}(\hat{\Gamma}_r)}.$$

The last term in the previous double inequality depends on how well the trace of a finite element function can be approximated by a linear combination of the bases of W. Particularly this term is zero if these bases coincide with $\phi_i^{\Gamma}|_{\hat{\Gamma}}, i = 1, \ldots, \mathcal{N}_{\Gamma}$, the FE basis functions associated to the nodes along the interfaces. More in general, it tends to zero when the dimension of the space W tends to the number $\mathcal{N}_{\Gamma}$ of the finite element bases along the interfaces.
Finally, the third term in estimate (5.2) measures the difference between the auxiliary function $p^{\mathcal{N}}(\boldsymbol{\mu})$ and its best RDF approximation. Since the RDF space V^N is defined as in (4.2) we have that the inf of $\|p^{\mathcal{N}}(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_V$ over the $w^N(\boldsymbol{\mu})$ is for sure smaller or equal than the value taken by choosing $w^N(\boldsymbol{\mu})$ equal to $p^{\mathcal{N}}(\boldsymbol{\mu})$ on the subdomain $\hat{\Omega}_{\Gamma} \subset \hat{\Omega}$ that represents the support of the FE basis functions of $V_{\Gamma}^{\mathcal{N}}$. Therefore we have:

$$\begin{aligned} \inf_{w^N(\boldsymbol{\mu}) \in V^N} \|p^{\mathcal{N}}(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_V \leq & \inf_{\substack{w^N(\boldsymbol{\mu}) \in V^N \\ w^N_{|\hat{\Omega}_{\Gamma}} = p^{\mathcal{N}}}} \|p^{\mathcal{N}}(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_V \\ = & \inf_{\substack{w^N(\boldsymbol{\mu}) \in V^N \\ w^N_{|\hat{\Omega}_{\Gamma}} = p^{\mathcal{N}}}} \|p^{\mathcal{N}}(\boldsymbol{\mu}) - w^N(\boldsymbol{\mu})\|_{H^1(\hat{\Omega} \setminus \hat{\Omega}_{\Gamma})}, \end{aligned} \tag{5.3}$$

where $\hat{\Omega} \setminus \hat{\Omega}_{\Gamma}$ represents the subdomain of $\hat{\Omega}$ obtained by the union of the supports of the RB basis functions. Hence, this term may be reduced either by lowering the tolerance of the greedy algorithm (which amounts to increase the number of local reduced basis functions) or by enlarging the subdomain $\hat{\Omega}_{\Gamma}$ such that the measure of $\hat{\Omega} \setminus \hat{\Omega}_{\Gamma}$ is reduced.
Summarizing the error introduced by solving the problem with the RDF method is affected by four quantities which represent a trade-off for accuracy and computational costs:

- the maximum diameter h of the elements of the triangulation $\mathcal{T}_h$, that affects the accuracy of the underlying FE solutions,
- the dimension of the space W used for recovering the trace of the solution along the internal interfaces,
- the number N_i of reduced basis on the domain $\hat{\Omega}_i$,
- the number $\mathcal{N}_{\Gamma}$ of nodes where we consider as bases the FE ones.

6 Comparison to existing methods

In this section, we highlight the originality of the method presented in this manuscript with respect to similar methods that share with our RDF method the feature of combining RB and DD: the reduced basis element method (RBEM) [17,16,19,20], the reduced basis hybrid method (RBHM) [11], the static condensation reduced basis element method (SCRBEM) [23] and reduced basis methods used for heterogeneous domain decomposition (RBHDD) [22].
As already mentioned in the Introduction, the common goal of these methods is to extend the range of applicability of the classical RB method and to improve its

flexibility principally in terms of more realistic geometrical configurations defining the computational domain. Indeed a complex geometry (e.g. a network) defined by a large set of parameters may be difficult to address by classical RB methods.
In general, the combination of RB and DD methods offers a more flexible tool for the solution of $\boldsymbol{\mu}$PDEs, nevertheless we have to take into account the further costs and effort of the coupling conditions needed to ensure the continuity of the solutions at the interfaces.
In the following we summarize the basic ideas of the previously listed methods, in order to focus on the novel ingredients proposed in the current RDF method.
The RBEM, just like the RDF method, considers local reduced basis spaces built by suitable local snapshots associated to the solutions of suitable restricted local problems. Several options have been used in literature for the computation of these bases in RBEM, but never parametric BCs (like in the RDF). For instance, one possibility is represented by the definition of local problems in a domain composed by one subdomain (block) and a small part of the contiguous subdomains, along the interfaces of the latter proper fixed BCs are used. In this way, by varying the geometrical parameter values, the restrictions of the local solutions in each block have different interface profiles and not bound to any particular BCs [17].
The final RBEM solution is found through a Galerkin projection on the reduced spaces, moreover this method requires an additional ingredient: the Lagrangian multipliers associated to the jump of the state variables along the internal interfaces. This procedure has some limitations for the Stokes problem, it ensures only the continuity of the velocity of the RBEM solution but not that of the normal stress [16,17].
The RBHM is built upon the idea of the RBEM and it is mainly thought for the Stokes problem [11]. It introduces an alternative way for the computation of the local snapshots in order to deal with solutions that, due to their construction, once coupled they feature continuous stresses along the internal interfaces. The idea is to consider for each reference subdomain three different local Stokes problems, one for each possible position of the corresponding internal interface in the whole computational domain (inflow, middle or outflow). The solution of the local Stokes problems with zero-stress BCs imposed to the boundary that correspond to the internal interface are considered as snapshots of the RBHM. Due to the fact that all these snapshots have zero-stress on the internal interfaces, by construction the final solution will have continuous normal stress. In order to be able to recover also the value of the normal stress (not zero) of the final solution we need to enrich the spaces of the snapshots. This is done by adding to the local spaces a global coarse solution of the original problem, which is then restricted to the subdomains and added to the local spaces. Finally the RBHM solution is provided, as well as the RBEM, through a Galerkin projection on the reduced velocity and pressure spaces and the use of the Lagrangian multipliers associated to the jump of the velocity functions.
Both methods, RBEM and RBHM, provide a suitable solution of the Laplace problem, nevertheless the local bases are not able to provide a complete versatile solution of the whole problem, since they are strongly related to the local problems, that are somehow less flexible than having parametric BCs as in the RDF. The RDF method has been developed in order to improve the performance and versatility of the RBHM, by avoiding the computation of a global coarse solution and the use of Lagrange multipliers, with the aim of extending it for the Stokes problem.
The SCRBEM makes use of the RB method for speeding up the computational time of local functions, called bubble functions, needed to solve a standard static condensation problem along the internal interfaces of the decomposed domain, in particular dealing with the associated Schur complement [7]. Since every subdomain is characterized by a set of parameters, the bubble functions are parametric and are approximated by the RB method used locally in a standard way. However, even if

each RB bubble function is rapidly computed, the number of bubble functions, and consequently the number of RB spaces required, can be very large (especially in a three dimensional context) since it equals the number of degrees of freedom on each internal interface. The idea of the bubble functions can be compared to the RDF concept of the interface basis functions used as local BCs (Lagrangian or Fourier). While in the SCRBEM a RB space is built for every single bubble function, in the RDF every different interface basis function is seen as an interface profile associated to a parameter, and this parameter is included in the set of local subdomain parameters, such that for each subdomain there exists only one RB local space able to cover changes of parameter values and of interface profiles. This strategy allows to reduce the number of local reduced spaces and the global amount of bases required, specially in the case of Fourier basis set.
The RBHDD shares the main idea with the RDF method, however it is aimed at solving heterogeneous domain decomposition problems, e.g. parametric Stokes-Darcy problems. Together with the coupling of heterogeneous equations, this method differs from the RDF for a new option used to define interfaces profiles, coming from the traces of proper global solutions (obtained by varying the parameters values of the problem). As in the RDF method, the set of different interface profiles is represented through an additional parameter used for the selection of the basis, however this option is strongly related to the final geometry of the problem, that in RDF wants to remain completely arbitrary.
By concluding, in the following we list the advantages of the RDF method with respect with the described existing ones.

- Compared with the RBEM and RBHM, the introduction of the additional parametric BCs in the RDF method allows to improve considerably the accuracy of the solution at the cost of a slight increase of the local problem complexity. The latter is due to the use of the additional interface parameter that allows to deal with a much more generic interface profiles, able to capture every behaviour of the final global solution. Moreover, in the RDF method neither a global coarse solution nor the use of the Lagrangian multipliers are required.

- With respect to the SCRBEM, the proposed RDF method represents a good trade-off between CPU time and accuracy thanks to many possible combinations of choices to be adopted for the basic ingredients of the method (number of RB functions, number of local parametric BCs and number of FE bases used). Moreover, while in the SCRBEM the number of RB spaces is considerably large (one for each mesh node along the interface multiplied by the number of reference shapes required), in the RDF method only one RB space is used for each reference shape.

- Finally, compared with RBHDD method, the RDF method has the capability to deal with a much more general combination of computational domains, since the local BCs are generic and not obtained from global solutions of the problem as in the RBHDD. The RDF approach allows to compute offline the basis functions independently from the assembling of the final global computational domain vastly increasing the range of problems solvable online.

7 Numerical results

In this section, we present some numerical results obtained by using the RDF method to approximate a steady conduction Laplace problem represented by a thermal fin with an arbitrary number of subdomains. This problem has previously solved as test problem in the framework of the classical RB method in [32] and of

the RBEM in [20]. We will consider two different main sets of reduced basis functions arising from two different choices of parametric Dirichlet BCs for the local problems: the Lagrange functions and the Fourier functions. Then we compute the corresponding relative errors between the RDF and the FE solution. Finally we carry out several numerical tests obtained by varying the main ingredients of the RDF method: the dimension of the space W used for defining the basis functions along the internal interfaces (i. e. the BCs of the local problems); the number N_i of reduced basis used per subdomain Ω_i; the number $\mathcal{N}_\Gamma$ of FE basis functions used to enrich the RB space. A discussion about the computational time savings with respect to the FE method will also be made.

7.1 Thermal fin problem

We consider the computational domain composed by the union of R non-overlapping subfins Ω_r, $r = 1, \cdots, R$, each constituted by a different material, with its own constant thermal conductivity μ_1^r. Each block can be seen as a deformation of a reference one $\hat{\Omega}_r$, see Figure 3 (left), through a suitable affine map and a geometrical parameter μ_2^r, that defines the length of the fins of the blocks, so that $\boldsymbol{x} = T(\hat{\boldsymbol{x}}, \mu_2^r)$, $\boldsymbol{x} \in \Omega_r, \hat{\boldsymbol{x}} \in \hat{\Omega}_r$.
Figure 3 (right) shows a possible configuration for the computational domain with $R = 4$. In this case there are 4 physical parameters $\boldsymbol{\mu}_1 = [\mu_1^1, \ldots, \mu_1^4]$ and 4 geometrical parameters $\boldsymbol{\mu}_2 = [\mu_2^1, \ldots, \mu_2^4]$, so that $\overline{\Omega}_{\boldsymbol{\mu}_2} = \cup_{r=1}^R \overline{\Omega}_r$. We assume that $\mu_1^i \in [0.1, 10]$ and $\mu_2^i \in [3, 9]$, for $i = 1, \ldots, 4$.

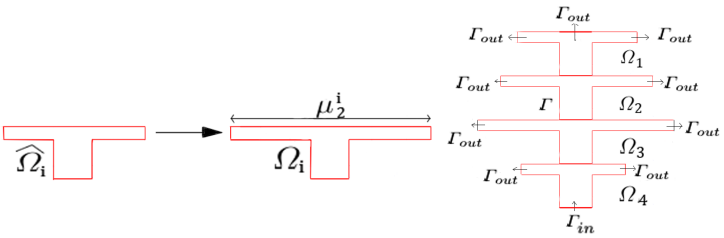

Fig. 3 Scheme for the geometrical transformation of a single block (left) and example of computational domain composed by R=4 blocks (right).

We impose a non-zero uniform temperature at the bottom of the multiblock (Γ_{in}), zero temperature on the vertical surfaces of the spreaders and at the top of the domain (Γ_{out}), and zero heat flux (conservative) on the remaining boundaries of the fins (Γ). We are interested in studying the temperature distribution inside the whole computational domain when the number of blocks and the corresponding values of the parameters vary.
The governing steady conduction problem for the temperature u in the thermal fin is:

$$\begin{cases} \boldsymbol{\mu}_1 \Delta u = 0 \text{ in } \Omega_{\boldsymbol{\mu}_2}, \\ u = 1 \text{ on } \Gamma_{in}, \\ u = 0 \text{ on } \Gamma_{out}, \\ \dfrac{\partial u}{\partial n} = 0 \text{ on } \Gamma, \end{cases} \tag{7.1}$$

where $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_1(\boldsymbol{x}) = \mu_1^i$ if $\boldsymbol{x} \in \Omega_{\mu_2^i}$ $i = 1, \ldots, 4$.

7.2 Geometrical parametrization of the problem

We define here the geometrical parametrization of the elliptic problem (7.1) with affine parametric dependence. Let us introduce a piecewise affine map [31] $T(\hat{\boldsymbol{x}}, \mu_2^r) : \hat{\Omega}_r \to \Omega_r$ such that $\boldsymbol{x} = C_{\mu_2^r}\hat{\boldsymbol{x}} + \boldsymbol{c}_{\mu_2^r}$. We denote with $\overline{\hat{\Omega}} = \cup_{r=1}^{R}\overline{\hat{\Omega}}_r$ the reference domain of $\overline{\Omega} = \cup_{r=1}^{R}\overline{\Omega}_r$. For this problem we have only one reference domain, that we denote here by Λ, and we consider the map $\mathcal{T}_r : \Lambda \to \hat{\Omega}_r$, as shown in Figure 4.

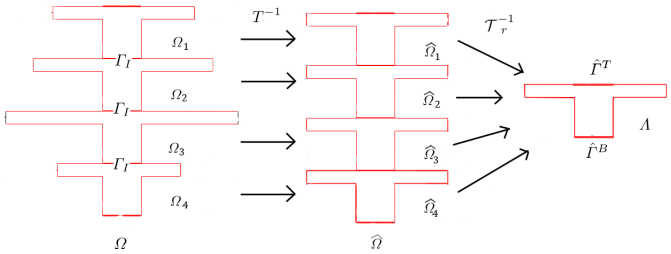

Fig. 4 Computational domain Ω composed by R=4 deformed blocks Ω_r(left) and reference domain $\hat{\Omega}$ composed by $R = 4$ reference blocks $\hat{\Omega}_r$ (center) and reference shape Λ (right).

The weak formulation of problem (7.1) together with its affine decomposition reads: find $\hat{u}_0 \in V_0(\hat{\Omega})$ such that $\forall \hat{v} \in V_0(\hat{\Omega})$:

$$a_{\boldsymbol{\mu}}(\hat{u}_0, \hat{v}) = \sum_{r=1}^{R}\sum_{q=1}^{4} \Theta_q^r(\boldsymbol{\mu}) a_q^r(\hat{u}_0, \hat{v}) = -\sum_{r=1}^{R}\sum_{q=1}^{4} \Psi_q^r(\boldsymbol{\mu}) f_q^r(\hat{v}) = f_{\boldsymbol{\mu}}(\hat{v}). \tag{7.2}$$

7.3 RB and FE basis function description

To define the reduced approximation space, we show graphically in Figure 5 some particular RB functions v_i^r, $r = 1, \ldots, 4$, associated to the solutions of local heat conduction problems on Λ by imposing non-homogeneous constant Dirichlet BCs on $\hat{\Gamma}^T$ and $\hat{\Gamma}^B$ and then mapped onto the subdomains $\hat{\Omega}_r$ as described in Section 4. We defer to Section 7.5 for a detailed numerical description on the way the local BCs are set for the local problems together with the representation of further local solutions featuring other interface profiles.

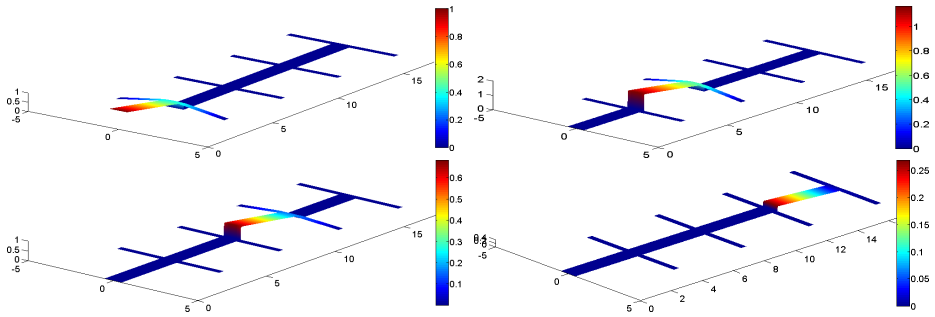

Fig. 5 Example of RB functions associated to each subdomain $\hat{\Omega}_r, r = 1, \ldots, 4$.

The number of reduced basis functions computed is N, and the same set is used on each subdomain. Figure 6 (left) shows the FE functions associated to each node belonging to the subdomain $\hat{\Omega}_\Gamma$. We recall from Section 4, that the FE basis functions used in the RDF method are defined in $\hat{\Omega}_\Gamma$, the union of subdomains of $\hat{\Omega}$

containing the internal interfaces. In order to be able to visualize all the functions involved in the considered FE subspace, we used a very coarse mesh that consists of only 15 interface nodes. These are the FE functions that enrich the final reduced space.

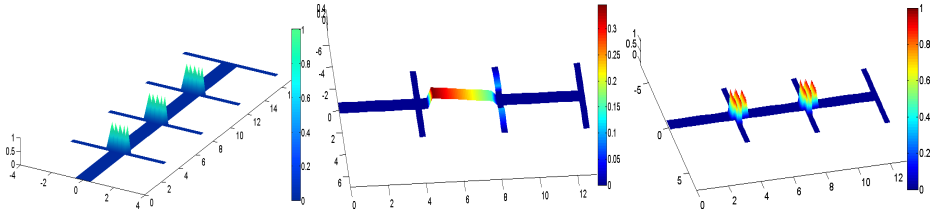

Fig. 6 FE functions associated to every node belonging to the internal interfaces $\hat{\Gamma}_r$ and plotted in the same graph (left).Example of RB function associated to subdomain $\hat{\Omega}_2$ (center).FE functions associated to the nodes belonging to a larger subdomain $\hat{\Omega}_\Gamma$ and plotted in the same graph (right).

The dimension of $\hat{\Omega}_\Gamma$ is arbitrary and can be fixed in the online phase of the method. Let us consider the case where the number of nodes involved in $\hat{\Omega}_\Gamma$ is larger than the previous example. In this case, the support of the RB functions is smaller as it is given by the subdomains $\hat{\Omega}_r \backslash \hat{\Omega}_\Gamma, r = 1, \dots, R$. We show in Figure 6 (center) an example of the function v_i^2 in this different configuration but still obtained as the solution of local heat conduction problems on Λ by imposing non-homogeneous constant Dirichlet BCs on $\hat{\Gamma}^T$ and $\hat{\Gamma}^B$. As before, we show in Figure 6 (right) the plot of all the FE functions associated in this case to a larger subdomain $\hat{\Omega}_\Gamma$ by using a very coarse grid for the scope of the visualization. In order to emphasize that the method is suitable for domains with an arbitrary number of blocks, $\hat{\Omega}$ was first defined by 4 subdomains (Figures 5 and 6, left), then by 3 subdomains (Figures 6, center and right).
In the following section we present some strategies for the computation of the local reduced basis functions, since the idea is to define a basis able to recover the solution for a problem featuring an arbitrary domain configuration.

7.4 Local thermal fin subproblems

We consider the reference shape Λ and we want to define the reduced basis space V_{RB}. We introduce the following parametric local Laplace problem in a generic subdomain Ω_r. We denote with Γ_T and Γ_B the two boundaries of Ω_r that may correspond to the inlet, the outlet or the internal interfaces in the whole domain Ω. We denote with the function $\lambda(\mu_\lambda)$ the Dirichlet BC on $\Gamma_D = \Gamma_T \cup \Gamma_B$. The parameter μ_λ characterizes the profile of the function $\lambda(\mu_\lambda)$ on the boundary Γ_D.

$$\begin{cases} \mu_1^r \Delta u = 0 & \text{in } \Omega_{\mu_2^r}, \\ u = \lambda(\mu_\lambda) & \text{on } \Gamma_D, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \Gamma_r. \end{cases} \tag{7.3}$$

On the weak formulation of problem (7.3), we can apply the same local affine decomposition to recast problem (7.3) on $\hat{\Omega}_r$ and consequently on Λ, that represents a translation of $\hat{\Omega}_r$.
Thanks to the lifting function $u_\Gamma = u_\Gamma(\boldsymbol{\mu}, \mu_\lambda) \in V_h$, $\boldsymbol{\mu} = (\mu_1, \mu_2)$, we define on Λ the finite element approximation $\tilde{u}$ of problem (7.3) such that $\tilde{u} = u + u_\Gamma$ and we

find $u = u(\boldsymbol{\mu}, \mu_\lambda) \in V_h$, such that

$$\sum_{q=1}^{M_a} \Theta_q^a(\boldsymbol{\mu}) a_q(u, v_h) = -\sum_{q=1}^{M_f} \Theta_q^f(\boldsymbol{\mu}, \mu_\lambda) f_q(v_h),\ \forall v_h \in V_h, \tag{7.4}$$

where

$$V_h = X_h^1 \equiv \{v_h \in C^0(\Lambda) : v_h|_K \in \mathbb{P}_1\ \ \forall K \in \mathcal{T}_h\} \cap H^1_{0\Gamma_D}.$$

We apply now the greedy algorithm of the classical reduced basis method for the parametrized problem (7.4) in Λ in order to select a set of suitable parameter values $\{(\boldsymbol{\mu}_i, \mu_\lambda^i), i = 1, \dots N\}$ that will identify the local reduced basis functions.

7.5 The local BCs and the local greedy algorithms

We deal with two possible options when defining the set of local BCs of problem (7.4) defined in Λ. In particular, we consider, first, that $\lambda(\mu_\lambda^i), i = 1, \dots, M$, are the M Lagrangian piecewise functions defined on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ (see Figure 4). Then, we explore the possibility to define $\lambda(\mu_\lambda^i), i = 1, \dots, M$ as M different Fourier functions on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$. The next subsections describe in detail the role of both sets of bases in the offline stage of the method.

7.5.1 The Lagrangian piecewise interface functions

Our first options are $W = W_H$ represented by different spaces of Lagrangian piecewise basis functions defined on the interfaces $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$. Every space W_H is characterized by Lagrangian bases with a different support dimension, denoted by H. The μ_λ is defined in a discrete domain of integer numbers and the function $\lambda(\mu_\lambda)$ indicates the Lagrangian function associated to the node of the interface numerated by the number μ_λ.
We suppose that $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ have the same dimension and denote by $\mathcal{N}_\Gamma$ the number of nodes on these boundaries. We assume that the nodes of the mesh belonging to $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ are equipartitioned and h is the distance between two adjacent nodes. For every space W_H the support of $\lambda(\mu_\lambda)$ can be smaller or equal to the interface length, $H \leq (\mathcal{N}_\Gamma - 1)h$, so that the number of the Lagrangian functions defining W_H is equal or smaller than the number of nodes $\mathcal{N}_\Gamma$. We associate a value of $\mu_\lambda \in \{1, 2, \dots, 2(\frac{(\mathcal{N}_\Gamma - 1)h}{H} + 1)\}$ to identify each Lagrangian basis functions.
We assume at first, that the support coincides with the interface $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$, so that $H = (\mathcal{N}_\Gamma - 1)h$ (in these numerical tests we have $(N_\Gamma - 1)h = 1$) and $\mu_\lambda \in \{1, \dots, 4\}$ (we recall that we have parametric BCs on two interfaces). The functions $\lambda(1)$ and $\lambda(2)$ are defined as the Lagrangian piecewise functions associated to the two extreme nodes of $\hat{\Gamma}_T$ and equal to zero on $\hat{\Gamma}_B$ and $\lambda(3)$ and $\lambda(4)$ are the Lagrangian piecewise functions associated to the two extreme nodes of $\hat{\Gamma}_B$ and equal to zero on $\hat{\Gamma}_T$, see Figure 7. Figure 8 shows, for an arbitrary value of the parameter $\boldsymbol{\mu}$, the solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4) by fixing $H = (\mathcal{N}_\Gamma - 1)h$ and in correspondence of the "Lagrange BCs" $\lambda(\mu_\lambda)$ with $\mu_\lambda = 1, 2, 3, 4$.
By varying the value of H, we have different sets of Lagrangian functions $\lambda(\mu_\lambda)$ to be used as BCs and consequently we have different sets of possible values of μ_λ. By recalling that in the local parametric problem (7.4) the parameters are $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and μ_λ, we observe that the parameter μ_1 (for the specific considered problem) does not influence the solution $u(\boldsymbol{\mu}, \mu_\lambda)$ of problem (7.4), so that we can set $\mu_1 = 1$ and consider only two parameters involved in the greedy algorithm: μ_2 and μ_λ. Figure 9 shows the distribution of the parameter values μ_2^i and μ_λ^i selected by the greedy algorithm, with $H = 1$.

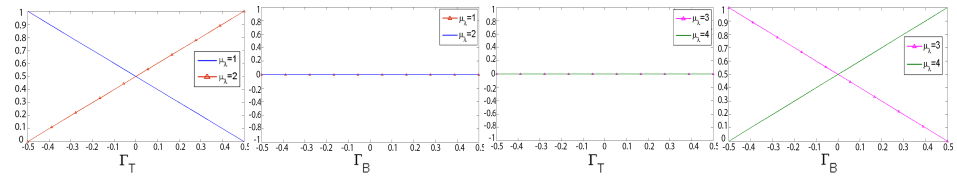

Fig. 7 Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for H=1 and $\mu_\lambda = 1, 2, 3, 4$.

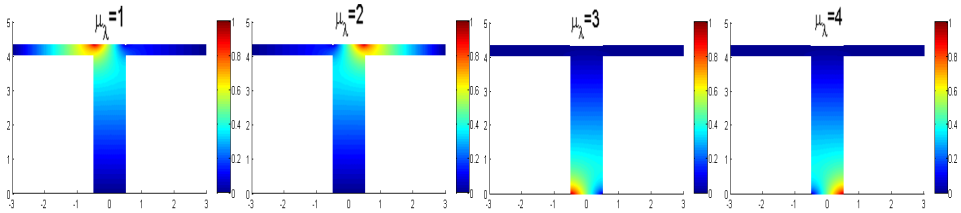

Fig. 8 Solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4) $\mu_\lambda = 1, 2, 3, 4, H = 1$.

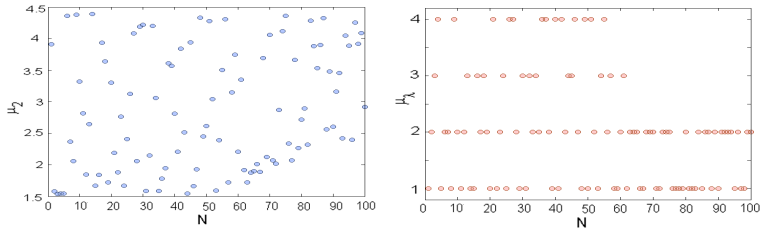

Fig. 9 Parameter values of the selected μ_2^i (left) and μ_λ^i (right), $i = 1, \ldots, N$ during the greedy algorithm for $H = 1$.

In general, we can reduce the value of H in order to consider a larger space of Lagrangian functions along the interface that may better approximate the final solution of the problem. We consider a second example with $H = 1/2$, in this case $\mu_\lambda \in \{1, \ldots, 6\}$ and the profiles of the function λ dependent on μ_λ are shown in Figure 10 (here we report only the interface in which λ is non-zero). Figure 11 shows, for fixed parameter $\boldsymbol{\mu}$, the solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4) associated to the values $\mu_\lambda = 1, \ldots, 6$ for $H = 1/2$. Figure 12 (left) shows the distribution of the parameter values μ_λ^i selected during the greedy algorithm, with $H = 1/2$. The distribution of the parameter values μ_2^i is similar to the previous case.

We report in detail a last case in which $H = 1/4$ and $\mu_\lambda \in \{1, \ldots, 10\}$ and the displacements of the function λ in dependence of μ_λ are shown in the Figure 13. Figure 14 shows, for fixed parameter $\boldsymbol{\mu}$, the solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4) associated to the values $\mu_\lambda = 1, \ldots, 10$ for $H = 1/4$.

Figure 12 (right) shows the distribution of the parameter values μ_λ^i selected during the greedy algorithm, $i = 1, \ldots, N$. Also in this case the distribution of the parameter values μ_2^i is similar to the previous cases.

We note that in the three cases, the selected values of μ_λ have a bigger concentration in the interval $[1, \frac{\max(\mu_\lambda)}{2}]$, for which $\lambda(\mu_\lambda)$ is not zero along $\hat{\Gamma}_T$, that is the boundary close to the geometrical deformation (see the geometrical representation of Λ in Figure 4). In particular, in this set of functions, the number of selected bases with BCs equal to the Lagrangian basis corresponding to the extreme nodes of $\hat{\Gamma}_T$ is larger with respect to the others. These solutions are obviously more affected to the geometrical deformation of the domain, so that the greedy algorithm needs to select more basis functions referred to these value of μ_λ in correspondence of different values of μ_2.

We can decrease the value of H progressively until we consider $H = h$, in this case we take one Lagrangian basis for each node on the interface and $\mu_\lambda \in \{1, \ldots, 2\mathcal{N}_\Gamma\}$.

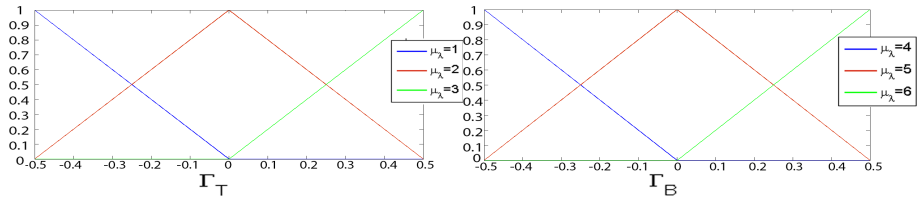

Fig. 10 Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for H=1/2 and $\mu_\lambda = 1, 2, 3, 4, 5, 6$.

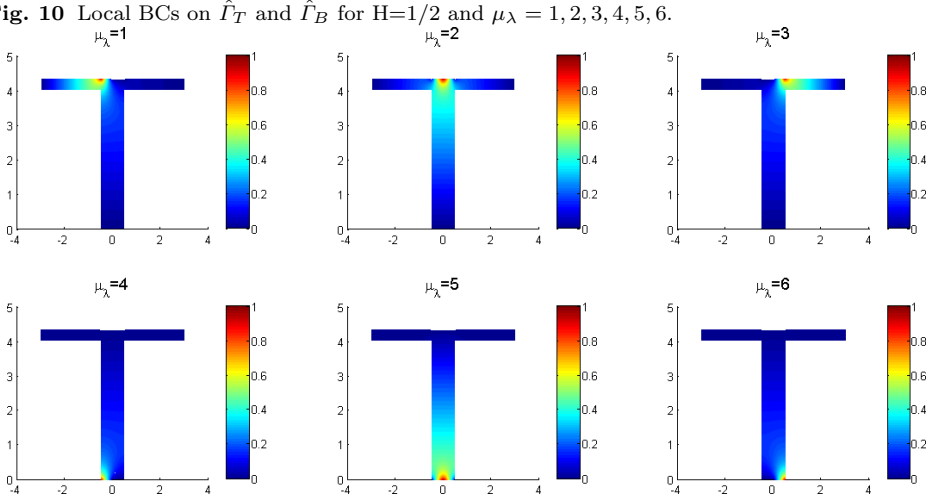

Fig. 11 Solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4), $\mu_\lambda = 1, 2, 3, 4, 5, 6$, $H = 1/2$.

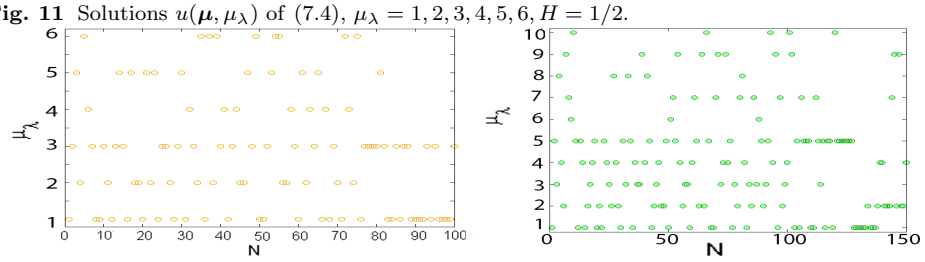

Fig. 12 Parameter values of the selected μ_λ^i, $i = 1, \ldots, N$ during the greedy algorithm for $H = 1/2$ (left) and $H = 1/4$ (right).

In theory, this is the case in which we can approximate in the best way the final solution, but we have a large range of the parameter values of μ_λ, this means that we increase both the computational complexity of the greedy algorithm and the number of the reduced basis functions necessary to reach a suitable accuracy. For these numerical tests we set 65 FE nodes belonging to the interface $\hat{\Gamma}_T$ as well as to the interface $\hat{\Gamma}_B$, so that if $H = h$ the maximum number of possible Lagrangian functions is 130. Figure 15 shows the average of the relative errors between the RB and FE solutions obtained by solving the local problem (7.4) and by using different values of H for the setting of the BCs. The errors are reported by varying the number of reduced basis (computed by the greedy algorithm). We observe that by enlarging the range of possible values of μ_λ, the number of reduced basis functions needed to reach a suitable accuracy become bigger. For instance, we note that in the case of $max\mu_\lambda = 130$, i.e. H=h (the case of one Lagrange function for each interface nodes), the error curve drastically decreases when $N > 130$. This is due to the fact that a basis corresponding to a particular Lagrangian function can not be well approximated by a combination of the solutions corresponding to the other independent Lagrangian basis (centered on the other different nodes of the interfaces). Then the minimum number of local reduced basis functions corresponds to the number of local Lagrangian basis functions used as BC.

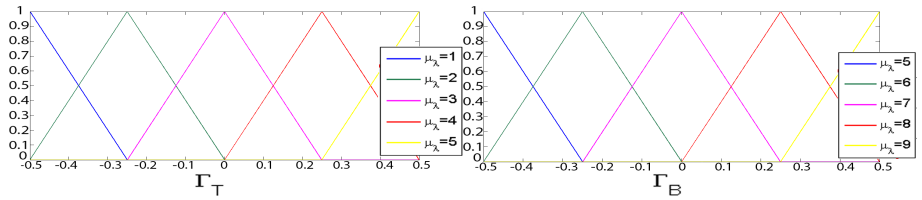

Fig. 13 Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for H=1/4 and $\mu_\lambda = 1, \ldots, 10$.

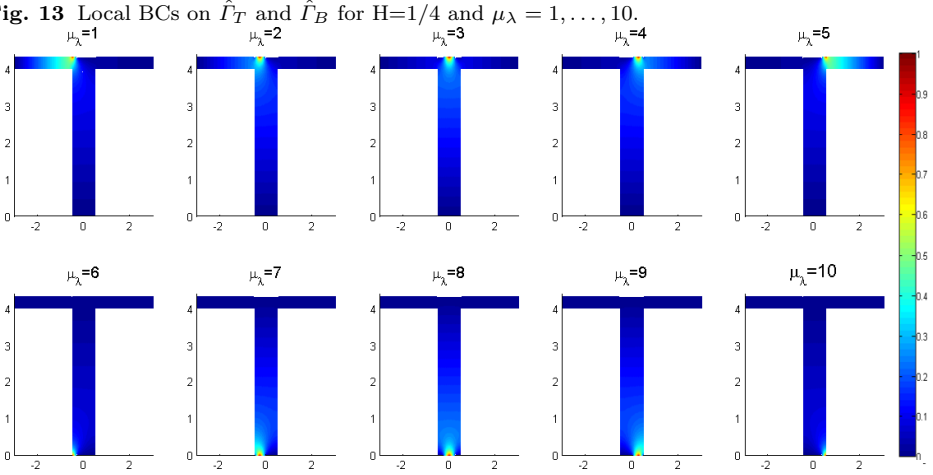

Fig. 14 Solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4), $\mu_\lambda = 1, 2, \ldots, 10, H = 1/4$.

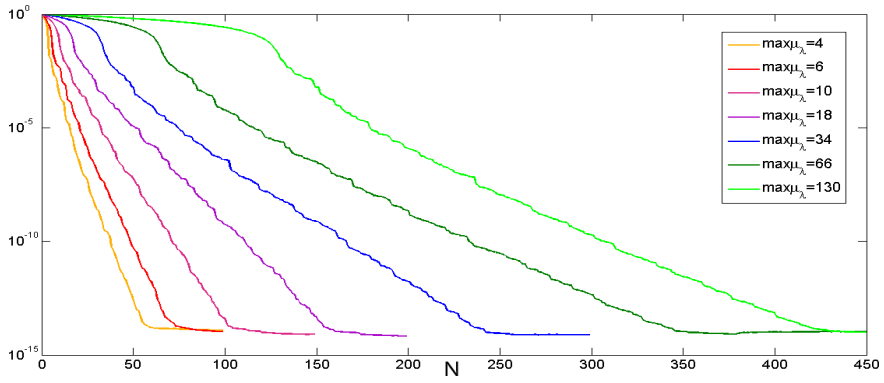

Fig. 15 Average of relative error between the RB local solution of (7.4) and the FE one on a sample test of 1000 parameter values.

7.5.2 The Fourier interface functions

With the aim of improving the offline stage, we explore a second option: the use of $W = W_M$ as Fourier functions spaces associated to the local parametric BCs for the problem (7.4). Also in this case we have to fix the number, denoted by M, of possible functions to consider on $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$. The function $\lambda(\mu_\lambda)$, when it is different from zero, by varying the value of the parameter μ_λ, represents a Fourier basis function defined on $\hat{\Gamma}_T$ or $\hat{\Gamma}_B$. In particular, we consider the following set of M different local BCs, by defining λ as follows: if $\mu_\lambda = 1, \ldots, M/2$

$$\lambda(\mu_\lambda) = \begin{cases} \cos(m(\mu_\lambda)\pi \boldsymbol{x}) & \boldsymbol{x} \in \hat{\Gamma}_T, m(\mu_\lambda) = \mu_\lambda - 1, \\ 0 & \boldsymbol{x} \in \hat{\Gamma}_B, \end{cases} \tag{7.5}$$

and if $\mu_\lambda = M/2+1,\dots,M$

$$\lambda(\mu_\lambda) = \begin{cases} 0 & \boldsymbol{x} \in \hat{\Gamma}_T \\ \cos(m(\mu_\lambda)\pi \boldsymbol{x}) & \boldsymbol{x} \in \hat{\Gamma}_B, m(\mu_\lambda) = \mu_\lambda - M/2 - 1, \end{cases} \tag{7.6}$$

Figure 16 shows the function λ by varying the values of $\mu_\lambda \in \{1,\dots,6\}$.

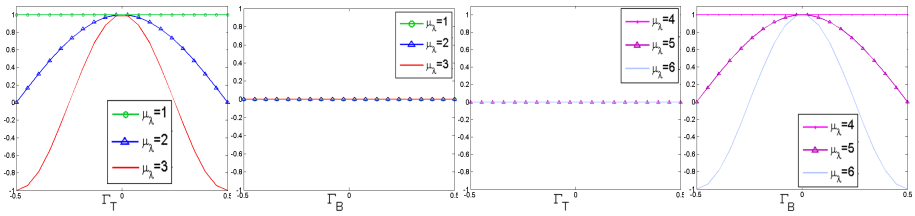

Fig. 16 Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for M=6 and $\mu_\lambda = 1,\dots,6$.

Figure 17 shows, for a fixed parameter $\boldsymbol{\mu}$, the solutions $u(\boldsymbol{\mu},\mu_\lambda)$ of (7.4) associated to the value $M=6$ and $\mu_\lambda = 1,\dots,6$.

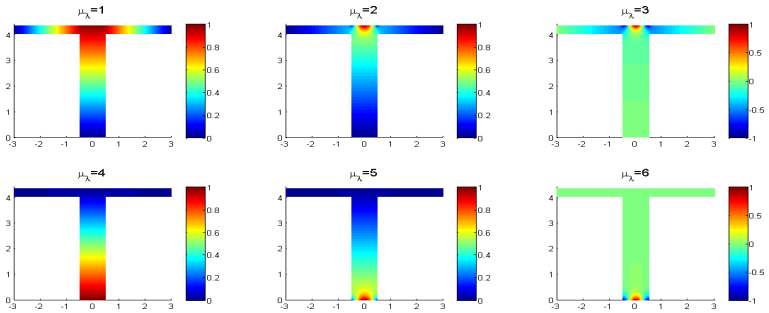

Fig. 17 Solutions $u(\boldsymbol{\mu},\mu_\lambda)$ of (7.4) in correspondence of $\mu_\lambda = 1,\dots,6$.

Figure 18 (left) shows the distribution of the parameter values μ_λ^i selected during the greedy algorithm with $M=6$.

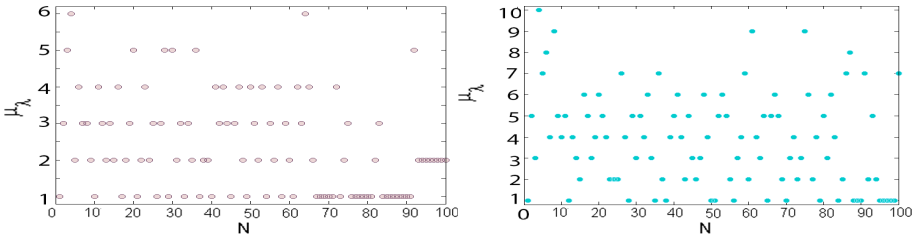

Fig. 18 Parameter values of the selected μ_λ^i, $i = 1,\dots,100$ during the greedy algorithm when $M = 6$(left) and $M = 10$ (right).

By increasing the quantity M, we add more functions to the previous set of BCs, as shown in Figure 19 by the plot of the function $\lambda(\boldsymbol{x},\mu_\lambda)$, when it is non-zero, for $M=10$ and consequently $\mu_\lambda = 1,\dots,M/2$. By following the same descriptions of the previous case, we show, for $M=10$, both the set of basis functions and the distributions of the parameter values μ_λ^i, $i=1,\dots,N$, in Figures 20 and 18 (right).

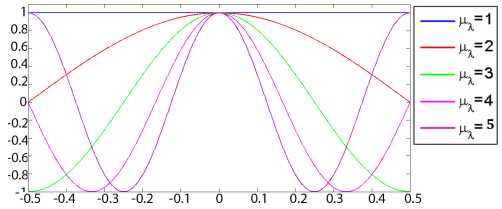

Fig. 19 Local BCs on $\hat{\Gamma}_T$ and $\hat{\Gamma}_B$ for M=10 and $\mu_\lambda = 1, \ldots, 5$.

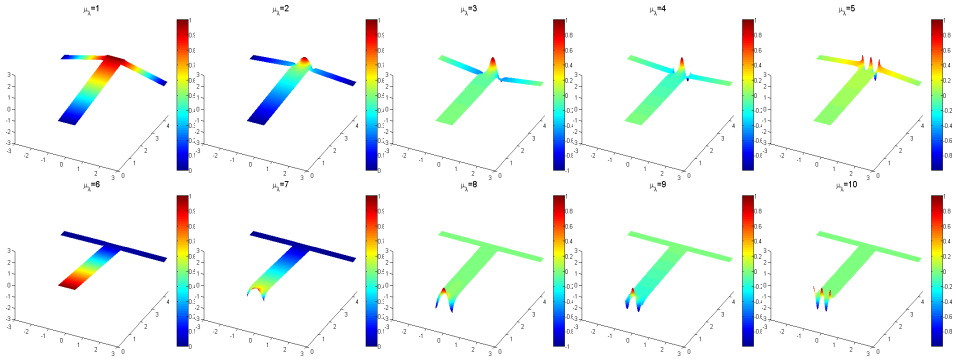

Fig. 20 Solutions $u(\boldsymbol{\mu}, \mu_\lambda)$ of (7.4) in correspondence of $\mu_\lambda = 1, \ldots, 10$.

Figure 21 shows the average of the relative errors between the RB local solution and the FE one obtained by solving the local problem (7.4) and by using different boundary "Fourier" conditions by varying the number of reduced basis functions (computed by the greedy algorithm). The same considerations of the Lagrangian case about the offline computation complexity hold this time. Due to the fact that the support of the Fourier bases is larger than that of the Lagrange bases, the number of reduced basis functions that we need in order to reach a certain accuracy is considerably smaller than in the Lagrange case.

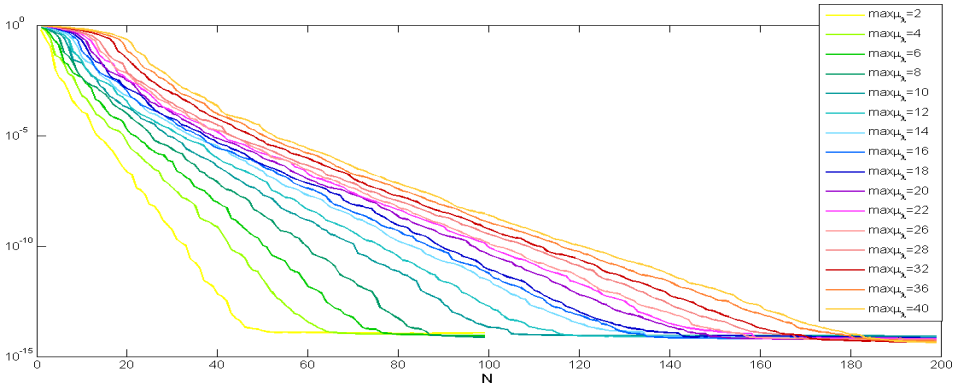

Fig. 21 Average of relative errors between the RB local solution of (7.4) and the FE one on a samples set of 1000 parameter values.

7.6 Global solution and error analysis

Several numerical tests have been carried out in order to test all the different options introduced in the previous section. In the following subsections we report how the choice of $\lambda(\mu_\lambda)$ (and that of M or H) affects the accuracy of the global approximated

solution. The results will be shown in relation to different combinations of *(i)* the set of local BCs, *(ii)* the number of nodes involved in the FE region and *(iii)* the number of reduced basis functions used. Moreover, some considerations regarding the computational saving will follow.

7.6.1 Varying the set of local BCs

We have performed, in different and independent offline stages, several greedy algorithms in order to build many reduced basis spaces. As already mentioned, each reduced basis space represents a possible choice for applying the described RDF method. In particular, we have always used a fine mesh of 37505 nodes and 65 local interface nodes and we have considered 7 options for the Lagrangian piecewise functions, by setting $H = 1, 1/2, 1/4, 1/16, 1/32, 1/64$ and 20 options of the Fourier fuctions, by setting $M = 4, 6, 8, \ldots, 40$. Then we have applied the RDF method by using the 26 different local reduced basis spaces on the 3 blocks network configuration to approximate the Laplace problem (7.1) by using 100 randomly selected parameter samples.
Figure 22 (left) shows the average relative errors between the RDF solution and the finite element solution, depending on the number of local BCs used (dependent on H) and the number of reduced basis functions used in each block (N), by using the Lagrangian piecewise functions set.
The same comparison has been performed with the Fourier set of BCs. Precisely, we have built in different and independent offline stages 19 reduced basis spaces computed through the greedy algorithm and each time by imposing a different value of $M = \{4, \ldots, 40\}$. Figure 22 (right) shows the average relative errors between the RDF solution and the FE solution obtained again as function of the number of local BCs (dependent on M) and the number of reduced basis functions (N).

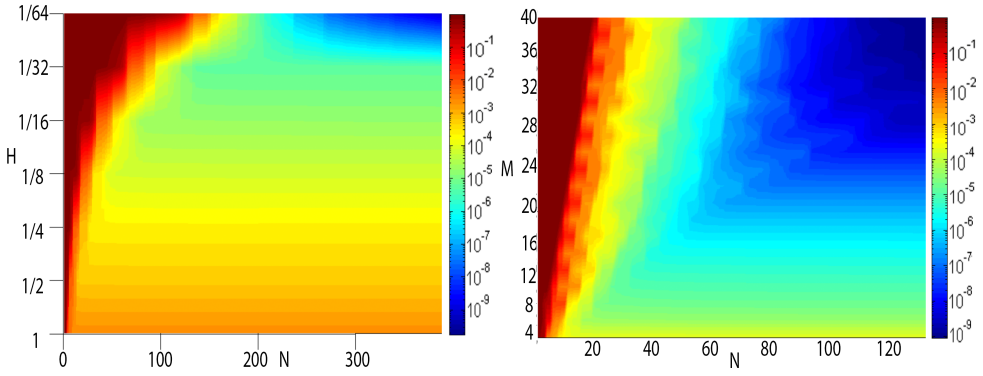

Fig. 22 Average relative error between the RDF the FE solutions on a samples set of 100 parameter values by using different Lagrangian BCs (left) and different Fourier BCs (right).

We note that, in both cases, for a fixed number of local BCs, the error decreases by increasing the number of local RB functions and reaches a plateau that depends on the number M or H characterizing the choice of the RB space. On the other hand, for a fixed number of local RB functions the error diminishes by increasing the number of local BCs, however, if the number of local RB functions is too small, the error increases as the RB space is not large enough to accurately approximate solutions corresponding to all the different local BCs. Here the domain $\hat{\Omega}_\Gamma$ involves only the nodes on the internal interfaces. Moreover, we observe that, at least for the problem considered, the Fourier basis used to recover the trace of the solution along the internal interfaces seems a more practical and effective choice for the RDF method.

This choice allows to reach a reasonable level of accuracy by using a smaller number of RB functions if compared to the choice of the Lagrangian BCs. For instance, in order to achieve an accuracy of the order 10^{-6}, by using the Fourier option we have to choose M=16 and N=50, while by using the Lagrange option this requires more than 200 RB functions and, at least, H=1/32. Figure 23 shows the RDF solution obtained by using Fourier BCs, $M = 6$ and $N = 40$ corresponding to the parameter values $\boldsymbol{\mu}_1 = (4.3725, 9.7089), \boldsymbol{\mu}_2 = (4.3947, 9.5760), \boldsymbol{\mu}_3 = (2.0272, 4.9052)$.

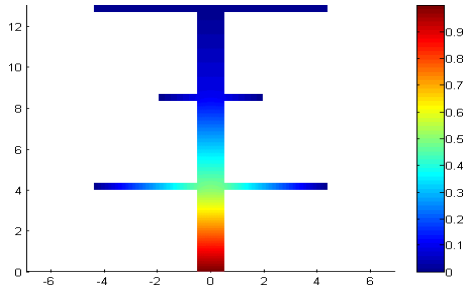

Fig. 23 RDF Solution for a three blocks configuration corresponding to $\boldsymbol{\mu}_1 = (4.3725, 9.7089), \boldsymbol{\mu}_2 = (4.3947, 9.5760), \boldsymbol{\mu}_3 = (2.0272, 4.9052)$, obtained by using Fourier BCs, $M = 6$, N=40.

7.6.2 Extending the finite element region

The second battery of tests that we have performed is concerned with a larger definition of the domain $\hat{\Omega}_\Gamma$. Figure 24 (left) reports the average relative error between the RDF solution and the FE one by considering the FE region $\hat{\Omega}_\Gamma$ comprising about 4.32% nodes of the total number of mesh nodes of the domain. Error was computed by using the spaces of Fourier BCs and by varying the number of RB functions and the local BCs.

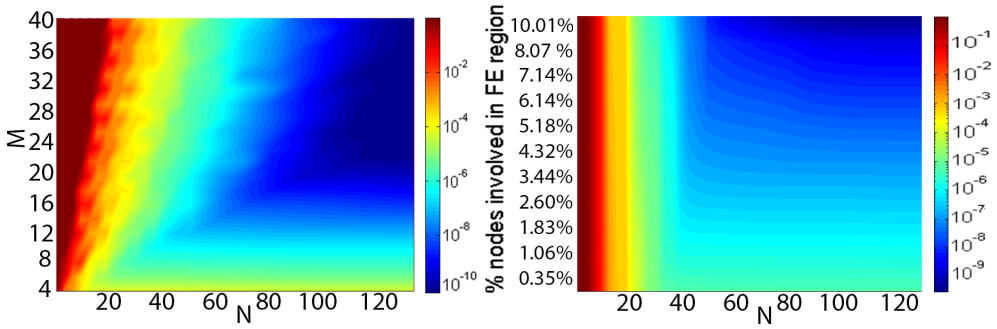

Fig. 24 Average relative error between the RDF and FE solutions on a sample set of 100 parameter values by using local Fourier BCs, *(i)* $\hat{\Omega}_\Gamma$ composed of 4.32% nodes of the total number of mesh nodes and by varying the number of RB functions N and the number of boundary functions M (left) and *(ii)* by using $M = 10$ and by varying N and the percentage of nodes involved in the FE regions.

We observe that the error plot has a similar pattern than the one obtained with FE nodes considered only on the internal interfaces, with the difference that here the error values have a lower order of magnitude. For instance, this time, in order to reach an accuracy error of 10^{-6} it is necessary to choose M=5 and N=25. In the next section we discuss how the addition of further FE functions affects the

computational time. In order to clearly visualize (still in the case of Fourier option) the decreasing of the error by enlarging the subdomain $\hat{\Omega}_\Gamma$ (i.e. increasing the number of FE functions), we report in Figure 25 (left) the relative error between the RDF solution and the FE one by increasing the FE functions and by varying the number of local BCs. This result shows that the enlargement of $\hat{\Omega}_\Gamma$ by adding nodes proximal to the internal interfaces at fixed number of local RB functions is particularly effective for a specific range of the number of local BCs ($5 \leq M \leq 22$).

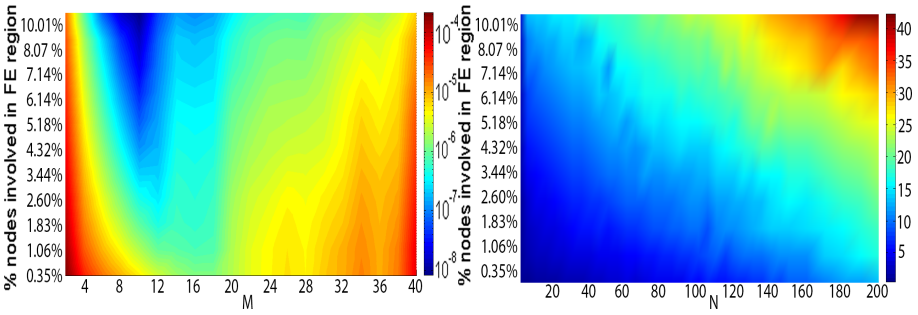

Fig. 25 Average relative error between the RDF and FE solutions by using local Fourier BCs, N=50 RB functions and by varying the number of boundary functions and the number of nodes in $\hat{\Omega}_\Gamma$ (left). Average computational time of the RDF method as percentage of the FE method applied to the global problem by varying the number of RB functions used and the number of nodes in $\hat{\Omega}_\Gamma$ (right). For both plots we used a sample set of 100 parameter values.

Figure 24 (right) shows the relative error between the RDF solution and the FE solution by increasing both the number of RB and FE functions (by using the Fourier option and M=10). Therefore, by increasing the number of FE functions is a viable strategy to decrease the error. When FE nodes are 100% of the total (limit case), RDF reduces to the FE method on the global domain. The pictures relative to different numbers of local BCs present similar patterns.
Figure 25 (right) shows the computational time of the RDF method as percentage of the FE method applied to the global problem. The computational time increases when increasing either the number of local RB functions and the number of FE functions.
By a suitable combination of number of FE nodes and local RB functions we can achieve a good trade-off between computational time saving and accuracy.

7.6.3 Increasing the number of blocks

The last train of tests has been performed by increasing the number of blocks on the computational domain. The number of FE bases used for enriching the RB space equals the number of nodes belonging to the internal interfaces. Figure 26 displays the relative errors between the RDF solution and the FE solution by using the Fourier local BCs with values $M = 6$, $M = 20$ and $M = 36$, the errors are plotted versus the number R of computational domains ($R = 3, \ldots, 50$) and the number N of reduced basis functions.
To start with, we note that the order of magnitude of the error does not depend on the number of blocks composing the domain. Moreover, the method can be tuned a-priori according to the desired solution accuracy and computational time. For instance, a very low approximation error (e.g 10^{-8}) can be achieved by using a large set of BCs (M=32), but a large number of RB functions will be required ($N \approx 90$). This level of accuracy can not be reached with a smaller set of BCs (i.e. M=6), however if the required accuracy is larger, say about 10^{-5}, it is more convenient to use a low number of BCs, since the number of the required RB functions will be

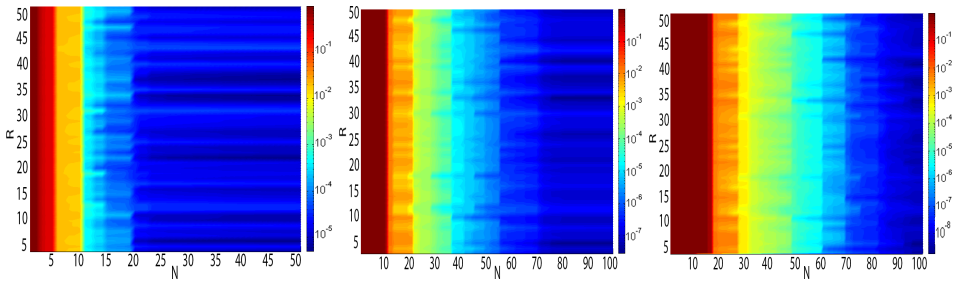

Fig. 26 Relative error between the RDF solution and the FE solution by varying the number of blocks (R) on the domain and the number of RB functions (N) with $M = 6$ (left), $M = 20$ (central), $M = 32$ (right).

low as well. We need 25 RB functions if $M = 6$, 40 RB functions if $M = 20$, more than 50 RB functions if $M = 32$.
Finally, regarding the computational times, we observe that the performance of the RDF method compared with the FE one is still attractive even if the number of needed RB functions is large and if several blocks compose the computational domain. Figure 27 (right) shows the percentage of the computational time of the RDF time with respect to the FE method by varying both the number of blocks (R) and the number of RB functions (N). Figure 27 (left) shows the computational time (in seconds) of both methods. We remark that the type of RB functions (as well as the options of local BCs) does not affect the computational time.

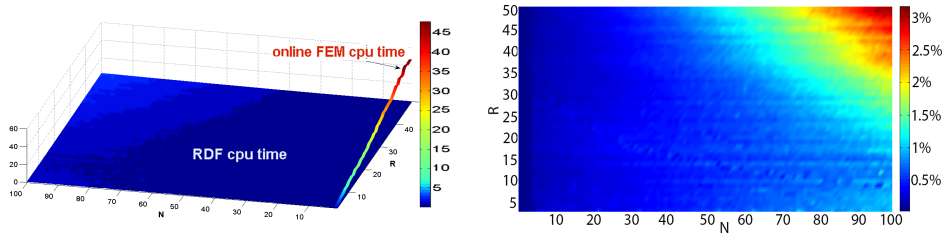

Fig. 27 Computational time of the RDF method and the FE method in seconds (left) and RDF cpu times as percentage of the FE one (right) applied to an increasing number of blocks on the domain and varying the number of reduced basis functions.

8 Concluding remarks on the numerical results

In this section we revisit the results presented in Section 7.6 in the light of the *a priori* error estimate of Section 5, stating that the RDF error depends on the maximum diameter h of the elements of the triangulation $\mathcal{T}_h$, the number $\mathcal{N}_\Gamma$ of nodes involved in $\hat{\Omega}_\Gamma$, the number N of reduced basis functions on the subdomains $\hat{\Omega}_i$, and the function space W used for the local BCs and its dimension. We focus on the last three factors as they affect the error between the RDF solution and the FE one.
We start considering the number and the type of local BCs determining the space W. We used both piecewise linear Lagrangian bases and Fourier hierarchical bases. The convergence of the local greedy algorithm depends on the number of local BCs but not on the type of BCs (Fourier or Lagrangian). Particularly, the number of local reduced basis functions N (necessary to achieve a prefixed tolerance) sensibly increases if we increase the number of local BCs. As a matter of fact, by increasing

the number of possible parametric BC profiles, the range of the parameter μ_λ increases as well, hence we need a larger number of RB functions. However, the kind of BCs chosen may affect the global RDF error. In this respect, the Fourier basis functions are more efficient than the piecewise Lagrangian one.
From the computational point of view, treating a larger number of RB functions requires a larger computational time during both the online and offline phases. In general, by increasing the number of RB functions the error of the RDF method decreases but the *offline* computational time (required for their selections) and the *online* computational time (the size of the local RB submatrices leading to a larger algebraic linear system) increase. Moreover we note that the RB submatrices are full. The dimension of the FE region also affects both the accuracy and the *online* computational time of the method. In general, by increasing this dimension, the error of the RDF method decreases, but the *online* computational time increases (meaning that we add to final linear system sparse matrices of dimensions equal to the number of nodes in the FE region).
The results shown in Section 7.6 suggest however to take some cautions when tuning the method. For example, it may be inefficient to increase the number of local reduced basis N because, at some point, the second term in our *a priori* error estimate may be dominating. In this circumstance, increasing N only slows down the *online* computational time without yielding any substantial benefits from the point of view of the RDF accuracy. On the other side, increasing the number of local boundary conditions without increasing N may lead to the opposite situation in which the third term of the error estimate is dominating hence not producing any improvement on the accuracy of the RDF solution but possibly reducing it due to a worse approximation of the local solutions.
The challenge of the method is to find the best combination of parameters (i. e. a reasonable trade-off) to reach a desired accuracy by minimizing the *online* and *offline* computational times.

9 Conclusions and perspectives

In this work we introduced a new method (RDF) that efficiently combines reduced basis method, domain decomposition techniques and finite element method. The method is particularly suitable for parametrized PDEs in networks of repetitive blocks and has been successfully applied to second order elliptic problems. By exploiting the repetitiveness of few reference blocks, the method reduces drastically the *offline* computational time with respect to global reduced basis approach, especially for large networks. In fact the reduced basis functions are pre-built locally on the few reference blocks (using specific BCs) whose space of parameters is much lower than that of the global one. Moreover, finite element basis functions are used to enrich the reduced space. The method is flexible in the sense that by modifying the dimension of the local reduced bases, the number of finite element functions involved, the space used for the local BCs and its dimension, it is possible to tune and balance the accuracy and computational time. In general, by increasing the value of these parameters the accuracy of the numerical RDF solution is increased, the drawback, however, is that also both *online* and *offline* computational times may be affected.
Future developments may include further investigation of the theoretical framework, as *a posteriori* error estimates, extension to Stokes problems and testing of other, possibly more effective, basis functions for the trace of the solution on the internal interfaces.

Acknowledgements

This work has been supported by the Swiss National Science Foundation under the Project 122136, by ERC-Mathcard Project (ERC-2008-AdG-2270058), by the EU FP7 Marie Curie Zukunftskolleg Incoming Fellowship Programme, University of Konstanz (grant no. 291784), by NOFYSAS Excellence Grant, SISSA. INdAM-GNCS is kindly acknowledged.

References

1. Antonietti, P.F., Pacciarini, P., Quarteroni, A.: A discontinuous Galerkin reduced basis element method for elliptic problems. Submitted (2014)
2. Barrault, M., Maday, Y., Nguyen, N., Patera, A.: 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)
3. Ciarlet, P.G.: The Finite Element Method for Elliptic Problems. North Holland (1978)
4. Deparis, S.: Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. SIAM J. Numer. Anal. **46**(4), 2039–2067 (2008)
5. Deparis, S., Løvgren, E.: Stabilized reduced basis approximation of incompressible three-dimensional Navier-Stokes equations in parametrized deformed domains. Journal of Scientific Computing pp. 1–15 (2011)
6. Deparis, S., Rozza, G.: Reduced basis method for multi-parameter-dependent steady Navier-Stokes equations: Applications to natural convection in a cavity. Journal of Computational Physics **228**(12), 4359–4378 (2009)
7. 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**(1), 3 (2014)
8. Eftang, J.L., Patera, A.T.: Port reduction in parametrized component static condensation: approximation and a posteriori error estimation. International Journal for Numerical Methods in Engineering **96**(5), 269–302 (2013)
9. Huynh, D., Knezevic, D., Patera, A.: A static condensation reduced basis element method: Complex problems. Computer Methods in Applied Mechanics and Engineering **259**(0), 197 – 216 (2013)
10. Iapichino, L.: Reduced basis methods for the solution of parametrized PDEs in repetitive and complex networks with application to CFD. Ph.D. thesis, N. 5529, École Polytechnique Fédérale de Lausanne (2012)
11. 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. Engrg. **221–222**, 63–82 (2012)
12. Jäggli, C., Iapichino, L., Rozza, G.: An improvement on geometrical parameterizations by transfinite maps. Comptes Rendus Mathematique **352**(3), 263 – 268 (2014)
13. Knezevic, D., Nguyen, N., Patera, A.: Reduced basis approximation and a posteriori error estimation for the parametrized unsteady Boussinesq equations. Mathematical Models and Methods in Applied Sciences **21 (7)**, 1415–1442 (2011)
14. Krylov, N.V.: Lectures on Elliptic and Parabolic Equations in Sobolev Spaces. Amer Mathematical Society (2008)
15. Lassila, T., Rozza, G.: Parametric free-form shape design with pde models and reduced basis method. Computer Methods in Applied Mechanics and Engineering **199**(23-24), 1583–1592 (2010)
16. Løvgren, A., Maday, Y., Rønquist, E.: A reduced basis element method for the steady Stokes problem. Mathematical Modelling and Numerical Analysis **40**(3), 529–552 (2006)
17. Løvgren, A., Maday, Y., Rønquist, E.: A reduced basis element method for complex flow systems. Proceedings of ECCOMAS CFD, P. Wesseling, E. Onate, J. Periaux (Eds.) TU Delft, The Netherlands (2006)
18. Løvgren, A., Maday, Y., Rønquist, E.: Global C^1 maps on general domains. Mathematical Models and Methods in Applied Sciences (M3AS) **19**(5), 803–832 (2009)
19. Maday, Y., Rønquist, E.: A reduced-basis element method. J.Sci. Comput. **17**, 447–459 (2002)
20. Maday, Y., Rønquist, E.: The reduced-basis element method: Application to a thermal fin problem. SIAM J.Sci. Comput. **26**, 240–258 (2004)
21. Maier, I., Haasdonk, B.: A Dirichlet-Neumann reduced basis method for homogeneous domain decomposition problems. Applied Numerical Mathematics **78**, 31–48 (2014)
22. Maier, I., Rozza, G., Haasdonk, B.: Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-Darcy system. Applied Computational Mathematics (2014). In press.

23. Phuong Huynh, D.B., 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**, 213–251 (2013)
24. Quarteroni, A.: Numerical Models for Differential Problems. Springer, Series MS&A , Vol. 8, 2nd ed. (2014)
25. Quarteroni, A., Rozza, G.: Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. Numer. Methods Partial Differential Equations **23**(4), 923–948 (2007)
26. Quarteroni, A., Rozza, G., Manzoni, A.: Certified reduced basis approximation for parametrized partial differential equations in industrial applications. J. Math. Ind. **1**(3) (2011)
27. Quarteroni, A., Valli, A.: Numerical Approximation of Partial Differential Equations. Springer Verlag (2008)
28. Rozza, G.: Reduced-basis methods for elliptic equations in sub-domains with *a posteriori* error bounds and adaptivity. Appl. Numer. Math. **55**(4), 403–424 (2005)
29. Rozza, G.: Reduced basis methods for Stokes equations in domains with non-affine parameter dependence. Comput. Vis. Sci. **12(1)**, 23–35 (2009)
30. Rozza, G., Huynh, D., Manzoni, A.: Reduced basis approximation and a posteriori error estimation for stokes flows in parametrized geometries: roles of the inf-sup stability constants. Numerische Mathematik **125**(1), 115–152 (2013)
31. Rozza, G., Huynh, D., Patera, A.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Arch. Comput. Methods Engrg. **15**, 229–275 (2008)
32. Rozza, G., Nguyen, C., Patera, A., Deparis, S.: Reduced basis methods and a posteriori error estimators for heat transfer problems. Proceedings of HT2009, ASME Summer Heat Transfer Conference, San Francisco, CA, USA, paper HT 2009–88211 **2**, 753–762 (2009)
33. Rozza, G., Veroy, K.: On the stability of the reduced basis method for Stokes equations in parametrized domains. Comput. Meth. Appl. Mech. Engr. **196**(7), 1244–1260 (2007)
34. Veroy, K., Patera, A.: Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. International Journal for Numerical Methods in Fluids **47 (8-9)**, 773–788 (2005)
35. Volpert, V.: Elliptic Partial Differential Equations. Fredholm Theory of Elliptic Problems in Unbounded Domains. Birkhauser (2011)