Proper Generalized Decomposition solutions within a Domain Decomposition strategy†

Antonio Huerta*[1] | Enrique Nadal[2] | Francisco Chinesta[3]

[1]Laboratori de Calcul Numeric (LaCaN), Universitat Politècnica de Catalunya, Barcelona, Spain
[2]Centro de Investigación en Ingeniería Mecánica, Departamento de Ingeniería Mecánica y de Materiales, Universitat Politècnica de València, Valencia, Spain
[3]ESI Group International Chair, Laboratoire PIMM, École Nationale des Arts et Métiers (ENSAM), Paris, France

Correspondence
Antonio Huerta,
Laboratori de Càlcul Numèric (LaCàN),
E.T.S. Ingenieros de Caminos, Canales y Puertos,
Universitat Politècnica de Catalunya,
Jordi Girona 1, E-08034 Barcelona, Spain.
Email: antonio.huerta@upc.edu

Abstract

Domain Decomposition strategies and the Proper Generalized Decomposition are efficiently combined to obtain a fast evaluation of the solution approximation in parameterized elliptic problems with complex geometries. The classical difficulties associated to the combination of layered domains with arbitrarily oriented mid-surfaces, which may require in-plane–out-of-plane techniques, are now dismissed. More generally, solutions on large domains can now be confronted within a Domain Decomposition approach. This is done with a reduced cost in the offline phase. Because, the Proper Generalized Decomposition gives an explicit description of the solution in each subdomain in terms of the solution at the interface. Thus, the evaluation of the approximation in each subdomain is a simple function evaluation given the interface values (and the other problem parameters). The interface solution can be characterized by any a priori user-defined approximation. Here, for illustration purposes, hierarchical polynomials are used. The repetitiveness of the subdomains is exploited to reduce drastically the offline computational effort. The online phase requires to solve a nonlinear problem to determine all the interface solutions. But this problem only has degrees of freedom on the interfaces and the Jacobian matrix is explicitly determined. Obviously, other parameters characterizing the solution (material constants, external loads, geometry) can also be incorporated in the explicit description of the solution.

KEYWORDS:
Reduced Order Models; Proper Generalized Decomposition; Domain Decomposition; Parameterized solutions

1 | INTRODUCTION

Proper Generalized Decomposition (PGD) [1, 2, 3, 4] has proven its advantages and applicability in many parameterized problems. PGD reduces the computational complexity induced by a large number of dimensions (the sum of number of spatial dimensions plus the number of parameters) to the iterative resolution of low dimensional problems, usually 1-D or 2-D. Moreover, the PGD approach provides an explicit expression of the approximated solution. Thus, the *online* phase, that is the evaluation of the approximation given the parameters, is very fast and does not require any extra interpolation or another solve (typical, for instance, in Proper Orthogonal Decomposition strategies).

However, PGD has never been combined with Domain Decomposition strategies. The methodology proposed here makes it possible. The resolution is divided into two phases. The first one builds (*offline*) a PGD model (i.e. an explicit solution) of the Dirichlet-to-Neumann operator for each subdomain as a function of given parameters (materials, geometry...) and of the displacement/temperature along the interface. The imposition of displacement/temperature along the interface is done with Nitsche's method in order to grant the interface values its own discretization. Then, the second phase, the *online* one, imposes the weak balance of fluxes at the interface. It results in a small nonlinear system to be solved at the interface, which advantageously uses the explicit solutions in each subdomain to obtain the Jacobian matrix for the Newton-Raphson method. Since the PGD generalized approximation in each subdomain is explicit there are no subdomain solves.

This approach, which can be valuable in many sizable problems, is crucial when confronted to large problems with complex geometries. Splitting the original problem into different subdomains can be beneficial in many occasions. An obvious case occurs when the problem domain can be built as a non-overlapping union of a small set of subdomains geometrically similar, or when material parameters only affect one subdomain or are different from one subdomain to another, or with complex non-cartesian geometries. Recall that PGD separation in space works best in cartesian domains (generalized cubes) whereas its extensions to non-cartesian domains, although possible, see [5], may suffer as the complexity of the geometry increases.

Implementing a domain decomposition strategy with PGD is nonetheless mandatory when the domain is composed of subdomains where the PGD separation strategy is different from subdomain to subdomain. This is typical in problems obtained by the union of subdomains having non collinear local coordinates. For instance problems combine plates and shells with arbitrarily oriented mid-surfaces or mixed with solids areas.

In recent papers, PGD has shown its capabilities to solve shell (plate) structures with a 3D formulation at a 2D computational cost and without typical simplifications of shell (plate) elements, see [6, 7, 8]. They use an *in-plane–out-of-plane* strategy for the spatial PGD separation. This has even been extended to flow problems [9]. Such advantages can only be exploited if all shells are aligned. Here, the formulation proposed overcomes these limitations.

Domain Decomposition (DD) techniques [10] allow to overcome the difficulties of standard FE in order to: i) decrease the size of the problem, ii) combine different physics or iii) use different approximation methods in each subdomain. The subdomains generated with the DD technique are coupled via artificial boundaries. The problem coupling all subdomains can be solved with a wide range of alternatives [10]. In this work, only non-overlapping methods are considered. Because, even the simplest implementation of overlapping DD with an in-plane–out-of-plane separated representation presents too many geometrical restrictions. In particular, the requirement of separability on the overlapping domains imposes major restrictions. A first attempt to combine PGD and DD strategies can be found in [11] where the Arlequin coupling strategy was considered. This reference shows the difficulties associated to the treatment of Lagrange multipliers, the geometrical transformation associated with general quadrilateral domains and the solution procedure for computing monolithically the in-plane and then the out-of-plane problems. Moreover, only homogeneous materials were considered to avoid the difficulties of arbitrarily oriented laminates and overlapping DD. For all these reasons, non-overlapping DD techniques are preferred in the presence of space separated representations for describing complex laminate structures.

In this context, it is possible to distinguish three different approaches: *i)* Dirichlet-Neumann [12], *ii)* Neumann-Neumann and *iii)* Dirichlet-Dirichlet methods. A popular variant of the latter is the Finite Element Tearing and Interconnecting (FETI) method [13] or its extensions. The methodology proposed here is, however, closer to the concept introduced in [14] of interfaces with a discretizations independent of the subdomain and using Lagrange multipliers for coupling the subdomains. Here Nitsche's method [15, 16, 17, 18] is used instead.

More recently, the idea of mixing DD techniques with Reduced Order Models (ROM) has show increasing interest. The aim being problems in which the domain can be built as a non-overlapping union of a small set subdomains geometrically similar [19, 20, 21, 22, 23, 24, 25, 26]. In this sense, [19] proposes the Reduced Basis (RB) hybrid method to solve Stokes flow in a domain composed by different non-overlapping subdomains. These subdomains are grouped into geometrically similar subdomains, that is, subdomains with the same number of sides in 2D. For each set of similar subdomains a parametric RB technique is used. The parameters are geometrical ones. Therefore, by only generating a reduced basis for a small set of subdomain types, the method is able to generate a global reduced basis for the whole problem domain. This approach requires at the online phase to solve a global problem that includes the reduced basis (standard in this technique) plus a coarse finite element mesh (to capture the normal fluxes on the interfaces) and Lagrange multipliers (to impose the continuity of the primal function on the interface). This is further improved in [26] by introducing (apart from the geometry) a set of parameters that characterize the profile of the function on the internal interfaces of the computational domain. This implies to solve, at the *online* phase, a system with has as

many unknowns as parameters characterizing the solution on the interfaces plus all the reduced basis for each subdomain of the DD. In [26] the proposed approach is further compared with other reduced basis strategies [27, 19, 28, 23].

In this paper, in order to combine domains with different tensor product geometries, a DD approach within a PGD framework is proposed. Thus obtaining generalized solutions for each subdomain and reducing drastically the cost of the online phase. This approach is inspired by the original Schur complement method, which is considered the simplest and first version of non-overlapping DD methods. The important feature is that it evaluates a parametric solution in each subdomain. The rationale proposed here requires first to obtain a parameterized PGD approximation for each set of similar subdomains. This *offline* phase is similar to [19, 26] but computes a generalized solution instead of a reduced basis. The parameters characterize the Dirichlet boundary conditions along the boundaries of these subdomains that are interfaces of the DD. These parameters describe the approximations used along the interface. Here for simplicity hierarchical polynomials are considered along the interfaces, without loss of generality and for illustration purposes.

Then, the *online* phase requires to solve a global problem, which in this case is a nonlinear system of equations, but only with unkowns along the interfaces. Moreover, the Newton-Raphson method can be readily applied because the Jacobian is easily computed. In fact, this global problem is established to guarantee equilibrium over the interfaces (transmission conditions for the fluxes). Note that the proposed approach allows for non-conforming discretizations between subdomains. More precisely, in the next sections it will be clear that this approach imposes weakly both the continuity of the solution and the equilibrium of the normal fluxes along the interfaces.

Except for one numerical example, here, for simplicity, the parameters are restricted to the boundary conditions along the interface. But, in practice, this PGD approach can be extended easily to a more general framework with material or geometrical parameters [2, 29, 30].

Note also that another advantage of the present approach is that the coupling problem is independent of the particular numerical method used in each subdomain. For instance, the proposed methodology allows to solve structures that are composed of degenerated subdomains, in which the space separation (in-plane–out-of-plane) is a must, and others in which that space separation is not required or even when in some subdomains the solution is governed by a nonlinear equation.

2 | PROBLEM STATEMENT

The heat equation is used here for presentation purposes, because the same strategy can be readily extended to a more complex mechanical behavior. Moreover, also for presentation purposes, the geometry is simplified from the previously discussed shell reinforced structures to an open bounded domain in $n_{\mathtt{sd}}$ dimensions, that is $\Omega \subset \mathbb{R}^{n_{\mathtt{sd}}}$, with boundary $\partial\Omega$ split in the Dirichlet, Γ_D, and Neumann, Γ_N, parts. The strong form of the steady heat equation is well known:

$$\begin{cases} -\boldsymbol{\nabla}\cdot(\mathbf{K}\boldsymbol{\nabla}u) = f & \text{in } \Omega, \\ \boldsymbol{n}\cdot(\mathbf{K}\boldsymbol{\nabla}u) = t & \text{on } \Gamma_N \subset \partial\Omega, \\ u = u_D & \text{on } \Gamma_D := \partial\Omega \setminus \Gamma_N, \end{cases} \tag{1}$$

where u is the unknown temperature, $\mathbf{K}$ is the thermal conductivity matrix, f the source/sink body term, t the imposed heat flux on the Neumann boundary and u_D the imposed temperature on the Dirichlet boundary. Note that Robin boundary conditions, which are typical in thermal problems, can also be considered. They are not included here to simplify the presentation.

Furthermore, Ω is partitioned in $n_{\mathtt{dd}}$ disjoint subdomains Ω_i with boundaries $\partial\Omega_i$, such that

$$\overline{\Omega} = \bigcup_{i=1}^{n_{\mathtt{dd}}} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j,$$

and the union of all the DD interfaces Γ_k, for $k = 1, \dots, n_{\mathtt{fc}}$, between the subdomains is denoted as

$$\bigcup_{i=1}^{n_{\mathtt{fc}}} \Gamma_i = \Gamma = \bigcup_{i=1}^{n_{\mathtt{dd}}} \partial\Omega_i \setminus \partial\Omega.$$

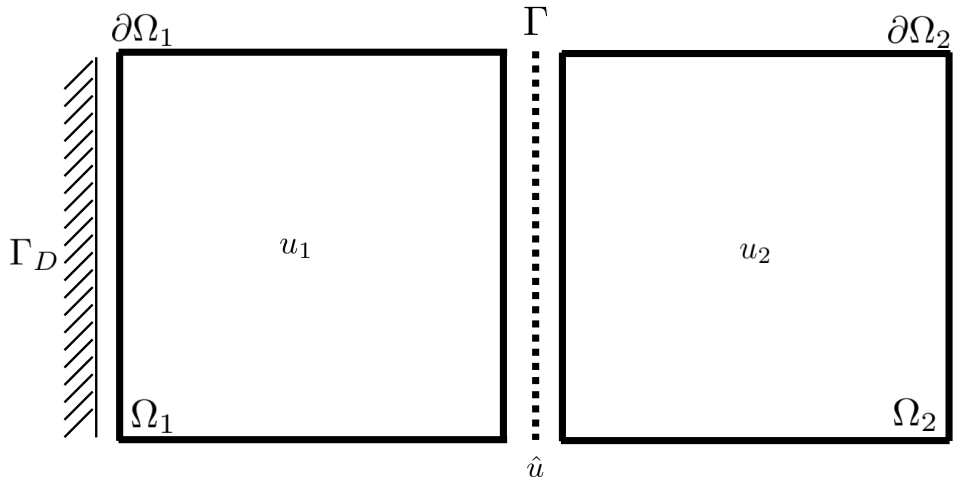

FIGURE 1 Splitting of the whole domain Ω into two domains Ω_1 and Ω_2.

The discontinuous setting induces a new problem equivalent to (1). That is, the exact solution of (1) also satisfies the following problems,

$$\left.\begin{aligned} -\nabla\cdot(\mathbf{K}_i\nabla u_i) &= f && \text{in } \Omega_i,\\ \boldsymbol{n}_i\cdot(\mathbf{K}_i\nabla u_i) &= t && \text{on } \partial\Omega_i\cap\Gamma_N,\\ u_i &= u_D && \text{on } \partial\Omega_i\cap\Gamma_D, \end{aligned}\right\} \text{for } i=1,\dots,n_{\text{dd}} \tag{2a}$$

$$[\![\boldsymbol{un}]\!] = \mathbf{0} \qquad \text{on } \Gamma, \tag{2b}$$

$$[\![\boldsymbol{n}\cdot(\mathbf{K}\nabla u)]\!] = 0 \qquad \text{on } \Gamma. \tag{2c}$$

Note that the *jump* $[\![\cdot]\!]$ operator has been introduced, it follows the definition in [31]. That is, along each portion of the interface Γ it sums the values from the left and right of Γ (say, Ω_i and Ω_j), namely

$$[\![\circledcirc]\!] = \circledcirc_i + \circledcirc_j,$$

and always must contain the normal vector $\boldsymbol{n}$.

3 | THE DOMAIN DECOMPOSITION STRATEGY

The rationale proposed here is based on defining local problems for each subdomain and a global problem to determine the interface parameters characterizing each local problem. Note that the domain decomposition strategy presented here is not an algebraic domain decomposition strategy, but a physical-based strategy adapted for mechanical problems. The local problems will be solved offline using the PGD approach, as shown in the next section, and the global problem, which is nonlinear, is solved online. The unknowns of the global problem are new variables characterizing the primal unknown along the interfaces. This approach is described as a hybrid method [32] because a new variable is introduced on the interface, which coincides with the trace of the unknown. Figure 1 introduces schematically the notation used in what follows.

3.1 | The local problem

By introducing a new variable $\hat{u}$ along the interface Γ problem (2) becomes

$$\begin{cases} -\nabla\cdot(\mathbf{K}_i\nabla u_i) = f & \text{in } \Omega_i,\\ \boldsymbol{n}_i\cdot(\mathbf{K}_i\nabla u_i) = t & \text{on } \partial\Omega_i\cap\Gamma_N,\\ u_i = u_D & \text{on } \partial\Omega_i\cap\Gamma_D,\\ u_i = \hat{u} & \text{on } \partial\Omega_i\cap\Gamma, \end{cases} \tag{3}$$

for $i=1,\dots,n_{\text{dd}}$, which accounts for (2a) and (2b), leaving the flux continuity imposed by (2c) for the global equation. Note that the set of problems defined in (3) are similar to those used by the first phase of a Neumann-Neumann based DD method [10].

Solving problem (3) in each subdomain determines the solution u as a function of the unknown $\hat{u}$ on the interface Γ. The weak problem associated to (3) is defined such that the Dirichlet boundary conditions are imposed weakly following a Nitsche rationale [15, 16, 17, 18]. Thus, it consists in finding $u_i \in \mathcal{H}^1(\Omega_i)$, for all $w \in \mathcal{H}^1(\Omega_i)$ such that

$$\int_{\Omega_i} \nabla w \cdot (\mathbf{K}_i \nabla u_i)\, d\Omega - \int_{\partial\Omega_i \setminus \Gamma_N} w\, \boldsymbol{n}_i \cdot \widehat{(\mathbf{K}_i \nabla u_i)}\, d\Gamma = \int_{\Omega_i} w\, f\, d\Omega + \int_{\partial\Omega_i \cap \Gamma_N} w\, t\, d\Gamma,$$

where the traces of the fluxes, denoted by $\widehat{\mathbf{K}_i \nabla u_i}$, are imposed as

$$\boldsymbol{n}_i \cdot \widehat{(\mathbf{K}_i \nabla u_i)} = \begin{cases} \boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla u_i) - \tau_i (u_i - u_D) & \text{on } \partial\Omega_i \cap \Gamma_D, \\ \boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla u_i) - \tau_i (u_i - \hat{u}) & \text{on } \partial\Omega_i \cap \Gamma, \end{cases}$$

and τ_i is a stabilization parameter, whose selection has an important effect on the stability and accuracy of the results and is proportional to $1/h$ where h is the characteristic element size. More precisely, in the examples shown below, τ_i is taken as

$$\tau_i = 2 \max k_i / \min h_i.$$

That is, the maximum value of the diffusion coefficient at subdomain i divided by the minimum characteristic size of the element at subdomain.

Finally, the symmetrized local weak problem becomes: find $u_i \in \mathcal{H}^1(\Omega_i)$ such that

$$a_i(u_i, w) = \ell_i(w) \quad \forall w \in \mathcal{H}^1(\Omega_i) \tag{4a}$$

with

$$a_i(v, w) = \int_{\Omega_i} \nabla w \cdot (\mathbf{K}_i \nabla v)\, d\Omega - \int_{\partial\Omega_i \setminus \Gamma_N} w\, \boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla v)\, d\Gamma - \int_{\partial\Omega_i \setminus \Gamma_N} v\, \boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla w)\, d\Gamma + \int_{\partial\Omega_i \setminus \Gamma_N} w\, \tau_i\, v\, d\Gamma, \tag{4b}$$

and

$$\ell_i(w) = \int_{\Omega_i} w\, f\, d\Omega + \int_{\partial\Omega_i \cap \Gamma_N} w\, t\, d\Gamma - \int_{\partial\Omega_i \cap \Gamma_D} u_D\, \boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla w)\, d\Gamma - \int_{\partial\Omega_i \cap \Gamma} \hat{u}\, \boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla w)\, d\Gamma + \int_{\partial\Omega_i \cap \Gamma_D} w\, \tau_i\, u_D\, d\Gamma + \int_{\partial\Omega_i \cap \Gamma} w\, \tau_i\, \hat{u}\, d\Gamma. \tag{4c}$$

It is important to remark that a similar strategy has been implemented imposing all Dirichlet boundary conditions (i.e. on Γ_D and Γ) as essential ones, see APPENDIX A:. However, imposing u_D on Γ_D and $\hat{u}$ along Γ as essential boundary conditions, in spite of being feasible, has two drawbacks. On one hand, as usual on PGD approaches, essential boundary conditions require and ad hoc initial mode. And, on the other hand, weak imposition of these conditions allow to use non conforming meshes on each side of the interface.

3.2 | The interface problem

The discrete counterpart of local problem described in the previous section for each subdomain allows to evaluate approximations of u as functions of $\hat{u}$. This unknown $\hat{u}$ is determined with a interface problem or global problem, which is constructed imposing continuity of fluxes along the DD interface Γ, see (2c). These are the transmission conditions for the fluxes. The corresponding global weak problem can be stated as: find $\hat{u} \in \mathcal{L}_2(\Gamma)$ for all $\mu \in \mathcal{L}_2(\Gamma)$ such that

$$\sum_{i=1}^{n_{\text{dd}}} \left\{ \int_{\partial\Omega_i \setminus \partial\Omega} \mu \left[\boldsymbol{n}_i \cdot (\mathbf{K}_i \nabla u_i) - \tau_i u_i \right] d\Gamma + \int_{\partial\Omega_i \setminus \partial\Omega} \mu\, \tau_i\, \hat{u}\, d\Gamma \right\} = 0. \tag{5}$$

In fact, the function $\hat{u} \in \mathcal{L}_2(\Gamma)$ is defined by multiple subfunctions, with each subfunction applying to an interface of the domain decomposition, namely

$$\hat{u}(\boldsymbol{x}) := \hat{u}_k(\boldsymbol{x}) \qquad \text{for } \boldsymbol{x} \in \Gamma_k,$$

and $\hat{u}_k \in \mathcal{L}_2(\Gamma_k)$ for $k = 1, \ldots, n_{\text{fc}}$.

4 | THE PGD DOMAIN DECOMPOSITION STRATEGY

The previous section allows to split the original problem into many local problems, which determine the primal solution as a function of the interface $\hat{u}$, and one global problem to solve precisely for $\hat{u}$. As usual, the unknow function $\hat{u}$ is characterized by

a set of coefficients $(\hat{u}_1, \dots, \hat{u}_{\hat{n}})^T = \hat{\mathbf{u}}$. Consequently, each local problem is now a *parameterized* problem and the parameters are precisely the coefficients in $\hat{\mathbf{u}}$. The PGD strategy proposed here, corresponding to the classical PGD formulation, solves all the parameterized local problems in the *offline* phase in order to obtain the generalized solution in terms of $\hat{u}$, i.e. the Dirichlet boundary conditios as parameters. In case of similar subdomains, the number of *offline* PGD problems is always much smaller than the number of subdomains, see the examples in Section 5. Subsections 4.1 to 4.3 describe how to define and obtain generalized solutions for each subdomain during the *offline* phase of the PGD strategy. Then Subsection 4.4 describes the global problem that implies solving a nonlinear system of equations to determine $\hat{u}$. Note that the Jacobian matrix needed to implement a Newton-Raphson scheme can be easily computed. This global solve is the only computational burden of the *online* phase because once $\hat{u}$ is determined the primal variable u is determined on-the-fly given the explicit expressions for each subdomain obtained by the PGD.

4.1 | The parameterized subdomain problem

To proceed with this strategy a parametric characterization of $\hat{u}$ along Γ is required, for instance

$$\hat{u}(\boldsymbol{x}) = \sum_{s=1}^{\hat{n}} \widehat{N}_s(\boldsymbol{x})\, \hat{\mathrm{u}}_s = \widehat{\mathbf{N}}^T(\boldsymbol{x})\, \hat{\mathbf{u}}, \qquad \text{for } \boldsymbol{x} \in \Gamma, \tag{6}$$

where $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{n}}$ are the coefficients and functions $\widehat{N}_s(\boldsymbol{x})$ are the basis used to represent the solution along the interfaces. As noted earlier, this approximation is constructed from the subfunctions on each interface Γ_k, for $k = 1, \dots, n_{\mathtt{fc}}$. In fact, for each interface

$$\hat{u}_k(\boldsymbol{x}) = \sum_{r=1}^{\hat{v}_k} \widehat{N}^k_r(\boldsymbol{x})\, \hat{\mathrm{u}}^k_r \qquad \text{for } \boldsymbol{x} \in \Gamma_k, \tag{7}$$

where $\widehat{N}^k_r$ are hierarchic shape functions derived from the set of integrated Legendre polynomials as introduced in [33]. To simplify the presentation the approximation degree in every interface is uniform $\hat{v}_k = \hat{v}$. Consequently, $\hat{n} = \hat{v}\, n_{\mathtt{fc}}$ but, as commented in Remark 1, this is not necessary.

The parametrization of $\hat{u}$ defines a finite dimensional space $\hat{\mathcal{V}} \subset \mathcal{L}_2(\Gamma)$, with $\dim \hat{\mathcal{V}} = \hat{n}$. The global problem (5) is solved precisely in $\hat{\mathcal{V}}$. But this space is completely independent of the trial spaces $\mathcal{V} \subset \mathcal{H}^1(\Omega_i)$ used in each local problem (each subdomain). That is, $\widehat{N}_s(\boldsymbol{x})$ is not related to the shape functions of the mesh in Ω_i for $i = 1, \dots, n_{\mathtt{dd}}$. And consequently, non-conforming meshes between subdomains are naturally taken into consideration.

The strategy proposed here, which follows precisely the *computational vademecum* rationale proposed in [2], is to solve each local problem (4) using parameters $\hat{\mathbf{u}}$, which characterize $\hat{u}$, as new coordinates. Thus, the objective is to compute $u_i(\boldsymbol{x}, \hat{\mathbf{u}})$ with $(\boldsymbol{x}, \hat{\mathbf{u}}) \in \Omega_i \times \boldsymbol{\mathcal{I}} \subset \mathbb{R}^{n_{\mathtt{sd}}} \times \mathbb{R}^{\hat{n}}$. Where $\boldsymbol{\mathcal{I}} \subset \mathbb{R}^{\hat{n}}$ characterizes the range for parameters $\hat{\mathbf{u}}$ and can be defined as the Cartesian combination of the range for each parameter, namely, $\boldsymbol{\mathcal{I}} := \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_{\hat{n}}$ with $\hat{\mathrm{u}}_s \in \mathcal{I}_s$ for $s = 1, \dots, \hat{n}$. This range is user-defined depending on the expected values, a priori physical knowledge of the problem (for instance, avoiding non-physical temperatures), and the imposed boundary conditions (and source terms).

The solution $u_i(\boldsymbol{x}, \hat{\mathbf{u}})$ formally lies in a tensor product space, namely, $u_i \in \mathcal{V}_i \otimes \mathcal{L}_2(\mathcal{I}_1) \otimes \mathcal{L}_2(\mathcal{I}_2) \otimes \cdots \otimes \mathcal{L}_2(\mathcal{I}_{\hat{n}})$. A standard weighted residuals approach, with integrals in $\Omega_i \times \boldsymbol{\mathcal{I}}$ and the usual integration by parts only in Ω_i produces a weak form in this multi-dimensional setup.

For simplicity, in order to avoid cumbersome notation, the subdomain indices will not be explicitly indicated unless it is crucial to understand the development. The weak problem for each subdomain becomes: find $u \in \mathcal{V} \otimes \mathcal{L}_2(\mathcal{I}_1) \otimes \mathcal{L}_2(\mathcal{I}_2) \otimes \cdots \otimes \mathcal{L}_2(\mathcal{I}_{\hat{n}})$ such that

$$A(u, v) = L(v), \ \forall v \in \mathcal{V} \otimes \mathcal{L}_2(\mathcal{I}_1) \otimes \mathcal{L}_2(\mathcal{I}_2) \otimes \cdots \otimes \mathcal{L}_2(\mathcal{I}_{\hat{n}}), \tag{8a}$$

with the following definitions of the bilinear and linear forms

$$A(u, v) := \int_{\mathcal{I}_1}\int_{\mathcal{I}_2}\cdots\int_{\mathcal{I}_{\hat{n}}} a(u, v)\, d\hat{\mathrm{u}}_{\hat{n}} \cdots d\hat{\mathrm{u}}_2\, d\hat{\mathrm{u}}_1 \ \text{ and } L(v) := \int_{\mathcal{I}_1}\int_{\mathcal{I}_2}\cdots\int_{\mathcal{I}_{\hat{n}}} \ell(v)\, d\hat{\mathrm{u}}_{\hat{n}} \cdots d\hat{\mathrm{u}}_2\, d\hat{\mathrm{u}}_1, \tag{8b}$$

that require the previously defined spatial forms (4).

It is important to recall that the resolution of (8a) in a PGD context requires an affine parameter dependence of the different forms. This as usual in reduced order methods and it is very well discussed in [34, 35]. In fact, finding the affine parameter dependence of (8b) is discussed in subsequent sections.

4.2 | The PGD approximation in each subdomain

As usual in a PGD strategy, a separated representation u^n_{PGD} is imposed to approximate the solution of (8) in each subdomain, namely

$$u(\boldsymbol{x},\hat{\mathbf{u}}) \approx u^n_{\text{PGD}}(\boldsymbol{x},\hat{\mathbf{u}}) = \sum_{m=1}^{n}\left[F^m_{x_1}(x_1)\,F^m_{x_2}(x_2)\prod_{s=1}^{\hat{n}}\widehat{F}^m_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s)\right] = u^{n-1}_{\text{PGD}}(\boldsymbol{x},\hat{\mathbf{u}}) + F^n_{x_1}(x_1)\,F^n_{x_2}(x_2)\prod_{s=1}^{\hat{n}}\widehat{F}^n_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s) \tag{9}$$

The function u^n_{PGD} stands for the PGD approximation with n terms (or modes) and is defined as a sum of separable terms. Each term (mode) is the product of functions depending only on one of the arguments. This separation as a sum of products of unknown functions is standard in PGD, see for instance [3, 4]. Obviously, this implies that the unknown interface parameters are arguments of unknown functions $\widehat{F}^n_{\hat{\mathrm{u}}_s}$. Note that in some of the PGD implementations the separated modal functions are normalized and therefore a scalar coefficient affects each mode and characterizes its amplitude.

Moreover, it is important to note that expression (9) already considers a spatial separated representation of the solution. This space separation can precisely coincide with an in-plane (x_1) out-of-plane (x_2) strategy, which is useful when the geometry of the corresponding subdomain is a degenerated geometry such as a laminate [8]. Of course, all developments that follow in this section can also be applied when no space separation is required.

As usual, the separated approximation proposed by (9) is determined using a greedy algorithm. This assumes that u^{n-1}_{PGD} is known and the last mode, $F^n_{x_1}(x_1)\,F^n_{x_2}(x_2)\prod_{s=1}^{\hat{n}}\widehat{F}^n_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s)$, is computed replacing (9) into (8). Then, the fixed-point solver for this nonlinear problem is applied, details can be found elsewhere see for instance [2, 30].

4.3 | Affine parameter dependence

As indicated previously, the greedy strategy implies replacing (9) into (8) to determine the last mode. This entails solving a nonlinear problem to compute $F^n_{x_1} \in \mathcal{V}_{x_1}$, $F^n_{x_2} \in \mathcal{V}_{x_2}$, $\widehat{F}^n_{\hat{\mathrm{u}}_1} \in \mathcal{L}_2(I_1),\dots,\widehat{F}^n_{\hat{\mathrm{u}}_{\hat{n}}} \in \mathcal{L}_2(I_{\hat{n}})$ for all $\delta F_{x_1} \in \mathcal{V}_{x_1}$, $\delta F_{x_2} \in \mathcal{V}_{x_2}$, $\delta\widehat{F}_{\hat{\mathrm{u}}_1} \in \mathcal{L}_2(I_1),\dots,\delta\widehat{F}_{\hat{\mathrm{u}}_{\hat{n}}} \in \mathcal{L}_2(I_{\hat{n}})$ such that

$$A\left(F^n_{x_1}\,F^n_{x_2}\prod_{s=1}^{\hat{n}}\widehat{F}^n_{\hat{\mathrm{u}}_s},\upsilon\right) = L(\upsilon) - A(u^{n-1}_{\text{PGD}},\upsilon)$$

with

$$\upsilon = \delta F_{x_1}\,F_{x_2}\prod_{s=1}^{\hat{n}}\widehat{F}_{\hat{\mathrm{u}}_s} + F_{x_1}\,\delta F_{x_2}\prod_{s=1}^{\hat{n}}\widehat{F}_{\hat{\mathrm{u}}_s} + \sum_{r=1}^{\hat{n}}F_{x_1}\,F_{x_2}\,\delta\widehat{F}_{\hat{\mathrm{u}}_r}\prod_{\substack{s=1\\ s\neq r}}^{\hat{n}}\widehat{F}_{\hat{\mathrm{u}}_s}.$$

It is important to realize that the definition of these forms, see (8b), and the separated structure of the approximation allow to clearly see the affine parameter dependence, namely

$$A\left(F^n_{x_1}\,F^n_{x_2}\prod_{s=1}^{\hat{n}}\widehat{F}^n_{\hat{\mathrm{u}}_s},\upsilon\right) = \int_{I_1}\widehat{F}^n_{\hat{\mathrm{u}}_1}\int_{I_2}\widehat{F}^n_{\hat{\mathrm{u}}_2}\cdots\int_{I_{\hat{n}}}\widehat{F}^n_{\hat{\mathrm{u}}_{\hat{n}}}\,a\left(F^n_{x_1}\,F^n_{x_2},\upsilon\right)d\hat{\mathrm{u}}_1\cdots d\hat{\mathrm{u}}_{\hat{n}},$$

$$L(\upsilon) = \int_{I_1}\int_{I_2}\cdots\int_{I_{\hat{n}}}\left[\tilde{\ell}(\upsilon) + \sum_{s=1}^{\hat{n}}\hat{\mathrm{u}}_s\,\hat{\ell}_s(\upsilon)\right]d\hat{\mathrm{u}}_1\cdots d\hat{\mathrm{u}}_{\hat{n}},$$

$$\tilde{\ell}(\upsilon) = \int_{\Omega_i}\upsilon\,f\,d\Omega + \int_{\partial\Omega_i\cap\Gamma_N}\upsilon\,t\,d\Gamma - \int_{\partial\Omega_i\cap\Gamma_D}u_D\,\boldsymbol{n}_i\cdot(\mathbf{K}\boldsymbol{\nabla}\upsilon)\,d\Gamma + \int_{\partial\Omega_i\cap\Gamma_D}\upsilon\,\tau_i\,u_D\,d\Gamma,$$

$$\hat{\ell}_s(\upsilon) = \int_{\partial\Omega_i\cap\Gamma}\upsilon\,\tau_i\,\widehat{N}_s\,d\Gamma - \int_{\partial\Omega_i\cap\Gamma}\widehat{N}_s\,\boldsymbol{n}_i\cdot(\mathbf{K}\boldsymbol{\nabla}\upsilon)\,d\Gamma.$$

The details on the greedy approach and the alternate direction method to solve this nonlinear problem can be found elsewhere, see for instance [2, 3, 4, 30].

4.4 | Implementation aspects of the global problem

The generalized solution of every local problem provides an explicit expression of the solution in each subdomain for any value of the primal variable on the interfaces. Recall that only a small set of subdomains are actually solved, only those that are not similar and present variations in topology, boundary conditions on non-interface borders, different in-plane–out-of-plane strategies, etc. A trivial Schur complement strategy can now be implemented by means of imposing the flux transmission conditions on the interface Γ.

First, the global problem defined in (5) must be solved. As noted in Section 4.1, the parametrization of $\hat{u}$ defines a finite dimensional space $\hat{\mathcal{V}} \subset \mathcal{L}_2(\Gamma)$, with $\dim \hat{\mathcal{V}} = \hat{n}$. Since the global problem (5) is solved in $\hat{\mathcal{V}}$, the test functions μ are in $\hat{\mathcal{V}} = \text{span}\{\widehat{N}_1, \dots, \widehat{N}_{\hat{n}}\}$. Consequently, the nonlinear system of $\hat{n}$ equations and $\hat{\mathbf{u}} \in \mathbb{R}^{\hat{n}}$ unknowns is obtained after replacing the approximation $u^n_{\text{PGD}}(\boldsymbol{x}, \hat{\mathbf{u}}) \approx u_i(\boldsymbol{x}, \hat{\mathbf{u}})$ in (5), namely

$$\mathbf{F}(\hat{\mathbf{u}}) = \sum_{i=1}^{n_{\text{dd}}} \Big\{ \int_{\partial\Omega_i \setminus \partial\Omega} \widehat{\mathbf{N}}(\boldsymbol{x}) \big[\boldsymbol{n}_i \cdot \big(\mathbf{K}_i \boldsymbol{\nabla} u^n_{\text{PGD}}(\boldsymbol{x}, \hat{\mathbf{u}}) \big) - \tau_i u^n_{\text{PGD}}(\boldsymbol{x}, \hat{\mathbf{u}}) \big] d\Gamma + \int_{\partial\Omega_i \setminus \partial\Omega} \widehat{\mathbf{N}}(\boldsymbol{x})\, \tau_i\, \widehat{\mathbf{N}}^T(\boldsymbol{x})\, \hat{\mathbf{u}}\, d\Gamma \Big\} = \mathbf{0}. \tag{10}$$

Note that the nonlinearity comes from the fact that the expression of $u^n_{\text{PGD}}(\boldsymbol{x}, \hat{\mathbf{u}})$, see (9), involves sums of products of functions, $\widehat{F}^m_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s)$, of the unknowns $\hat{\mathbf{u}} = [\hat{\mathrm{u}}_1, \dots, \hat{\mathrm{u}}_{\hat{n}}]^T$.

Second, it is important to note that once the local problems are solved, the following expression can be readily computed

$$\frac{\partial u^n_{\text{PGD}}(\boldsymbol{x}, \hat{\mathbf{u}})}{\partial \hat{\mathrm{u}}_s} = \sum_{m=1}^{n} \Big[F^m_{x_1}(x_1)\, F^m_{x_2}(x_2) \frac{\partial \widehat{F}^m_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s)}{\partial \hat{\mathrm{u}}_s} \prod_{\substack{r=1 \\ r \neq s}}^{\hat{n}} \widehat{F}^m_{\hat{\mathrm{u}}_r}(\hat{\mathrm{u}}_r) \Big].$$

And, consequently, the chain rule allows to explicitly compute the Jacobian matrix needed to implement a Newton-Raphson scheme for solving (10).

Finally, it is important to observe that in all the linear examples discussed in the following sections the number of iterations never exceeded three. The quadratic convergence of the Newton-Raphson method allows to obtain the desired accuracy in this reduced number of iterations.

Remark 1. Hierarchical shape functions. Here, a polynomial basis is used to describe $\hat{u}$ along each interface, see (7). Obviously other options are possible, for instance, piecewise polynomials or any other user-defined linear approximation along each interface. To show the applicability of the approach, without loss of generality, polynomials are used along each interface moreover to simplify the presentation the approximation degree in every interface is uniform. In fact, the user can decide, using or not a priori information on the problem, to use continuous functions with discontinuous derivatives on heterogeneous interfaces or to adapt, on each interface in Γ, the degree of the continuous polynomial depending on the problem at hand. See [25] for more details.

5 | NUMERICAL EXAMPLES

5.1 | Academic example with two subdomains

This is a simple example where PGD is not needed and, consequently, applying the PGD with domain decomposition is unnecessary. It is used to check convergence to the FEM solution as the approximation on the interface is enriched. The material is considered isotropic $\mathbf{K} = k\mathbf{I}$, where $\mathbf{I}$ is the identity matrix.The diffusion coefficient in the domain is $k = 1$. The computational domain Ω is partitioned in $\Omega_i = [i-2, i-1] \times [-1, 1]$ for $i = 1, 2$. Dirichlet and Neumann boundary conditions are imposed according to the schematic presentation shown in Figure 2 . The prescribed source term in (2) is defined as

$$f(x_1, x_2) = 50 \exp\big(-10(x_1 - 0.5)^2 - 10x_2^2\big) \quad \text{for } (x_1, x_2) \in [-1, 1] \times [-1, 1].$$

A uniform FEM mesh of 60×30 or 120×60 bilinear elements is used in complete domain Ω as a reference. For comparison, the PGD separation in x and y uses linear elements of the same characteristic size in each subdomain. Therefore, in each subdomain a mesh of 30×30 or 60×60 elements is used. Along the interface Γ, as noted earlier, a normalized hierarchical polynomial basis is used to describe the unknown $\hat{u}$, see (6). The parameters $\hat{\mathrm{u}}_s$ ($s = 1, \dots, \hat{n}$) are allowed to range in $\mathcal{I}_s := [-10, 10]$. Each domain $\mathcal{I}_s$ is discretized in 100 uniform elements.

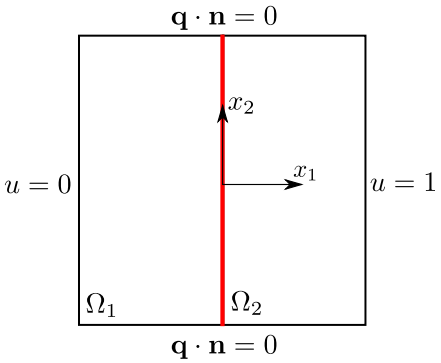

FIGURE 2 Academic example model problem and boundary conditions with an interface between two subdomains.

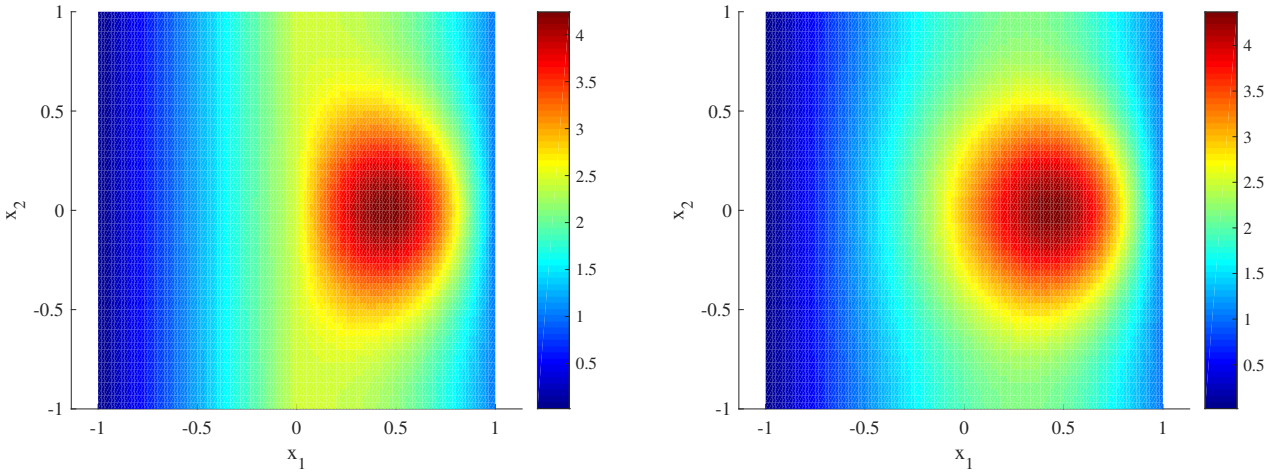

FIGURE 3 Academic example solution with a constant (left) and order 6th (rigth) approximation of $\hat{u}$ along the interface. 60×60 elements per subdomain are used.

Figure 3 shows the results obtained with the proposed method with two different approximations along the interface: constant, i.e. $\hat{\nu} = 1$, (left) and a 6th order hierarchical polynomial approximation, i.e. $\hat{\nu} = 7$, (right). Note that, due to the symmetry of the solution, only even-order polynomial are used (the odd ones behave exactly as the corresponding lower even approximation). The spatial mesh of 120×60 bilinear elements is used in the whole domain for the FEM reference solution. It corresponds to discretizations of 60 linear elements in the PGD separation. Figure 4 shows error maps between the Domain Decomposition PGD and the FEM approximation for the two interface parameterizations (constant and polynomial of degree 6). The absolute error is normalized by the maximum value of the solution in the domain (approximately 4.39). It is important to note that the FEM approximation on the 120×60 bilinear mesh is used to evaluate the error and it corresponds to an equivalent spatial discretization. Thus, this error map includes both the standard PGD truncation error, see [36], and the error induced by the approximation imposed along the interface. The second one is due to approximation of $\hat{u}$ along the interface and must be controlled. In fact, the influence of this error can be clearly seen comparing the two images in Figure 4 . Note that a crude constant approximation for $\hat{u}$ induces relative errors below 13.1%, whereas a 6th order polynomial approximation reduces the maximum relative error to 1.86%.

The computational cost of the offline and online phases and the number of modes required are shown in Table 1 . The offline phase implies solving two PGD problems since the source term is different in each subdomain. Since the offline computations are independent for each subdomain, the offline computational cost corresponds to the maximum cost for each subdomain.

The discretization is 60×60 elements per subdomain. Obviously, increasing the polynomial degree along the interface implies more modes of the PGD solution and more offline computational cost. The online cost is associated to the generation of the

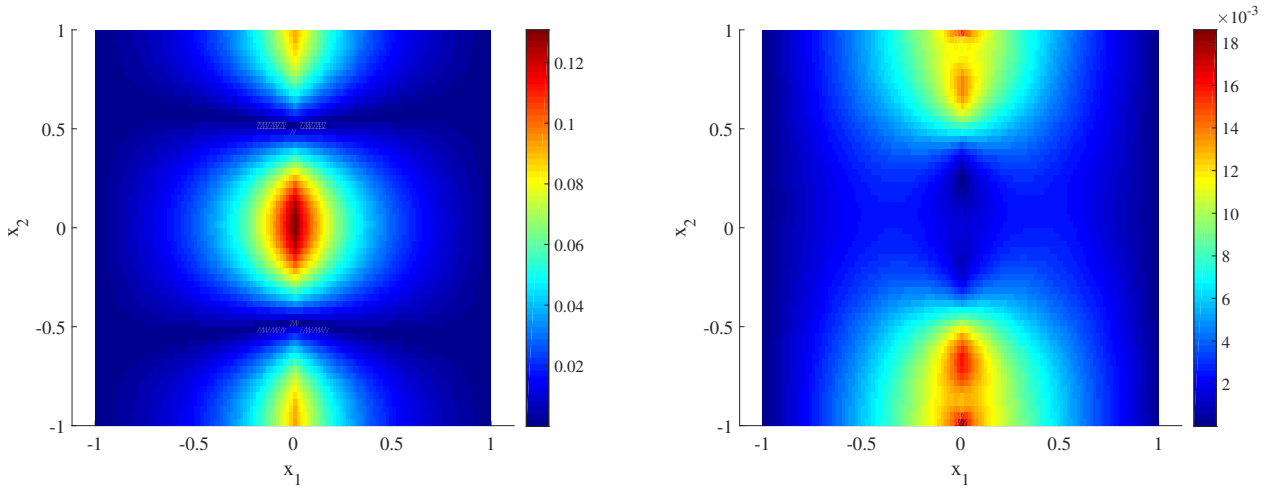

FIGURE 4 Academic example relative error distribution (with respect to the 120×60 bilinear FEM) for a constant (left) and order 6th (right) approximation of $\hat{u}$ along the interface. Note that the scales are different. 60×60 elements per subdomain are used.

degree	offline (s)	online (s)	modes Sub1/Sub2
0	3	0.004	30 / 60
2	10	0.005	40 / 160
4	716	0.012	720 / 700
6	1735	0.047	1000 / 1000

TABLE 1 Computational cost on a laptop and number of modes for the academic example with different degrees of approximation along the interface. A mesh of 60×60 elements per subdomain Ω_i is used.

Jacobian matrix needed to implement a Newton-Raphson scheme for solving (10) and the number of iterations. Obviously, this problem, which does not need a PGD separation in the subdomains and the decomposition, can be solved with FEM on a 120×60 mesh very fast, namely $0.077s$. However, even in this case, the online computational cost is smaller than the FEM computation. Next examples will show situations where the online solve is orders of magnitude smaller than the FEM resolution.

In fact, the influence of the polynomial order is further analyzed in Figure 5 where the $\mathcal{L}_2$ relative error along the interface is depicted as the polynomial order in the approximation of $\hat{u}$ is increased, at left. At right, the $\mathcal{L}_2$ relative error of the flux is also evaluated, considering the flux outwards subdomain Ω_1. The reference solution here is the one obtained with an overkilled FEM calculation on a mesh of 600×300 bilinear elements over the whole domain Ω. The dotted and solid lines correspond to DD PGD computations with spatial meshes of characteristic size $h = 1/30$ and $h = 1/60$ respectively. Additionally the influence of the truncation of the greedy algorithm used for the local (PGD) problem is analysed through the squared and diamond markers. The truncation tolerance is set to 10^{-4} and 10^{-5}, and is evaluated as the 2-norm of the last mode with respect to the first one. In the rest of the problems this tolerance is set to 10^{-5}.

There are three sources of error included in the curves of Figure 5 . The first two are standard in PGD approximations and correspond to the FE error and the PGD truncation, see [36]. The standard FE error is $\mathcal{O}(h^2)$, namely $\mathcal{O}(1/30^2) \approx 10^{-3}$ and $\mathcal{O}(1/60^2) \approx 3\,10^{-4}$ respectively. Note that it includes the weak imposition of the Dirichlet boundary condition on the interface using Nitsche's approach. The third source of error is associated to the approximation of $\hat{u}$ along the interface. The temperature on the interface instead of being piecewise linear (with the corresponding characteristic size of $h = 1/30$ or $h = 1/60$) is approximated by a polynomial of degree $\hat{v} - 1$. The error of approximating a piecewise linear function by a polynomial of increasing order is bounded below by $\mathcal{O}(h^2)$. Thus, the saturation observed in Figure 5 , which is one order of magnitude larger than h^2, is due to the PGD truncation and can be reduced by increasing the number of PGD modes n in (9) and the proper approximation of input data, see [36]. As you can appreciate, the increase in the accuracy when a smaller truncation error is considered, i.e. increasing the number of PGD modes, has only a marginal effect. A considerable higher number of modes

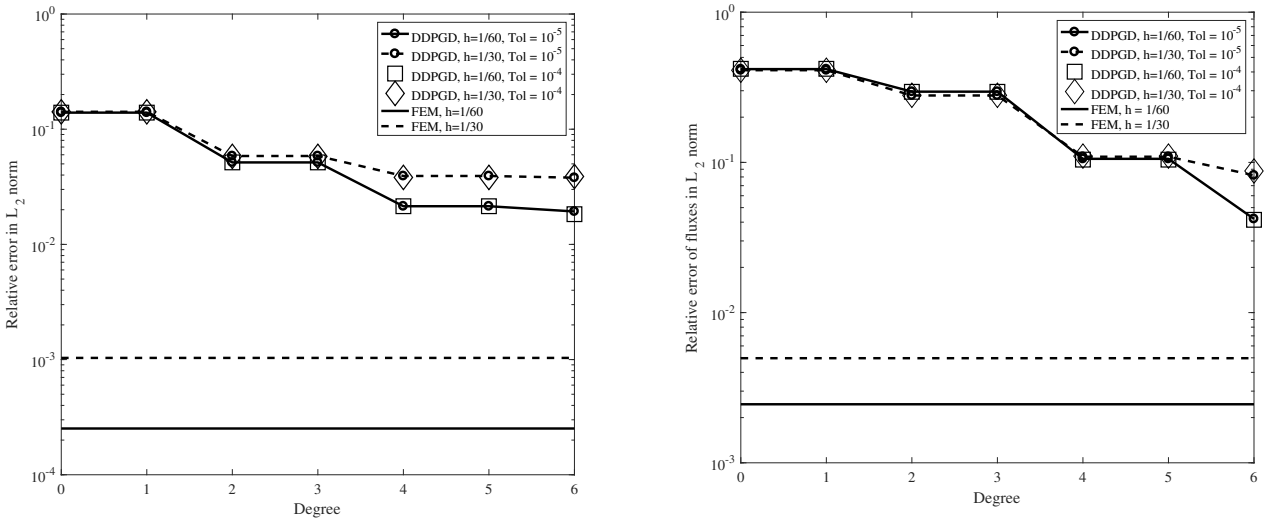

FIGURE 5 Academic example $\mathcal{L}_2$ relative error, with respect to a reference FEM solution obtained with 600×300 elements in Ω, along the interface versus polynomial order in the approximation of $\hat{u}$.

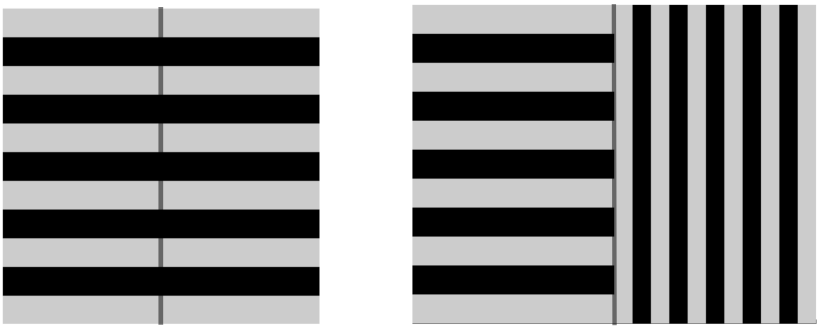

FIGURE 6 Academic example model with two layered subdomains. The interface between subdomains is in the middle and is represented by a grey vertical line.

should be added to be able appreciate the influence. However, this is not required since the accuracy levels that the proposed methodology shows are enough for engineering applications. On the other hand, the right graph in Figure 5 shows that the accuracy in the fluxes keeps increasing when the degree of approximation increases along the interface. As a difference from the solution error, the error in the flux does not seem to depend on the mesh size until a certain level of approximation, i.e. degree 6, is reached.

Finally, it is important to recall that as noted in Section 4.4 a nonlinear global problem is solved to determine the parameters characterizing the approximation of $\hat{u}$ along the interface. Quadratic convergence of the residual of the Newton-Raphson solver is always obtained since the Jacobian can be explicitly evaluated. Moreover, only three iterations are enough to obtain convergence (residual smaller than 10^{-8}). In fact, the next examples show that the convergence of the Newton-Raphson scheme is independent of the number of parameters and subdomains or interfaces.

5.2 | Layered academic example with two subdomains

The previous example (same source term and boundary conditions) is repeated here but with layered subdomains. Two cases are considered depending if layers are collinear or orthogonal in both subdomains, see Figure 6 . Each subdomain is composed by laminates: 5 fiber layers (black ones) and 6 matrix layers (gray ones). The thermal conductivity of the fiber layers ($k = 10$),

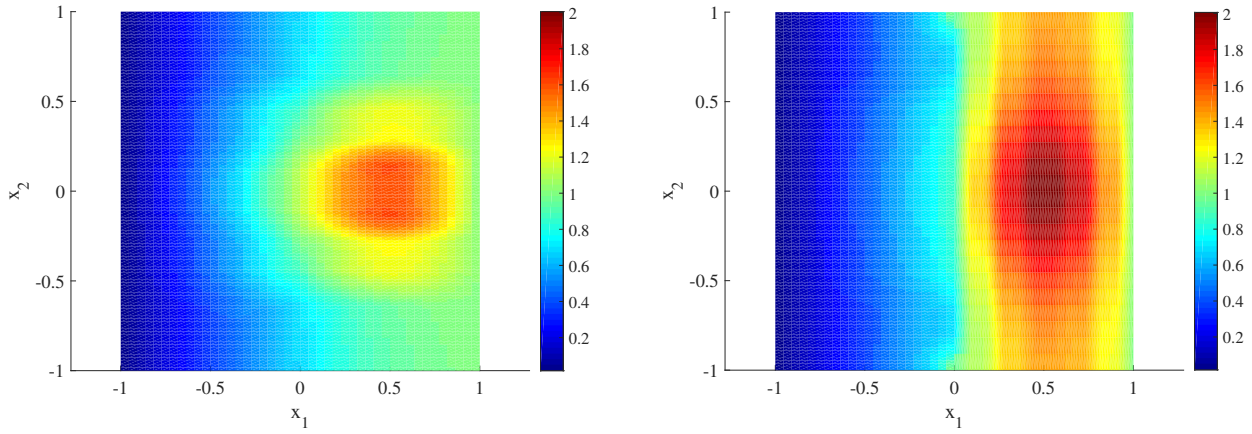

FIGURE 7 Solution for the collinear (left) and orthogonal (right) laminates.

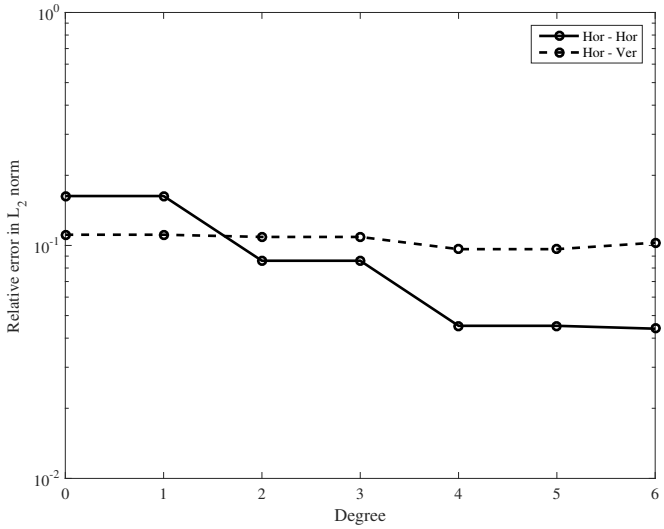

FIGURE 8 Layered subdomains, $\mathcal{L}_2$ relative error along the interface versus polynomial order in the approximation of $\hat{u}$.

is one order of magnitude larger than the one for the matrix layers ($k = 1$). The layered structure suggests for each subdomain an in-plane–out-of-plane formulation with 110 linear elements in the direction across the laminates and only 22 linear element in the layer direction. Note that the nonconformity along the interface for the example with orthogonal layers is not a problem in the proposed formulation. Because, as noted in section 2, along the interface temperature is imposed in weak form. Thus, the approach proposed here allows for non-conforming meshes at the interfaces. However, when the standard FEM is used in the whole domain for the orthogonal case, in order to have conformity along the interface Γ the mesh on the left subdomain must have also 110 linear element in the layer direction.

The influence of the laminates can be appreciated in Figure 7 . This figure shows the solutions obtained with a polynomial of degree 6 along the interface Γ. The in-plane–out-of-plane strategy combined with the domain decomposition produces reasonable results when compared with FE computations on an overkilled mesh, always conforming, of 2200×1100 bilinear elements. Note however from Figure 8 that as the degree on the interface increases the error *along the interface* decreases in the collinear case but remains constant in the orthogonal one. In this figure, the error is measured with respect to the previously indicated overkilled mesh. As expected, the hierarchical polynomials employed along the interface are not able to reproduce along the interface Γ the slope discontinuities of the temperature field across the laminates. Because as shown in Figure 9 double discontinuities (in both Cartesian directions) for the orthogonal layered case are difficult to capture by continuous polynomials.

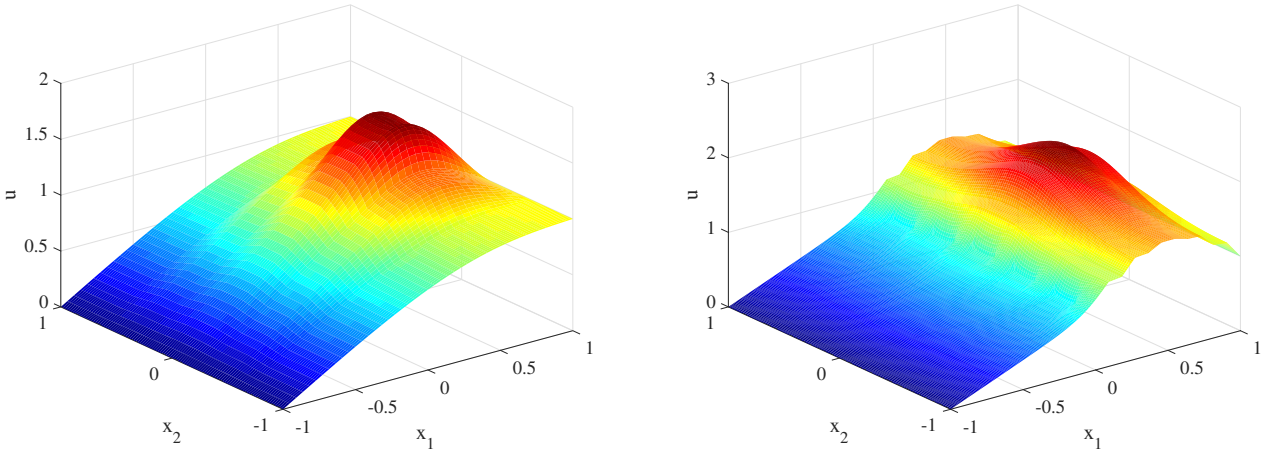

FIGURE 9 Layered subdomains with collinear (left) and orthogonal (orthogonal) layers FE reference solution.

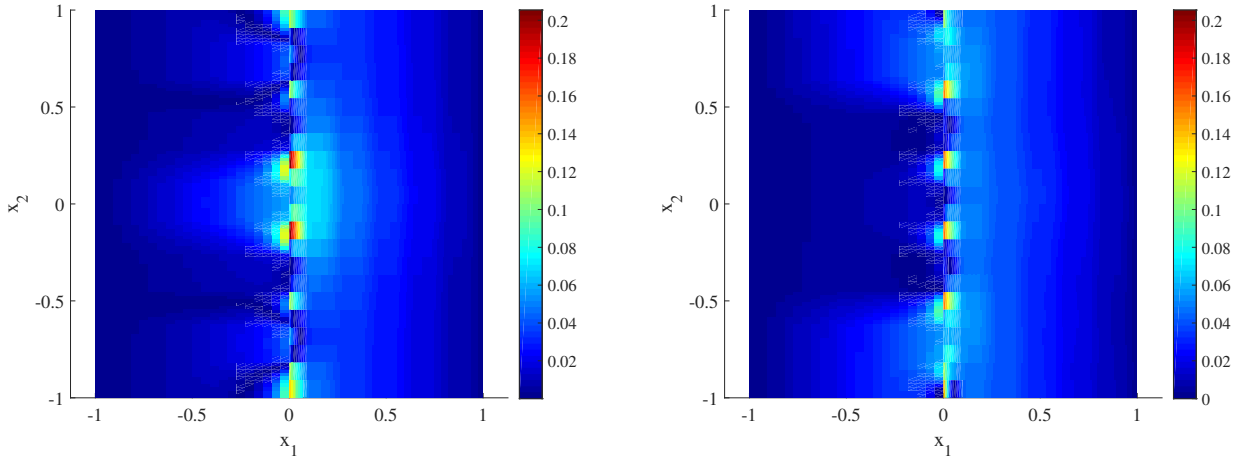

FIGURE 10 Orthogonal layered subdomains error distribution for a constant (left) and order 6th (right) approximation of $\hat{u}$ along the interface.

Consequently, if the interface area is of interest other approximations along the interface, such as piecewise polynomials, could be used instead of hierarchical shape functions.

In order to further analyze the quality of the proposed approach, error maps with respect to *equivalent* FE computations are plotted in Figure 10 . The equivalent FE computations require conforming meshes, thus a mesh of 110×220 bilinear elements is used as reference in Figure 10 . As expected from Figure 8 , the maximum error at the interface is not influenced as the polynomial degree is increased. However, as the order of the polynomial approximation on the interface increases, the error is clearly more localized and decreases away from the interface Γ.

Obviously, this issue is not present in the case of collinear laminates. Figure 11 shows the error with respect to an equivalent FE mesh of 110×44 bilinear elements. It can be observed that the error decreases drastically everywhere as the polynomial order increases on the interface. Finally, for the collinear configuration, Table 2 shows computational times for the offline, which implies solving two PGD problems because as noted in the previous example the source term is different in each subdomain (recall that only the maximum computational cost for each subdomain is indicated). Obviously, increasing the polynomial degree along the interface implies more modes of the PGD solution and more offline computational cost. As noted earlier, the online cost is associated to the generation of the Jacobian matrix needed to implement a Newton-Raphson scheme and the number of

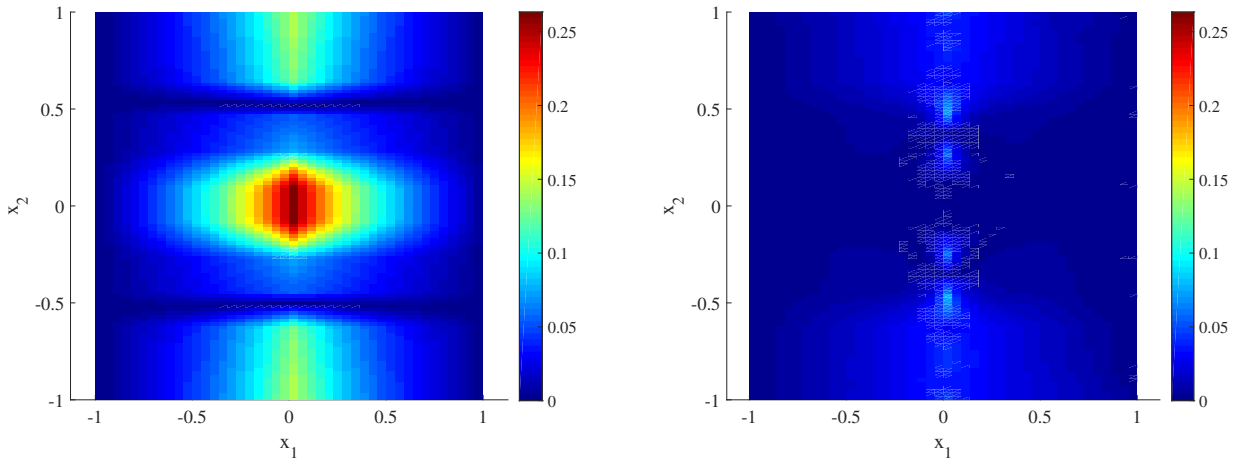

FIGURE 11 Collinear layered subdomains error distribution for a constant (left) and order 6th (right) approximation of $\hat{u}$ along the interface.

degree	offline (s)	online (s)	modes Sub1/Sub2
0	8	0.005	50 / 100
2	30	0.008	70 / 270
4	1503	0.039	1000 / 1000
6	1821	0.048	1000 / 1000

TABLE 2 Computational cost on a laptop and number of modes for the layered academic example in the collinear configuration with different degrees of approximation along the interface. A mesh of 22×110 elements per subdomain Ω_i is used.

iterations. Evidently, this academic problem can also be solved with FEM on a 110×44 mesh very fast, namely $0.06s$. As in the previous example, the online computational cost of the proposed technique is always smaller than the FEM computation.

In summary, the domain decomposition PGD strategy allows combining subdomains with different in-plane–out-of-plane orientations. The lack of mesh conformity is not a problem for this formulation and the approximation along the interface must be oriented to the desired quantities of interest.

5.3 | Composite thermal fin problem

This problem, inspired by [37, 26], models the heat diffusion in the fin depicted in Figure 12 with Dirichlet imposed temperatures at $\Gamma_D^{\texttt{in}}$ ($u = 1$) and $\Gamma_D^{\texttt{out}}$ ($u = 0$), and homogeneous Neumann elsewhere. In contrast with the cited previous references, here the material is composed by laminates with different orientations in each part of the geometry, see Figure 12 . As in the previous example, these laminates are composed of 5 fiber layers (black ones) and 6 matrix layers (gray ones). The thermal conductivity of the fiber layers ($k = 10$), is one order of magnitude larger than the one for the matrix layers ($k = 1$). To capture the features of the solution in the direction perpendicular to the laminate, an in-plane–out-of-plane strategy is used. Ten elements are considered for each layer, i.e. a total of 110 elements are used in the out-of-plane direction. Whereas, only 22 elements are employed along the in-plane direction.

The interface Γ, represented by shadowed lines, is now the union of all 15 sides joining these subdomains. In each one of these sides $\hat{u}$ is approximated by a polynomial of degree one or three. Given the temperatures imposed at $\Gamma_D^{\texttt{in}}$ and $\Gamma_D^{\texttt{out}}$, the parameters characterizing temperature at the interfaces are taken in $[-2, 2]$. All these 1D domains for each parameter are discretized with a uniform mesh of 100 linear elements.

As observed in Figure 12 , all the fins are geometrically similar. In fact, all of then can be transformed into a unit square with a simple scaling of each direction, see Figure 13 . The physical coordinates $\tilde{\boldsymbol{x}}$ are scaled by the physical dimensions of the subdomain $(x_1, x_2) = (\tilde{x}_1/L_1, \tilde{x}_2/L_2)$. Consequently, the thermal problem that originally was defined in the physical

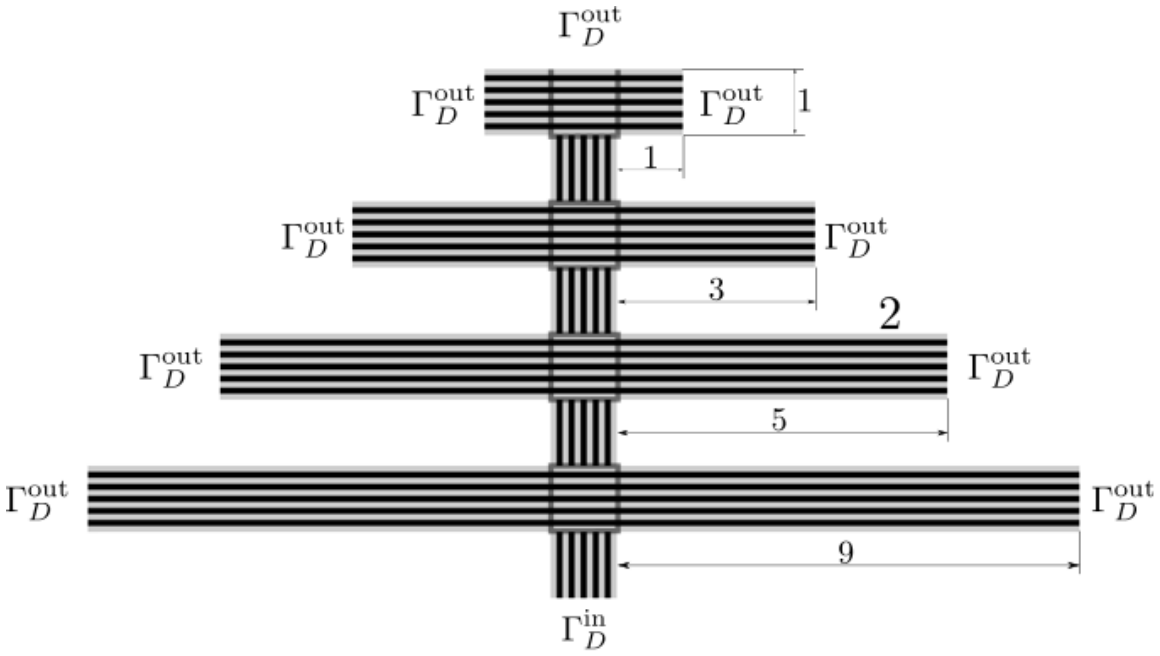

FIGURE 12 Composite thermal fin problem statement.

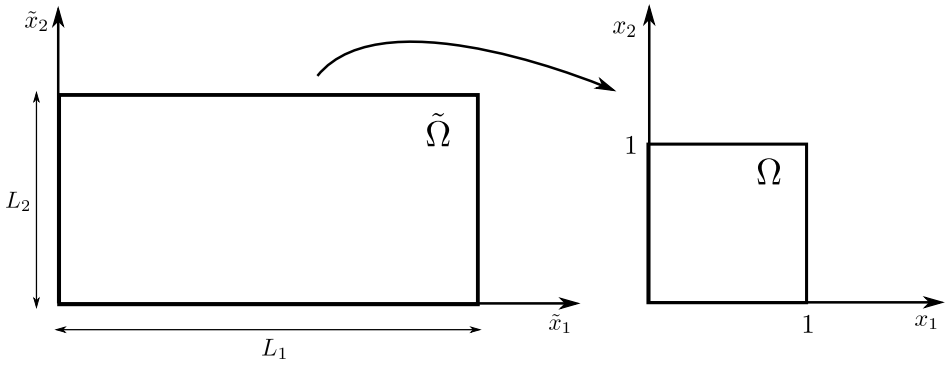

FIGURE 13 Scaling the physical domain into the unit square.

rectangle with layers in the $\tilde{x}_1$ direction an a thermal conductivity $\mathbf{K} = k(\tilde{x}_2)\mathbf{I}$ is now solved in the unit square (reference domain $]0, 1[\times]0, 1[$) with a thermal conductivity

$$\mathbf{K} = \begin{bmatrix} k(x_2)/L_1^2 & 0 \\ 0 & k(x_2)/L_2^2 \end{bmatrix}. \tag{11}$$

Obviously, for each subdomain the scaling may be changed.

In summary, the distinct geometrical domains are solved in the unit square as an orthotropic material, and, consequently, the geometry is simply parametrized as material constants. Thus, the PGD approximation in each subdomain includes now as parameter the dimensions L_1 and L_2, namely $u(\boldsymbol{x}, L_1, L_2, \hat{\mathbf{u}}) \approx u^n_{\text{PGD}}(\boldsymbol{x}, L_1, L_2, \hat{\mathbf{u}})$ with

$$u^n_{\text{PGD}}(\boldsymbol{x}, L_1, L_2, \hat{\mathbf{u}}) = \sum_{m=1}^{n} \left[F^m_{x_1}(x_1)\, F^m_{x_2}(x_2)\, F^m_{L_1}(L_1)\, F^m_{L_2}(L_2) \prod_{s=1}^{\hat{n}} \widehat{F}^m_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s) \right].$$

The dimensions of the subdomains are allowed to range two orders of magnitude, that is $L_i \in \mathcal{I}_{L_i} := [0.1, 10]$ for $i = 1, 2$. Each of these parametric domains $\mathcal{I}_{L_i}$ are discretized with a uniform mesh of 100 linear elements. Details on the PGD resolution of parameterized problems can be found, for instance, in [2, 3, 4].

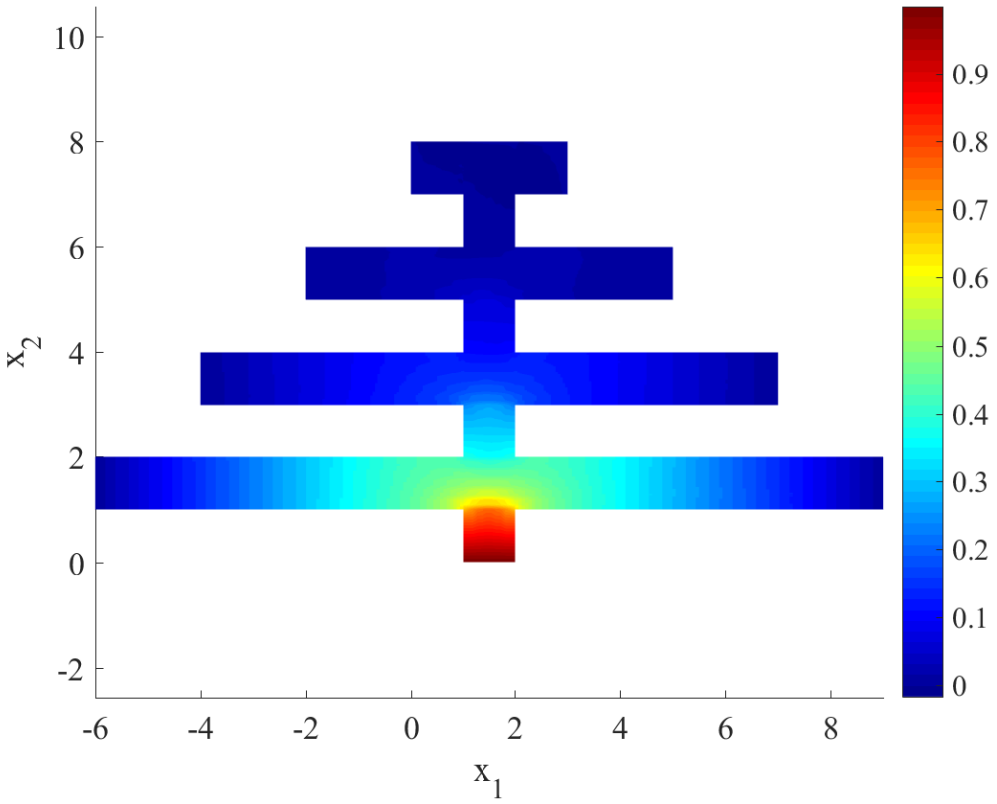

FIGURE 14 Composite thermal fin problem. Solution for a cubic approximation along the interfaces.

The transformation of every subdomain into a reference problem reduces drastically the number of subdomains requiring a PGD offline solution. Only 2 subdomains are required: 1) for all the tins of the left and right hand side subdomains and for the vertical connections of the tins (two opposite Dirichlet boundary conditions and two opposite homogeneous Neumann boundary conditions), and 2) for the horizontal connections of the tins (four Dirichlet boundary conditions).

Two cases are studied depending on the approximation along the interfaces: linear ($\hat{\nu} = 2$) or cubic ($\hat{\nu} = 4$). As observed earlier, here for simplicity the same approximation is used along all the interfaces but nothing precludes using different approximations along each interface depending on the richness of the temperature field. In fact, the hierarchical polynomials allow for an easy implementation of this approach.

Only the temperature field for the cubic interface approximation is depicted in Figure 14 , because the differences with the linear case are not distinguishable. Note that, in this problem the PGD approximation includes as extra parameters the dimensions of the subdomain (to account for the geometry variations) apart from the temperature on the interfaces.

The error map for the linear and cubic interface approximations are shown in Figure 15 . These errors are normalized by the maximum value of the solution (precisely $u = 1$ at Γ_D^{in}) and computed with respect to a FE solution on a mesh with the same resolution in the out-of-plane direction (orthogonal to the layers) and an *overkill* discretization in the in-plane-direction. This is done to avoid, in the FE computation, the use of nonconforming meshes along the interfaces where fibers orthogonal on each side. Note that the DD PGD proposed can handle these nonconforming interfaces naturally. With a linear approximation of the temperature on the interfaces, the maximum local error is near 11.4%. As expected, if approximation degree on the interfaces is increased, the error decreases and its maximum drops below 5.6%. Note also that, as commented previously, only three Newton-Raphson iterations suffice to converge with a tolerance of 10^{-8} and this is independent on the number of interfaces.

Finally, table 3 shows the computational time of the offline and online phases and the number of modes considered for the different subdomain types, with a mesh of 22×110 elements per subdomain. As in the previous examples, when the degree at the interfaces increases, the online computational cost slightly increases. The FEM solution for this problem requires a mesh of 110×110 elements for each of the 16 domains, with a total amount of 193600 elements, with a computational cost of $132s$.

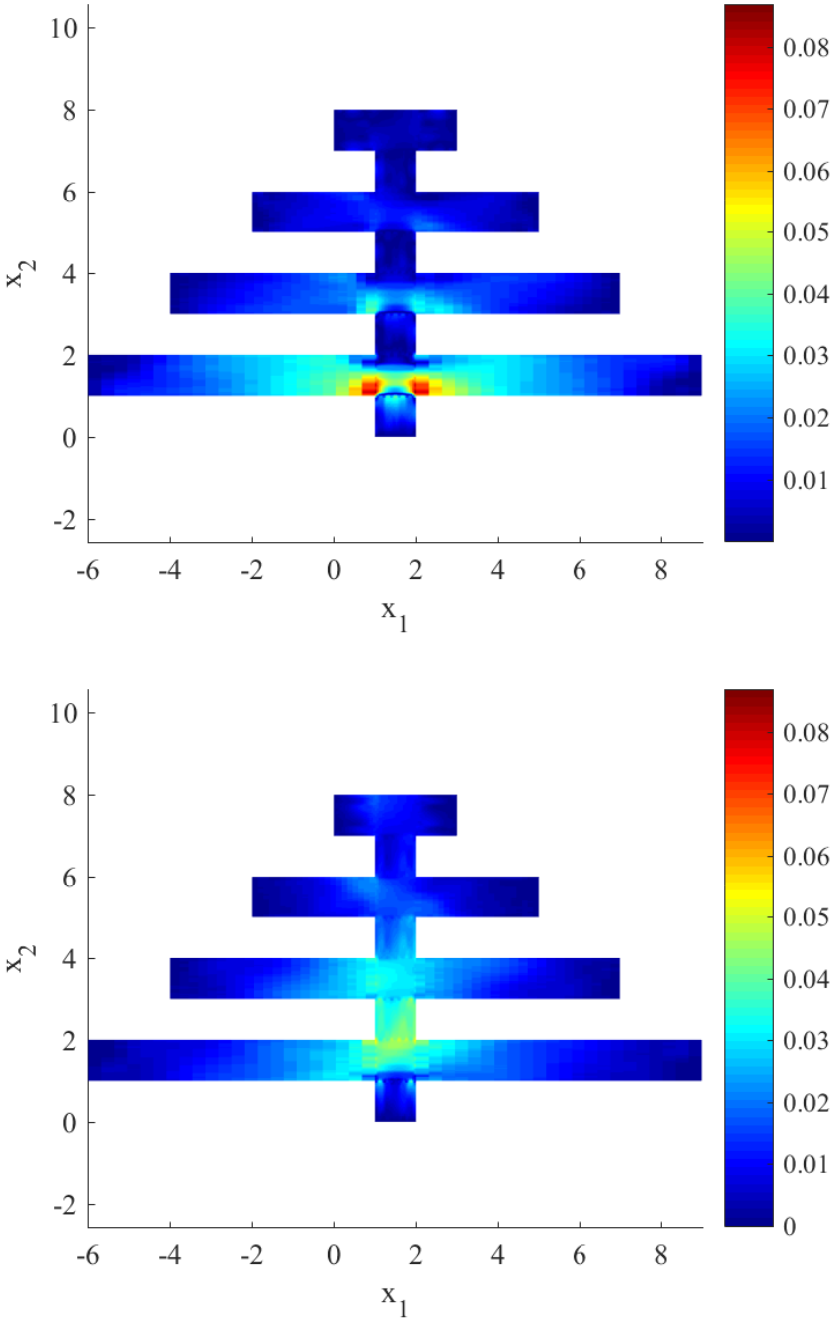

FIGURE 15 Composite thermal fin problem. Error map with respect to a FE mesh for a linear (top) and cubic (bottom) approximation of the solution along the interfaces.

For this problem the online computational cost is orders of magnitude smaller. Note that for this problem the variation of the dimensions of any subdomain will only require to perform another online evaluation and not the full process, considerably reducing now the computational cost with respect to FEM since the offline computation is performed only once. The extension for any other parameters is straightforward.

degree	offline (s)	online (s)	modes Sub: 1/2
1	1367	0.063	960 / 30
2	1918	0.166	1000 / 50
3	2277	0.404	1000 / 60

TABLE 3 Computational cost on a laptop and number of modes for the composite thermal fin example with different degrees of approximation along the interface. A mesh of 22×110 elements per subdomain Ω_i is used.

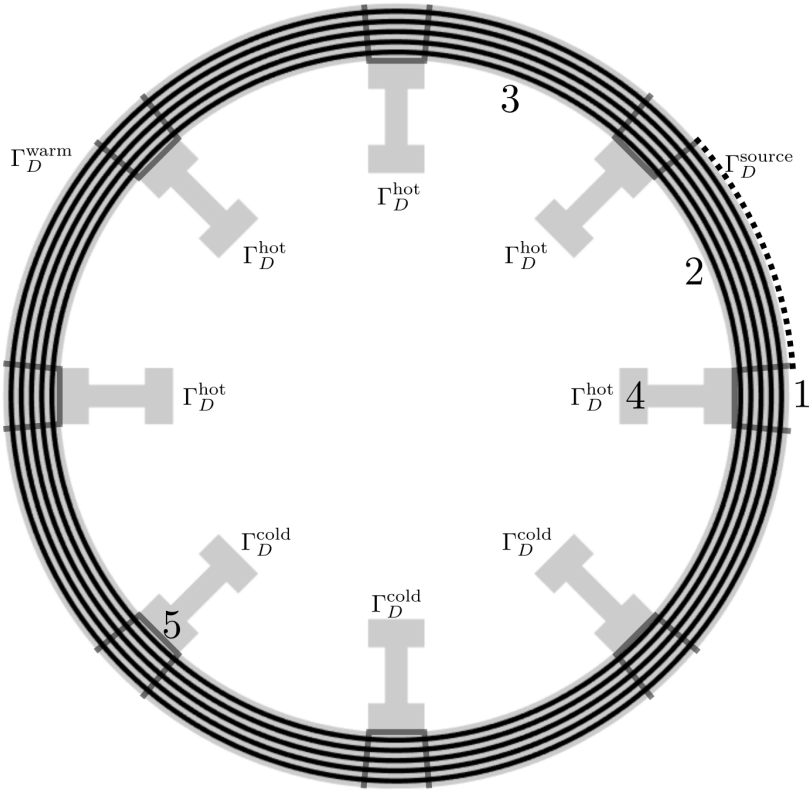

FIGURE 16 Laminate skin and stringers. Model of the problem and boundary conditions. The interface between subdomains is represented by a shadowed line.

5.4 | Laminate skin and stringers

This problem is devoted to show the capabilities of the proposed approach to a more complex and realistic geometry, see Figure 16 . The in-plane–out-of-plane PGD approach is need for the skin but adding the stringers precludes its use under classical implementations.

As shown in Figure 16 the domain is decomposed in 24 subdomains and its corresponding 24 interfaces (represented by shadowed lines). As in the previous example, the exterior annulus is formed by 11 layers (5 fibers depicted in black and 6 matrix in gray). The thermal conductivity in the fiber layers ($k = 10$) is one order of magnitude larger than the thermal ones ($k = 1$).

Given the geometry, the boundary conditions and the number of interfaces for each subdomain, five different typologies are identified. This imply that the PGD will only be applied to five different problems. These typologies are indicated in Figure 16 . Typologies 1 (skin connected to stringer), 2 (skin subjected to the external heat source on Γ_D^{source}) and 3 (skin with both external boundaries corresponding to Γ_D^{warm}) correspond to the skin, whereas 4 (stringer with one boundary Γ_D^{hot}) and 5 (stringer with one boundary Γ_D^{cold}) are associated to the stringers.

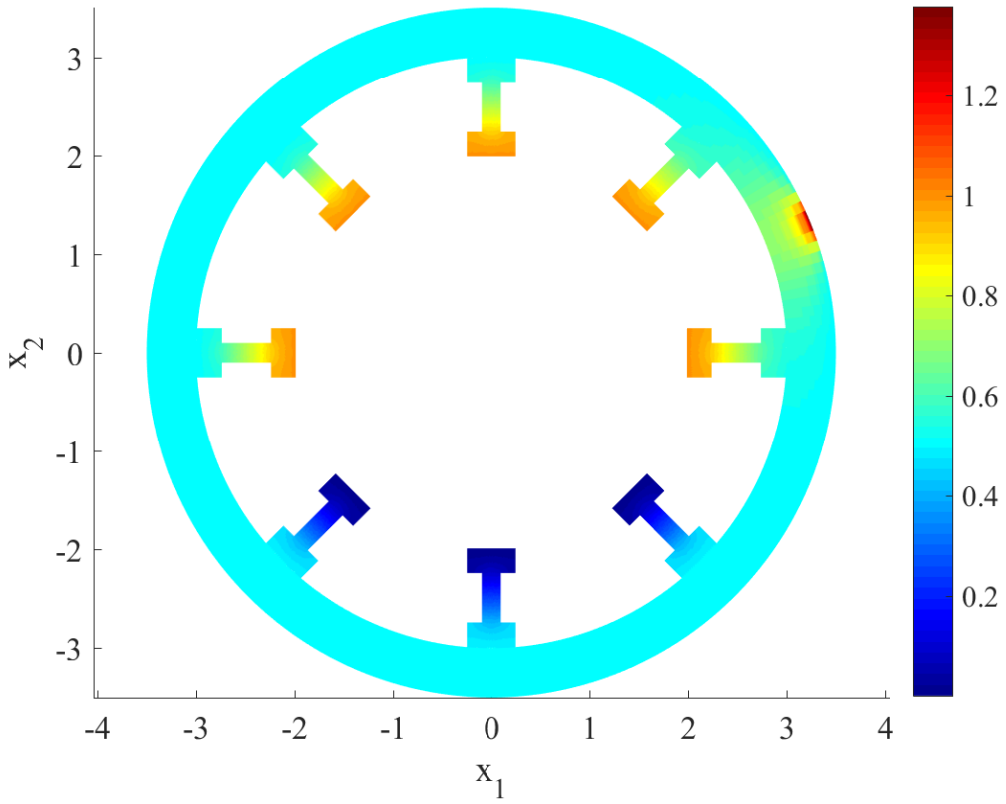

FIGURE 17 Laminate skin and stringers. Solution of the problem for a quadratic approximation of the solution along the interfaces.

The prescribed temperature at each boundary is

$$u_D = \begin{cases} 0 & \text{on } \Gamma_D^{\text{cold}}, \\ 1 & \text{on } \Gamma_D^{\text{hot}}, \\ 1/2 & \text{on } \Gamma_D^{\text{warm}}, \\ (1-s^2)\exp(-200s^2/9) + 1/2 & \text{on } \Gamma_D^{\text{source}}, \end{cases}$$

where $s \in [-1, 1]$ is the curvilinear coordinate along Γ_D^{source} (shown as a dotted line in Figure 16).

For subdomains corresponding to the skin (typologies 1, 2 and 3) and in order to capture the richness of the solution in the radial direction 110 1D linear elements are employed (10 elements per layer). Whereas only 20 1D linear elements are used along the circumferential direction for each subdomain. Stringers are discretized with a standard FE linear triangular elements for a total of 1105. The parametric space for each one of the parameters is $\Omega_s = [-2, 2]$ and it is discretized with 100 linear elements.

In this problem the temperature field along the interfaces is approximated linearly and quadratically. Figure 17 shows the PGD solution considering a quadratic approximation along the interface between all subdomains. The temperature field for the linear case is not shown because it is difficult to see any differences. But the improvement of the quadratic approximation is clearly seen if the error map is presented, see Figure 18 . These maps depict the error (normalized by the temperature imposed at Γ_D^{hot}, i.e. $u = 1$) between the PGD approximation and a FE computation on an equivalent mesh. For the linear approximation, the error is concentrated close to the right stringer between elements of typology 1 and 2 (8.6%). But just one extra degree of freedom on the interfaces reduces considerably the error (below 3.5%), which is now localized at the interfaces between stringers and annulus.

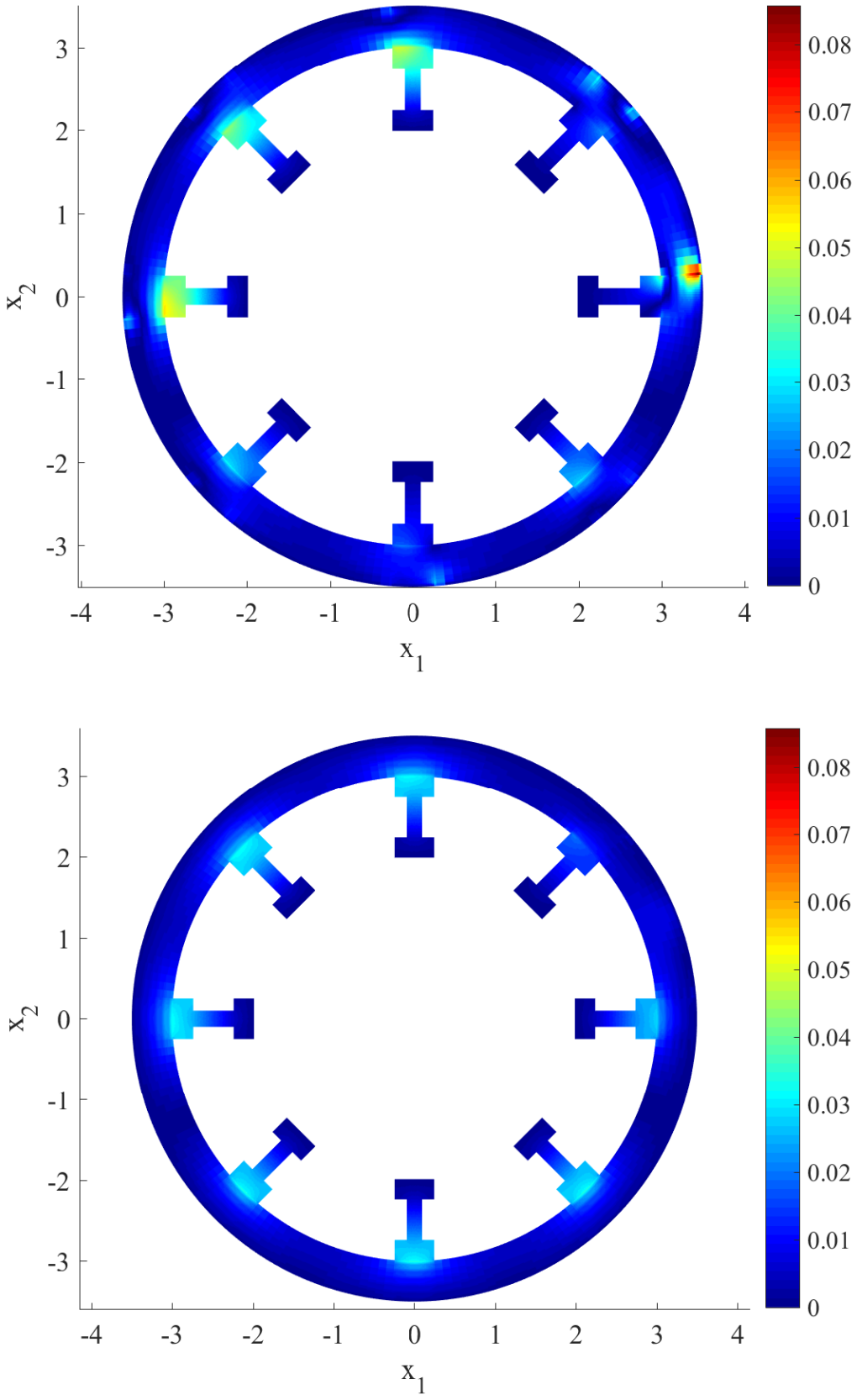

FIGURE 18 Laminate skin and stringers. Error map with respect to a FE mesh for a linear (top) and quadratic (bottom) approximation of the solution along the interfaces.

degree	offline (s)	online (s)	modesSub: 1/2/3/4/5
1	529	0.263	420 / 590 / 390 / 11 / 11
2	3374	1.167	1000 / 1000 / 1000 / 21 / 11

TABLE 4 Computational cost on a laptop and number of modes for the laminate skin and stringers example with different degrees of approximation along the interface. A mesh of 20×110 elements per subdomain Ω_i is used.

As introduced in the previous examples, the convergence of the nonlinear solver used for the global problem along the interfaces does not depend on the number of interfaces neither the number of parameters used to define the solution along the interfaces since it converges in only three iterations for a tolerance lower than 10^{-8}.

Finally, table 4 shows the computational time of the offline and online phases and the number of modes considered for the different subdomain types, considering a mesh of 20×110 elements per subdomain, with linear and quadratic approximation along the interfaces. As in the previous case, the equivalent FEM problems is also solved using an equivalent mesh with a computational cost of $43s$. As in the previous problems, the online computational cost is smaller than the FEM computational cost.

5.5 | Nonlinear problem

In this section we extend the proposed methodology to material nonlinear models. To illustrate its capabilities, the first academic example is revisited, see Section 5.1 and Figure 2 . The thermal source term is ignored but now the thermal conductivity is dependent on the temperature, in particular, it is assumed that

$$k(u) = 10 + 7u,$$

where u is the temperature field. This nonlinearity does not affect to the general methodology. The only difference is that the local parametric solution is obtained by solving the corresponding nonlinear parametric problem with the PGD. Among the different choices for addressing the nonlinearity, the simplest option is considered here: a fixed point strategy. In this case, the affine parameter dependence of the bilinear form is ensured because the conductivity can always be expressed in separated form given the polynomial dependence of k and the separated expression for the temperature $u \approx \underset{\text{PGD}}{u^{n-1}}$. Thus, the nonlinear solution does not implies a major difficulty.

Along the interface a linear approximation of the temperature field is considered. From the a-priori knowledge provided by the linear model solution discussed in Section 5.1, the parametric space for each one of the parameters was chosen in $\mathcal{I}_s := [-0.5, 2.5]$. A uniform mesh of 100 linear elements is used in each of theirs segments. The space coordinates are discretized using 30 linear elements in each direction.

Figure 19 shows the solution obtained with the proposed method. The influence of the material nonlinearity induces a slight qualitative difference with respect to the linear case where the temperature field is expected to increase linearly along the x_1-coordinate. Figure 19 also shows the error with respect to the FE solution using an equivalent mesh and the same nonlinearity. The error on the interface is almost negligible (because the approximation can exactly represent the exact solution on the interface), and the error in each subdomain is mainly due to the truncation of the finite sum decomposition.

Finally, figure 20 shows the convergence of the Newton-Raphson strategy used for solving the global problem at the interfaces. In this case, and despite the fact of having a final solution even more simple that the solution related to the linear problem (having neglected here the localized thermal source term) 5 iterations are needed to reach the same precision that the one considered in the linear case. This fact proves that now, the precomputed parametric solution have a more complex dependence with respect to the parameters related to the boundary conditions.

6 | CONCLUSIONS

A new approach has been proposed to obtain an explicit description of the solution in parameterized elliptic problems with complex geometries. In this context, complex geometries include multi-domains decomposed in arbitrarily oriented repetitive topologies. The proposed approach allows to combine different formulations, among them and as shown in the examples, standard finite elements in non-conforming subdomains and/or in-plane–out-of-plane techniques for layered domains with arbitrarily

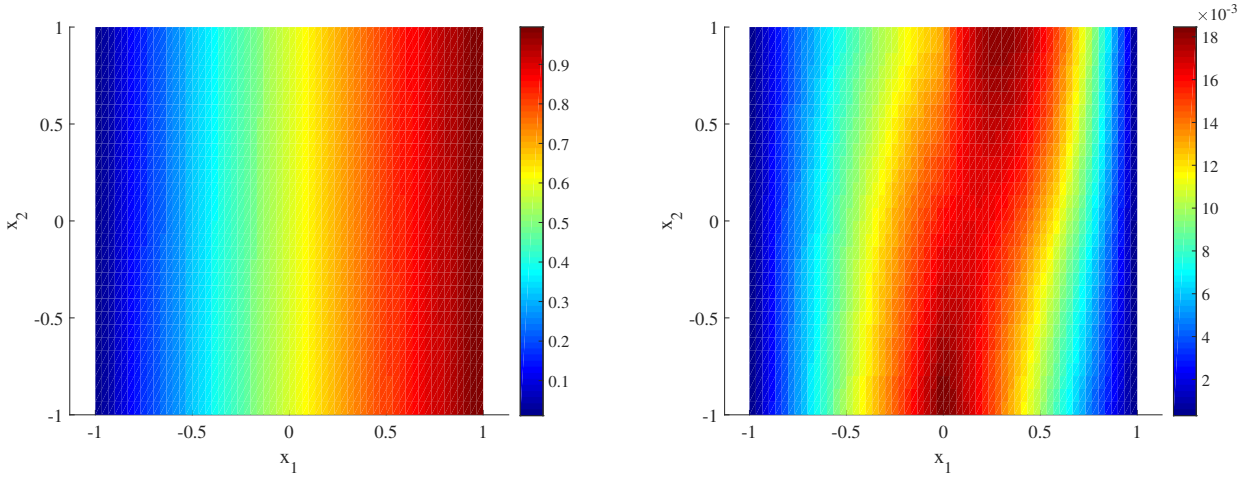

FIGURE 19 Nonlinear problem. DD-PGD solution (left) and relative error with respect to the FEM solution for a mesh of 30×30 bilinear elements (right).

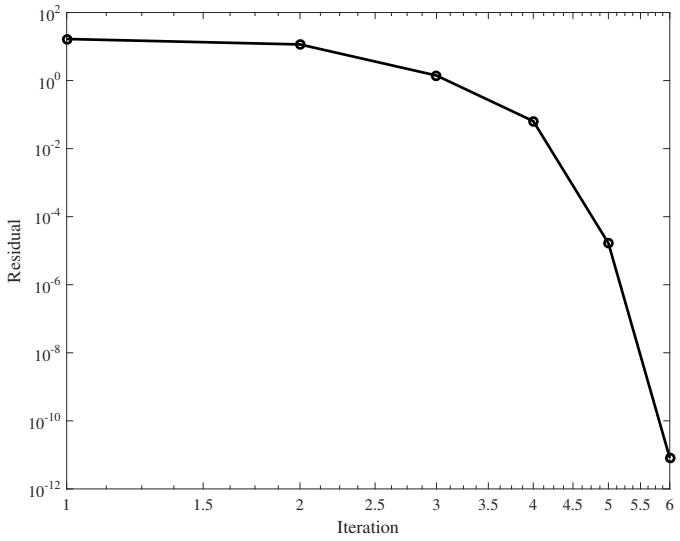

FIGURE 20 Convergence of the Newton-Raphson strategy used for solving the global problem in the nonlinear thermal heat equation.

oriented mid-surfaces. For this purpose, domain decomposition strategies and the Proper Generalized Decomposition are combined in an efficient manner. The repetitiveness of the subdomains is exploited to reduce drastically the offline computational effort. Moreover, an efficient global domain decomposition solve is also included in this offline phase to drastically reduce the cost of the online phase, which is now reduced to the explicit evaluation of the solution with no additional problem solving.

The numerical examples show the potential of the proposed methodology for practical engineering applications.

References

[1] Ammar A., Mokdad B., Chinesta F., Keunings R., A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. *J. Non-Newtonian Fluid Mech.*. 2006;139:153-176.

[2] Chinesta F., Leygue A., Bordeu F., et al. PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Arch. Comput. Method Eng.*. 2013;20:31-59.

[3] Chinesta F., Cueto E., Huerta A., PGD for solving multidimensional and parametric models. In: CISM Courses and Lectures, vol. 554: Springer, Vienna 2014 (pp. 27–89).

[4] Chinesta F., Keunings R., Leygue A., *The proper generalized decomposition for advanced numerical simulations. A primer*. Springer Briefs in Applied Sciences and TechnologySpringer, Cham; 2014.

[5] Gonzalez D., Ammar A., Chinesta F., Cueto E., Recent advances on the use of separated representations. *Int. J. Numer. Methods Eng.*. 2010;81(5):637–659.

[6] Ghnatios C., Chinesta F., Binetruy C., 3D Modeling of squeeze flows occurring in composite laminates. *Int. J. Mater. Form.*. 2015;8(1):73–83.

[7] Bognet B., Bordeu F., Chinesta F., Leygue A., Poitou A., Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity. *Comput. Methods Appl. Mech. Eng.*. 2012;201-204:1–12.

[8] Bognet B., Leygue A., Chinesta F., Separated representations of 3D elastic solutions in shell geometries. *Adv. Model. Simul. Eng. Sci.*. 2014;1(4).

[9] Ibáñez R., Abisset-Chavanne E., Chinesta F., Huerta A., Simulating squeeze flows in multiaxial laminates: towards fully 3D mixed formulations. *Int. J. Mater. Form.*. 2017;10(5):653–669.

[10] Toselli A., Widlund O., *Domain decomposition methods—algorithms and theory* Springer Series in Computational Mathematics, vol. 34: . Berlin: Springer-Verlag; 2005.

[11] Nazeer S. M., Bordeu F., Leygue A., Chinesta F., Arlequin based PGD domain decomposition. *Comput. Mech.*. 2014;54(5):1175–1190.

[12] Krause R. H., Wohlmuth B. I., A Dirichlet-Neumann type algorithm for contact problems with friction. *Comput. Vis. Sci.*. 2002;5(3):139–148.

[13] Farhat C., Roux F.-X., A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Eng.*. 1991;32(6):1205–1227.

[14] Brezzi F., Marini L. D., A three-field domain decomposition method. In: Contemp. Math., vol. 157: Providence, RI: Amer. Math. Soc. 1994 (pp. 27–34).

[15] Nitsche J. A., Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*. 1971;36:9–15.

[16] Freud J., Stenberg R., On weakly imposed boundary conditions for second order problems. In: ; 1995.

[17] Stenberg R., On some techniques for approximating boundary conditions in the finite element method. *J. Comput. Appl. Math.*. 1995;63(1-3):139–148.

[18] Becker R., Hansbo P., Stenberg R., A finite element method for domain decomposition with non-matching grids. *ESAIM-Math. Model. Numer. Anal.*. 2003;37(2):209–225.

[19] Iapichino L., Quarteroni A., Rozza G., A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *Comput. Methods Appl. Mech. Eng.*. 2012;221-222:63–82.

[20] Eftang J. L., Patera A. T., Port reduction in parametrized component static condensation: approximation and *a posteriori* error estimation. *Int. J. Numer. Methods Eng.*. 2013;96(5):269–302.

[21] Eftang J. L., Patera A. T., A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and *A Posteriori* error estimation. *Adv. Model. Simul. Eng. Sci.*. 2014;1(3).

[22] Vallaghé S., Patera A. T., The static condensation reduced basis element method for a mixed-mean conjugate heat exchanger model. *SIAM J. Sci. Comput.*. 2014;36(3):B294–B320.

[23] Martini I., Rozza G., Haasdonk B., Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-Darcy system. *Adv. Comput. Math.*. 2015;41(5):1131–1157.

[24] Smetana K., A new certification framework for the port reduced static condensation reduced basis element method. *Comput. Methods Appl. Mech. Eng.*. 2015;283:352–383.

[25] Smetana K., Patera A. T., Optimal local approximation spaces for component-based static condensation procedures. *SIAM J. Sci. Comput.*. 2016;38(5):A3318–A3356.

[26] Iapichino L., Quarteroni A., Rozza G., Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries. *Comput. Math. Appl.*. 2016;71(1):408–430.

[27] Maday Y., Rønquist E. M., A reduced-basis element method. *J. Sci. Comput.*. 2002;17(1-4):447–459.

[28] Huynh D. B. P., Knezevic D. J., Patera A. T., A static condensation reduced basis element method: approximation and *a posteriori* error estimation. *ESAIM-Math. Model. Numer. Anal.*. 2013;47(1):213–251.

[29] Ammar A., Huerta A., Chinesta F., Cueto E., Leygue A., Parametric solutions involving geometry: a step towards efficient shape optimization. *Comput. Methods Appl. Mech. Eng.*. 2014;268:178–193.

[30] Zlotnik S., Díez P., Modesto D., Huerta A., Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *Int. J. Numer. Methods Eng.*. 2015;103(10):737–758.

[31] Montlaur A., Fernández-Méndez S., Huerta A., Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *Int. J. Numer. Methods Fluids.* 2008;57(9):1071–1092.

[32] Ciarlet P. G., *The finite element method for elliptic problems* Classics in Applied Mathematics, vol. 40: . Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; 2002. Reprint of the 1978 original [North-Holland, Amsterdam].

[33] Szabó B., Babuška I., *Finite element analysis: formulation, verification, and validation*. Wiley Series in Computational MathematicsChichester: John Wiley & Sons, Inc.; 2011.

[34] Patera A. T., Rozza G., *Reduced Basis Approximation and A-Posteriori Error Estimation for Parametrized Partial Differential Equations.* MIT Pappalardo Graduate Monographs in Mechanical Engineering: Massachusetts Institute of TechnologyCambridge, MA, USA; 2007.

[35] Rozza G., Fundamentals of reduced basis method for problems governed by parametrized PDEs and applications. In: CISM Courses and Lectures, vol. 554: Springer, Vienna 2014 (pp. 153–227).

[36] Zlotnik S., Diez P., González D., Cueto E., Huerta A., Effect of the separated approximation of input data in the accuracy of the resulting PGD solution. *Adv. Model. Simul. Eng. Sci.*. 2015;2(28).

[37] Maday Y., Rønquist E. M., The Reduced Basis Element Method: Application to a Thermal Fin Problem. *SIAM J. Sci. Comput.*. 2004;26(1):240–258.

How cite this article: Huerta, A., Nadal, E., and Chinesta, F. (2017), The Proper Generalized Decomposition within a Domain Decomposition strategy, *Int. J. Numer. Methods Eng.*, *2017;00:1–15*.

APPENDIX A: DIRICHLET CONDITIONS AS ESSENTIAL BOUNDARY CONDITIONS

The local problem in each subdomain type can also be solved imposing Dirichlet boundary conditions as essential ones. In this case, the weak problem associated to (3) is: for $i = 1, \dots, n_{\mathtt{dd}}$ find $u_i \in \mathcal{H}^1_0(\Omega_i) + \{u_D\} + \{\hat{u}\}$ such that

$$a_i(u_i, w) = \ell_i(w) \quad \forall w \in \mathcal{H}^1_0(\Omega_i) \tag{A1a}$$

with

$$a_i(v, w) = \int_{\Omega_i} \nabla w \cdot (\mathbf{K}_i \nabla v)\, d\Omega \quad \text{and} \quad \ell_i(w) = \int_{\Omega_i} w\, f\, d\Omega + \int_{\partial\Omega_i \cap \Gamma_N} w\, t\, d\Gamma, \tag{A1b}$$

where $\mathcal{H}^1_0(\Omega_i) := \{v \in \mathcal{H}^1(\Omega_i) : v|_{\partial\Omega_i \setminus \Gamma_N} = 0\}$ and $\mathcal{H}^1$ is the standard Sobolev space of square integrable functions and first derivatives.

This approach, however, is a little more cumbersome for the PGD technique since it requires the evaluation of the first mode in a more specific manner. It is standard in PGD [3, 4] to impose the essential Dirichlet boundary conditions in the first mode and then solve the other modes with homogeneous Dirichlet boundary conditions.

Here, however, the Dirichlet boundary conditions on the interfaces are characterized as a set of parameters that PGD sees as new coordinates. This implies also a specific strategy for the first mode whereas all the others are, as usual, solved with homogeneous conditions.

In fact the strategy only affects the first spatial modes. In fact, the same separated representation proposed in (9) is used. Two extra terms are added, one to take care of the imposed Dirichlet conditions in Γ_D and another to account for the parameters $\hat{\mathrm{u}}$. The former is standard and requires solving the separated problem with homogeneous conditions everywhere except for $\partial\Omega_i \cap \Gamma_D$ to compute $F^{u_D}_{x_1}(x_1)$ and $F^{u_D}_{x_2}(x_2)$. The second term is the one associated to each parameter $\hat{\mathrm{u}}_s$ for $s = 1, \dots, \hat{n}$, see (6). It consists in solving for the spatial modes $F^{u_s}_{x_1}(x_1)$ and $F^{u_s}_{x_2}(x_2)$ as before but with homogeneous conditions everywhere except for $\partial\Omega_i \cap \Gamma \cap \mathrm{supp}(\widehat{N}_s)$ where the condition $u = \widehat{N}_s$ is imposed.

Recalling the expression (9) the PGD approximation for each subdomain introduces these new terms and becomes

$$u^n_{\mathtt{PGD}}(\boldsymbol{x}, \hat{\mathbf{u}}) = F^{u_D}_{x_1}(x_1)\, F^{u_D}_{x_2}(x_2) + \sum_{s=1}^{\hat{n}} \hat{\mathrm{u}}_s\, F^{u_s}_{x_1}(x_1)\, F^{u_s}_{x_2}(x_2) + \sum_{m=1}^{n} \left[F^m_{x_1}(x_1)\, F^m_{x_2}(x_2) \prod_{s=1}^{\hat{n}} \widehat{F}^m_{\hat{\mathrm{u}}_s}(\hat{\mathrm{u}}_s) \right]$$

This strategy, requires to interpolate the shape functions $\widehat{N}_s$ for the interface variables $\hat{u}$ on the finite element mesh used in the subdomain and special care must be taken of the corners of the subdomain in order to avoid duplicity in the imposed boundary condition.