# Proper Generalized Decomposition of a geometrically parametrized heat problem with geophysical applications

Sergio Zlotnik, Pedro Díez, David Modesto and Antonio Huerta[*]

*Laboratori de Càlcul Numèric (LaCàN).*
*E.T.S. de Ingenieros de Caminos, Canales y Puertos,*
*Universitat Politècnica de Catalunya, BarcelonaTech, E-08034 Barcelona, Spain.*
*e-mail: {sergio.zlotnik, pedro.diez, david.modesto, antonio.huerta}@upc.edu, web http://www.lacan.upc.edu*

## SUMMARY

The solution of a steady thermal multiphase problem is assumed to be dependent on a set of parameters describing the geometry of the domain, the internal interfaces and the material properties. These parameters are considered as new independent variables. The problem is therefore stated in a multidimensional setup. The Proper Generalized Decomposition (PGD) provides an approximation scheme especially well suited to preclude dramatically increasing the computational complexity with the number of dimensions. The PGD strategy is reviewed for the standard case dealing only with material parameters. Then, the ideas presented in [Ammar et al., "Parametric solutions involving geometry: A step towards efficient shape optimization." *Comput. Methods Appl. Mech. Eng.*, 2014; **268**:178–193] to deal with parameters describing the domain geometry are adapted to a more general case including parametrization of the location of internal interfaces. Finally, the formulation is extended to combine the two types of parameters. The proposed strategy is used to solve a problem in applied geophysics studying the temperature field in a cross section of the Earth crust subsurface. The resulting problem is in a 10-dimensional space, but the PGD solution provides a fairly accurate approximation (error $\leq 1\%$) using less that 150 terms in the PGD expansion.

KEY WORDS: Reduced order model, Proper Generalized Decomposition (PGD), Geometry parametrization, Interface, Thermal cross section, Geophysics, Inverse problem

## 1. INTRODUCTION

Many problems with practical application can be characterized by its parametric dependence. The general solution is sought in a multidimensional parametric space, with an explicit expression depending on the parameters as new independent variables (or coordinates). This strategy increases

the dimensionality of the problem (e.g. if you add three free parameters in a 2D spatial problem, the overall dimension is five) and it may lead to an explosion in the computational complexity often referred as the *curse of dimensionality*.

The Proper Generalized Decomposition (PGD) [1–3] is a Reduced Order Model technique particularly devised to preclude the curse of dimensionality. The PGD is first used in an offline phase computing the elements characterizing the general solution containing the parametric dependence (also denoted as *computational vademecum* [4]). Then, the second phase consists in multiple online queries in which real-time responses are achieved.

Inverse problems are one of the scenarios were PGD can be successfully utilized. These problems are employed to find the values of the parameters that better fit the available observations (experimental measurements). This approach requires solving a number times the *direct* model, each time for a different data set and selecting the set of parameters such that the computed values coincide with (or better fit) the observations. The ability to rapidly solve the direct models could represent the difference between computationally affordable and non-affordable inverse problems.

The problems considered in this paper include the parametric representation of the geometry and the shape and location of the internal interfaces. For instance, the geometry parameters describe the size of the domain, the thickness of a layer, the depth, inclination and curvature of interfaces... This type of problems, pertaining to the geometry optimization, have been addressed by many authors in standard contexts, see [5–11].

Here, the PGD is used to obtain a computational vademecum accounting for the parameters representing the geometry and the location of the interfaces. The proposed methodology is based on the ideas recently introduced in [12] to deal with the geometric parametrization. The novel contributions of this work consist in 1) the extension accounting for internal interfaces, 2) the combination of the geometry parameters with other parameters describing the material properties and boundary conditions, 3) the methodology allowing a general geometrical description by constructing separated representations of the elements introducing in the equation the effect of the geometrical variations (without significant loss of accuracy), and 4) the modular computation of the PGD modes pre-storing the necessary matrices and allowing an efficient implementation.

The remainder of the paper is structured as follows: Section 2 presents the problem statement and introduces, with some detail, the PGD technique in the simpler case of the diffusivity-parametrized Poisson equation. In Section 3 the proposed treatment of the geometrical parameters is presented and together with the separable version of the operators involved. The general PGD solution for the diffusivity and geometry parametrized problem is briefly described in Section 4 and, finally, an example of the proposed technique with application to a temperature distribution in a cross section in the subsurface of the Earth Crust is presented in Section 5.

## 2. PROBLEM STATEMENT AND PGD SOLUTION FOR THE DIFFUSIVITY-PARAMETRIZED POISSON EQUATION

### 2.1. Poisson equation with two materials

The function to be approximated is the solution $u$, taking values in $\Omega$, of the following boundary value problem governed by a Poisson equation. The strong form reads: find $u$ such that,

$$\begin{cases} -\nabla \cdot (\kappa \nabla u) = s & \text{in } \Omega, \\ (\kappa \nabla u) \cdot \boldsymbol{n} = g_N & \text{on } \Gamma_N, \\ u = u_D & \text{on } \Gamma_D, \end{cases} \tag{1}$$

where the source term $s$, the prescribed values on the Dirichlet boundary $u_D$, the prescribed flux on the Neumann value $g_N$ and the diffusivity $\kappa$ are the data set. The domain $\Omega$ is partitioned in two subdomains $\Omega_1$ and $\Omega_2$, such that the diffusivity function $\kappa$ is defined as

$$\kappa(\boldsymbol{x}) = \begin{cases} \kappa_1 & \text{for } \boldsymbol{x} \in \Omega_1, \\ \kappa_2 & \text{for } \boldsymbol{x} \in \Omega_2. \end{cases} \tag{2}$$

The diffusivity function $\kappa$ takes a constant value in each subdomain. However, it is straightforward to generalize the methodologies introduced in the next sections to more complex functions (characterized also by a set of parameters, for instance two as $\kappa_1$ and $\kappa_2$). Geometrically, $\Omega_1$ and $\Omega_2$ are complementary in $\Omega$, that is $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$, and $\partial\Omega_1 \cap \partial\Omega_2$ is the interface between the two open subdomains.

The usual variational form of the problem reads: find $u \in V$ such that

$$a(u, v) = \ell(v), \text{ for all } v \in V_0, \tag{3}$$

where $V := \{u \in \mathcal{H}^1(\Omega) : u = u_D \text{ in } \Gamma_D\}$ and its corresponding test functions space is $V_0 := \{u \in \mathcal{H}^1(\Omega) : u = 0 \text{ on } \Gamma_D\}$. The bilinear and linear forms $a(\cdot, \cdot)$ and $\ell(\cdot)$ are given by

$$a(u, v) := \int_{\Omega_1 \cup \Omega_2} \nabla u \cdot (\kappa \nabla v) \, d\Omega \quad \text{and} \quad \ell(v) := \int_{\Omega} sv \, d\Omega + \int_{\Gamma_N} g_N v \, ds. \tag{4}$$

*Remark 1* (Homegneous Neumann boundary conditions)
For the sake of a simple presentation and without any loss of generality, it is assumed hereafter that Neumann boundary conditions are homogeneous, that is $g_N = 0$. The treatment of non-homogenous Neumann conditions in the PGD context is described, for example, in [13].

### 2.2. Parametric problem

In order to compute a generalized solution corresponding to the different values of the set of parameters $\kappa_1$ and $\kappa_2$, problem (1) is solved taking $\kappa_1$ and $\kappa_2$ as additional independent variables (or parametric coordinates) and the unknown is formally written as $u(\boldsymbol{x}, \kappa_1, \kappa_2)$, being $\boldsymbol{x} \in \Omega$, $\kappa_1 \in I_{\kappa_1}$ and $\kappa_2 \in I_{\kappa_2}$. Thus, in (1) $u$ takes values in $\mathcal{D} = \Omega \times I_{\kappa_1} \times I_{\kappa_2}$ and formally, also $\kappa$ depends on $(\boldsymbol{x}, \kappa_1, \kappa_2)$ as indicated in (2).

Note that for a 2D spatial computational domain $\Omega$ functions $u$ and $\kappa$ are defined in the 4D space $\mathcal{D}$. Thus, formally, the solution lies in a tensor product space, namely $u \in V \otimes \mathcal{L}^2(I_{\kappa_1}) \otimes \mathcal{L}^2(I_{\kappa_2})$. In other words, freezing two of the arguments, the remaining function belongs to the corresponding sectorial space, that is

$$\text{for given values of } \kappa_1, \kappa_2; \ u(\cdot, \kappa_1, \kappa_2) \in V,$$
$$\text{for given values of } \boldsymbol{x}, \kappa_2; \ u(\boldsymbol{x}, \cdot, \kappa_2) \in \mathcal{L}^2(I_{\kappa_1}),$$
$$\text{for given values of } \boldsymbol{x}, \kappa_1; \ u(\boldsymbol{x}, \kappa_1, \cdot) \in \mathcal{L}^2(I_{\kappa_2}).$$

Note that the tensor product space $V \otimes \mathcal{L}^2(I_{\kappa_1}) \otimes \mathcal{L}^2(I_{\kappa_2})$ collects the multidimensional complexity of the problem and does not assume separability of the functions: if the three spaces have finite dimensions, the dimension of the tensor product space is the product of the three dimensions.

The integration is now in $\mathcal{D}$ and therefore in a multi-dimensional space. The derivation of (5) is performed using the standard weighted residuals technique integrating in $\mathcal{D}$ but requires standard integration by parts only in $\Omega$. The corresponding weak form in the multi-dimensional setup reads: find $u \in V \otimes \mathcal{L}^2(I_{\kappa_1}) \otimes \mathcal{L}^2(I_{\kappa_2})$ such that

$$A(u, v) = L(v), \ \text{ for all } v \in V_0 \otimes \mathcal{L}^2(I_{\kappa_1}) \otimes \mathcal{L}^2(I_{\kappa_2}), \tag{5}$$

with the following definitions of the bilinear and linear forms

$$A(u, v) := \int_{I_{\kappa_1}} \int_{I_{\kappa_2}} a(u, v) \, d\kappa_2 \, d\kappa_1 \ \text{ and } \ L(v) := \int_{I_{\kappa_1}} \int_{I_{\kappa_2}} \ell(v) \, d\kappa_2 \, d\kappa_1. \tag{6}$$

The number of dimensions of the solution domain increases with the number of parameters used in the problem characterization. The traditional numerical techniques (standard finite elements, finite volumes or finite differences) fail in providing an affordable methodology because the computational cost blows up exponentially with multidimensionality. Handling multidimensional functions raises the issue of precluding the *curse of dimensionality*. The PGD approach to tackle this problem consists in enforcing a separable structure in the function that has to approximate $u$.

### 2.3. Separable approximation

The PGD methodology assumes that $u(\boldsymbol{x}, \kappa_1, \kappa_2)$ can be fairly approximated by a sum of separated terms, namely

$$u(\boldsymbol{x}, \kappa_1, \kappa_2) \approx u_{\text{PGD}}^n(\boldsymbol{x}, \kappa_1, \kappa_2) = \sum_{m=1}^{n} F_x^m(\boldsymbol{x}) F_{\kappa_1}^m(\kappa_1) F_{\kappa_2}^m(\kappa_2)$$
$$= u_{\text{PGD}}^{n-1}(\boldsymbol{x}, \kappa_1, \kappa_2) + F_x^n(\boldsymbol{x}) F_{\kappa_1}^n(\kappa_1) F_{\kappa_2}^n(\kappa_2). \tag{7}$$

The function $u_{\text{PGD}}^n$ 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 of one of the arguments. 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.

Hereafter, the explicit dependence of the functions is dropped out in the notation because it is already indicated in the subscript. For example, $F_x^m$ stands also for $F_x^m(\boldsymbol{x})$ and it is assumed that it depends only on $\boldsymbol{x}$.

Derivation of the PGD algorithm requires assuming that the bilinear and linear operators, $A(\cdot,\cdot)$ and $L(\cdot)$, are also separable. This is equivalent to say that the functions involved in the operators, in this case, the diffusivity function $\kappa(\boldsymbol{x},\kappa_1,\kappa_2)$ and the source term $s(\boldsymbol{x},\kappa_1,\kappa_2)$, have both a separable structure similar to $u_{\mathrm{PGD}}(\boldsymbol{x},\kappa_1,\kappa_2)$, namely

$$\kappa(\boldsymbol{x},\kappa_1,\kappa_2) \approx \kappa_{\mathrm{PGD}}(\boldsymbol{x},\kappa_1,\kappa_2) = \sum_{k=1}^{n_\kappa} B_x^k B_{\kappa_1}^k B_{\kappa_2}^k \tag{8a}$$

$$s(\boldsymbol{x},\kappa_1,\kappa_2) \approx s_{\mathrm{PGD}}(\boldsymbol{x},\kappa_1,\kappa_2) = \sum_{l=1}^{n_s} S_x^l S_{\kappa_1}^l S_{\kappa_2}^l. \tag{8b}$$

The functions $S_\star^l$ and $B_\star^k$ (for $\star = \boldsymbol{x}$, $\kappa_1$ and $\kappa_2$, $k = 1,\ldots,n_\kappa$, $l = 1,\ldots,n_s$) describe the source term $s$ and the diffusivity $\kappa$ that are both part of the data. The separable representation (or approximation) of $s$ and $\kappa$ must be obtained as a preprocess.

Note that despite the dependence of the source term $s$ on the diffusivities might seem atypical, in other circumstances the source term could depend on the parameters. Thus, to present the general situation this dependence is explicitly kept.

*Remark 2* (Separability of diffusivity)

Note that the diffusivity function, $\kappa$, as given in (2), admits an exact separable form based on Heaviside like functions $H^i(\boldsymbol{x}) : \Omega \to \mathbb{R}$ defined as,

$$H^i(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} \in \Omega_i, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the separable form of the $\kappa(\boldsymbol{x},\kappa_1,\kappa_2)$ is,

$$\kappa(\boldsymbol{x},\kappa_1,\kappa_2) = H^1(\boldsymbol{x})\,\kappa_1 + H^2(\boldsymbol{x})\,\kappa_2. \tag{9}$$

This equation has the structure described by (8b), taking $B_x^1(\boldsymbol{x}) = H^1(\boldsymbol{x})$, $B_{\kappa_1}^1(\kappa_1) = \kappa_1$ and $B_{\kappa_2}^1(\kappa_2) = 1$, and similarly for the second term. There is no additional difficulty in extending this strategy to any number of different space subdomains larger than two, taking $n_\kappa$ equal to the number of subdomains.

If the separate representation is not as trivial as in the previous case, the Proper Orthogonal Decomposition (POD) can be used to obtain an approximated separable representation. For instance, this is to be applied in the general case to obtain an expression like (8b) for function $s$ defining the source term. The use of the standard POD is restricted to the case of having only two parameters. More than two parameters require using a High Order Singular Value Decomposition (HO-SVD) [14, 15].

## 2.4. Greedy algorithm: independent computation of terms for the PGD expansion

The number of terms required in the PGD approximation, $n$ in equation (7), is, in general, not known a priori. One of the assets of the PGD strategy is to compute the $n$ terms sequentially, using a greedy approach. The number of terms is set by stopping the computation whenever the obtained solution is trusted to be accurate enough. Goal oriented error assessment can also be used in this strategy [16]. Note also that each new term in (7) is computed using an alternating directions iterative scheme with a 2D complexity.

Once the previous terms $u_{\mathrm{PGD}}^{n-1}$ are known, the unknowns that have to be determined to compute $u_{\mathrm{PGD}}^n$ are functions $F_x$, $F_{\kappa_1}$, and $F_{\kappa_2}$ (the dependence on $n$ displayed in (7) is now omitted in the notation because attention is concentrated in the $n$ term and therefore there is no ambiguity). These functions are sought in the spaces $V$, $\mathcal{L}^2(I_{\kappa_1})$ and $\mathcal{L}^2(I_{\kappa_2})$, and the corresponding test functions belong to $V_0$, $\mathcal{L}^2(I_{\kappa_1})$ and $\mathcal{L}^2(I_{\kappa_2})$. The weak problem presented in (5) determines the last term of $u_{\mathrm{PGD}}^n$. Thus, the objective now is to evaluate the separated contributions $F_x \in V$, $F_{\kappa_1} \in \mathcal{L}^2(I_{\kappa_1})$ and $F_{\kappa_2} \in \mathcal{L}^2(I_{\kappa_2})$ for all $\delta F_x \in V_0$, $\delta F_{\kappa_1} \in \mathcal{L}^2(I_{\kappa_1})$ and $\delta F_{\kappa_2} \in \mathcal{L}^2(I_{\kappa_2})$ such that

$$A(F_x F_{\kappa_1} F_{\kappa_2}, v) = L(v) - A(u_{\mathrm{PGD}}^{n-1}, v) \text{ with } v = \delta F_x F_{\kappa_1} F_{\kappa_2} + F_x \delta F_{\kappa_1} F_{\kappa_2} + F_x F_{\kappa_1} \delta F_{\kappa_2}. \quad (10)$$

The problem is nonlinear because each equation (for a given $v$) involves the product of the unknown functions and the test set for $v$ is characterized in terms of the solution itself (increasing the nonlinear character of the problem). Moreover, the test functions $v$ belong to $V_0 \otimes \mathcal{L}^2(I_{\kappa_1}) \otimes \mathcal{L}^2(I_{\kappa_2})$ such that they are written as $v = \delta F_x F_{\kappa_1} F_{\kappa_2} + F_x \delta F_{\kappa_1} F_{\kappa_2} + F_x F_{\kappa_1} \delta F_{\kappa_2}$. This functional set is not a vectorial space and therefore the solution of (10) cannot be interpreted as a Galerkin projection. However, it can be seen as a relaxation of problem (5) in order to obtain as many equations as unknowns and force to vanish the residual in this particular functional set where the form of the test function is understood as a variation of the separable unknown, namely

$$v = \delta(F_x F_{\kappa_1} F_{\kappa_2}) = \underbrace{\delta F_x\, F_{\kappa_1} F_{\kappa_2}}_{v_x} + \underbrace{F_x \delta F_{\kappa_1}\, F_{\kappa_2}}_{v_{\kappa_1}} + \underbrace{F_x F_{\kappa_1} \delta F_{\kappa_2}}_{v_{\kappa_2}} = v_x + v_{\kappa_1} + v_{\kappa_2}.$$

As proved by by Falcó and Nouy , see [17], the nonlinear problem (10) is well posed and the fixed-point iteration method has guaranteed convergence for the case of the elliptic problems.

In the remainder of the presentation, the developments are presented for the discrete spaces. The discrete FE test spaces are denoted by $V_x^h$, $V_{\kappa_1}^h$ and $V_{\kappa_2}^h$, respectively. In the discrete setup, taking both the candidate solutions and the test functions in the discrete spaces problem (10) is formulated as: find the discrete separated contributions $F_x$, $F_{\kappa_1}$ and $F_{\kappa_2}$ such that

$$A(F_x F_{\kappa_1} F_{\kappa_2}, v_\star) = L(v_\star) - A(u_{\mathrm{PGD}}^{n-1}, v_\star) \begin{cases} \text{for all } \delta F_\star \in V_\star^h, \\ \text{with } v_\star = \delta F_\star F_x F_{\kappa_1} F_{\kappa_2} / F_\star, \\ \text{for } \star = x, \kappa_1 \text{ and } \kappa_2. \end{cases} \quad (11)$$

The iterative solver is devised using the idea of computing a new iteration $F_x$, $F_{\kappa_1}$ and $F_{\kappa_2}$ as a correction of the previous one $F_x$, $F_{\kappa_1}$ and $F_{\kappa_2}$ and assuming that the iteration scheme will lead

to convergence towards the solution of (11). Next section is devoted to present a fixed-point solver with an alternating directions strategy to be used in this context.

*Remark 3* (Reduction in number of unknowns)

The number of unknowns involved in finding $u_{\mathrm{PGD}}^n$ (or $F_x$, $F_{\kappa_1}$ and $F_{\kappa_2}$) in (11) is $d_x + d_{\kappa_1} + d_{\kappa_2}$. This corresponds to the sum of the dimensions of the spaces $V_x^h$, $V_{\kappa_1}^h$ and $V_{\kappa_2}^h$. The choice for the set of test functions states the same number of equations. These equations are however nonlinear. This is the price to pay for reducing the computational complexity of the original problem statement, see problem (5). Recall that in the original problem test functions range in $V_x^h \otimes V_{\kappa_1}^h \otimes V_{\kappa_2}^h$ and , consequently, have $d_x d_{\kappa_1} d_{\kappa_2}$ degrees of freedom and equations.

### 2.5. Fixed-point solver for the nonlinear problem

Following previous works, see for example [16, 18–20], an alternating directions scheme is sequentially applied to the non-linear problem (11). The alternating directions approach consists in solving each dimension separately, assuming that the other components are known. The idea is similar to a block-based Gauss-Seidel scheme.

Using this idea (11) is split in three sequential linear equations, the first problem reads: given $F_{\kappa_1}$ and $F_{\kappa_2}$, update $F_x$, such that

$$A(F_x F_{\kappa_1} F_{\kappa_2}, v_x) = L(v_x) - A(u_{\mathrm{PGD}}^{n-1}, v_x) \text{ for all } \delta F_x \in V_x^h \text{ with } v_x = \delta F_x F_{\kappa_1} F_{\kappa_2}. \quad (12a)$$

Then, the second problem reads : given $F_x$, and $F_{\kappa_2}$, update $F_{\kappa_1}$, such that

$$A(F_x F_{\kappa_1} F_{\kappa_2}, v_{\kappa_1}) = L(v_{\kappa_1}) - A(u_{\mathrm{PGD}}^{n-1}, v_{\kappa_1}) \text{ for all } \delta F_{\kappa_1} \in V_{\kappa_1}^h \text{ with } v_{\kappa_1} = F_x \delta F_{\kappa_1} F_{\kappa_2}. \quad (12b)$$

And, finally, the third problem reads: given $F_x$ and $F_{\kappa_1}$, update $F_{\kappa_2}$, such that

$$A(F_x F_{\kappa_1} F_{\kappa_2}, v_{\kappa_2}) = L(v_{\kappa_2}) - A(u_{\mathrm{PGD}}^{n-1}, v_{\kappa_2}) \text{ for all } \delta F_{\kappa_2} \in V_{\kappa_2}^h \text{ with } v_{\kappa_2} = F_x F_{\kappa_1} \delta F_{\kappa_2}. \quad (12c)$$

Note that problems (12) are linear because the unknowns are $F_x$, $F_{\kappa_1}$ and $F_{\kappa_2}$ respectively for each of them. The method proposed consists in iterating the three equations to reach a stationary solution. The converged solution is assumed to fulfill equation (11).

### 2.6. Matrix form of the resulting linear equations

As previously said, each of the problems described by (12) is linear. This section is devoted to detail the matrix form of these equations. With the definitions of (4), (6) and (8a), it turns out that

$$A(u, v) = \sum_{k=1}^{n_\kappa} \int_{I_{\kappa_1}} \int_{I_{\kappa_2}} \int_{\Omega_1 \cup \Omega_2} \nabla u \cdot (B_x^k B_{\kappa_1}^k B_{\kappa_2}^k \nabla v) \, d\Omega \, d\kappa_2 \, d\kappa_1 \quad (13)$$

Recalling (12), where both arguments of $A(\cdot, \cdot)$ $u$ and $v$ are separable, say $u = F_x F_{\kappa_1} F_{\kappa_2}$ and $v = \tilde{F}_x \tilde{F}_{\kappa_1} \tilde{F}_{\kappa_2}$, (13) becomes

$$
\begin{aligned}
A(u, v) &= \sum_{k=1}^{n_\kappa} \left[ \int_{\Omega_1 \cup \Omega_2} \nabla F_x \cdot (B_x^k \nabla \tilde{F}_x) \, d\Omega \right] \left[ \int_{I_{\kappa_1}} F_{\kappa_1} B_{\kappa_1}^k \tilde{F}_{\kappa_1} d\kappa_1 \right] \left[ \int_{I_{\kappa_2}} F_{\kappa_2} B_{\kappa_2}^k \tilde{F}_{\kappa_2} d\kappa_2 \right] \\
&= \sum_{k=1}^{n_\kappa} a^k(F_x, \tilde{F}_x) \; m_{\kappa_1}^k(F_{\kappa_1}, \tilde{F}_{\kappa_1}) \; m_{\kappa_2}^k(F_{\kappa_2}, \tilde{F}_{\kappa_2}),
\end{aligned}
\tag{14}
$$

where the bilinear forms $a^k(\cdot, \cdot)$, $m_{\kappa_1}^k(\cdot, \cdot)$ and $m_{\kappa_2}^k(\cdot, \cdot)$ are defined in Appendix A.

The FE discrete approximations of the functions depending on each of the independent variables, that is for $x$, $\kappa_1$ and $\kappa_2$ are expressed in terms of the interpolation functions constituting basis for the spaces $V_x^h$, $V_{\kappa_1}^h$ and $V_{\kappa_2}^h$. The basis functions are readily collected in the vectors of shape functions, namely $\boldsymbol{N}_x^{\mathsf{T}} = [N_x^1, N_x^2, \ldots, N_x^{d_x}]$, $\boldsymbol{N}_{\kappa_1}^{\mathsf{T}} = [N_{\kappa_1}^1, N_{\kappa_1}^2, \ldots, N_{\kappa_1}^{d_{\kappa_1}}]$ and $\boldsymbol{N}_{\kappa_2}^{\mathsf{T}} = [N_{\kappa_2}^1, N_{\kappa_2}^2, \ldots, N_{\kappa_2}^{d_{\kappa_2}}]$.

Thus, any function depending on, for example, $x$ is readily expressed in terms of a vector of $d_x$ nodal values, namely

$$
F_x^m = \boldsymbol{N}_x^{\mathsf{T}} \boldsymbol{f}_x^m
$$

where vector $\boldsymbol{f}_x^m$ collects the nodal values of $F_x^m$ in the nodes defining the FE discretization associated with the basis $\{N_x^1, N_x^2, \ldots, N_x^{d_x}\}$. Similarly for $F_{\kappa_1}^m = \boldsymbol{N}_{\kappa_1}^{\mathsf{T}} \boldsymbol{f}_{\kappa_1}^m$ and $F_{\kappa_2}^m = \boldsymbol{N}_{\kappa_2}^{\mathsf{T}} \boldsymbol{f}_{\kappa_2}^m$.

The matrices associated with the bilinear forms $a^k(\cdot, \cdot)$, $m_{\kappa_1}^k(\cdot, \cdot)$ and $m_{\kappa_2}^k(\cdot, \cdot)$ in the discrete spaces are denoted by $\mathbf{K}_x^k$, $\mathbf{M}_{\kappa_1}^k$ and $\mathbf{M}_{\kappa_2}^k$, respectively, as detailed in Appendix A.

A similar approach is followed with the right-hand-side term.

$$
\begin{aligned}
L(v) &= \sum_{l=1}^{n_s} \int_{I_{\kappa_1}} \int_{I_{\kappa_2}} \int_{\Omega_1 \cup \Omega_2} (S_x^l S_{\kappa_1}^l S_{\kappa_2}^l) v \, d\Omega \, d\kappa_2 \, d\kappa_1 \\
&= \sum_{l=1}^{n_s} \left[ \int_{\Omega_1 \cup \Omega_2} S_x^l \tilde{F}_x \, d\Omega \right] \left[ \int_{I_{\kappa_1}} S_{\kappa_1}^l \tilde{F}_{\kappa_1} \, d\kappa_1 \right] \left[ \int_{I_{\kappa_2}} S_{\kappa_2}^l \tilde{F}_{\kappa_2} \, d\kappa_2 \right],
\end{aligned}
\tag{15}
$$

where, as before, functions $S_x^l$, $S_{\kappa_1}^l$ and $S_{\kappa_2}^l$ are written in terms of nodal functions and values, namely, $S_x^l = \boldsymbol{N}_x^{\mathsf{T}} \boldsymbol{s}_x^l$, $S_{\kappa_1}^l = \boldsymbol{N}_{\kappa_1}^{\mathsf{T}} \boldsymbol{s}_{\kappa_1}^l$ and $S_{\kappa_2}^l = \boldsymbol{N}_{\kappa_2}^{\mathsf{T}} \boldsymbol{s}_{\kappa_2}^l$. Recall also that the term associated with $g_N$ is assumed to vanish. Further, the load vectors, corresponding to the different terms in the previous equation are denoted by $\boldsymbol{q}_x^l$, $\boldsymbol{q}_{\kappa_1}^l$ and $\boldsymbol{q}_{\kappa_2}^l$, as described in Appendix A.

This allows rewriting (12a) in matrix form: given $\boldsymbol{f}_x^m$, $\boldsymbol{f}_{\kappa_1}^m$ and $\boldsymbol{f}_{\kappa_2}^m$ for $m = 1, \ldots, n-1$ (characterizing $u_{\text{PGD}}^{n-1}$), $\boldsymbol{f}_{\kappa_1}$ (representing $F_{\kappa_1}$) and $\boldsymbol{f}_{\kappa_2}$ (representing $F_{\kappa_2}$), update $\boldsymbol{f}_x$ (representing $F_x$) such that

$$
\left[ \sum_{k=1}^{n_\kappa} c_x^k \mathbf{K}_x^k \right] \boldsymbol{f}_x = \boldsymbol{r}_x
\tag{16a}
$$

where the constant $c_x^k$ and the vector $\boldsymbol{r}_x$ are defined as

$$c_x^k := (\boldsymbol{f}_{\kappa_1}^\mathsf{T} \mathbf{M}_{\kappa_1}^k \boldsymbol{f}_{\kappa_1})(\boldsymbol{f}_{\kappa_2}^\mathsf{T} \mathbf{M}_{\kappa_2}^k \boldsymbol{f}_{\kappa_2}),$$

$$\boldsymbol{r}_x := \sum_{l=1}^{n_s} (\boldsymbol{f}_{\kappa_1}^\mathsf{T} \boldsymbol{q}_{\kappa_1}^l)(\boldsymbol{f}_{\kappa_2}^\mathsf{T} \boldsymbol{q}_{\kappa_2}^l) \boldsymbol{q}_x^l - \sum_{m=1}^{n-1} \sum_{k=1}^{n_\kappa} c_x^{k,m} \mathbf{K}_x^k \boldsymbol{f}_x^m, \text{ and}$$

$$c_x^{k,m} := (\boldsymbol{f}_{\kappa_1}^\mathsf{T} \mathbf{M}_{\kappa_1}^k \boldsymbol{f}_{\kappa_1}^m)(\boldsymbol{f}_{\kappa_2}^\mathsf{T} \mathbf{M}_{\kappa_2}^k \boldsymbol{f}_{\kappa_2}^m).$$

Similarly, problem (12b) becomes: given $\boldsymbol{f}_x^m$, $\boldsymbol{f}_{\kappa_1}^m$ and $\boldsymbol{f}_{\kappa_2}^m$ for $m = 1, \ldots, n-1$ (characterizing $u_{\text{PGD}}^{n-1}$), $\boldsymbol{f}_x$ (representing $F_x$) and $\boldsymbol{f}_{\kappa_2}$ (representing $F_{\kappa_2}$), update $\boldsymbol{f}_{\kappa_1}$ (representing $F_{\kappa_1}$) such that

$$\left[ \sum_{k=1}^{n_\kappa} c_{\kappa_1}^k \mathbf{M}_{\kappa_1}^k \right] \boldsymbol{f}_{\kappa_1} = \boldsymbol{r}_{\kappa_1} \tag{16b}$$

where the constant $c_{\kappa_1}^k$ and the vector $\boldsymbol{r}_{\kappa_1}$ are defined as

$$c_{\kappa_1}^k := (\boldsymbol{f}_x^\mathsf{T} \mathbf{K}_x^k \boldsymbol{f}_x)(\boldsymbol{f}_{\kappa_2}^\mathsf{T} \mathbf{M}_{\kappa_2}^k \boldsymbol{f}_{\kappa_2})$$

$$\boldsymbol{r}_{\kappa_1} := \sum_{l=1}^{n_s} (\boldsymbol{f}_x^\mathsf{T} \boldsymbol{q}_x^l)(\boldsymbol{f}_{\kappa_2}^\mathsf{T} \boldsymbol{q}_{\kappa_2}^l) \boldsymbol{q}_{\kappa_1}^l - \sum_{m=1}^{n-1} \sum_{k=1}^{n_\kappa} c_{\kappa_1}^{k,m} \mathbf{M}_{\kappa_1}^k \boldsymbol{f}_{\kappa_1}^m, \text{ and}$$

$$c_{\kappa_1}^{k,m} := (\boldsymbol{f}_x^\mathsf{T} \mathbf{K}_x^k \boldsymbol{f}_x^m)(\boldsymbol{f}_{\kappa_2}^\mathsf{T} \mathbf{M}_{\kappa_2}^k \boldsymbol{f}_{\kappa_2}^m).$$

And, finally, problem (12c) becomes: given $\boldsymbol{f}_x^m$, $\boldsymbol{f}_{\kappa_1}^m$ and $\boldsymbol{f}_{\kappa_2}^m$ for $m = 1, \ldots, n-1$ (characterizing $u_{\text{PGD}}^{n-1}$), $\boldsymbol{f}_x$ (representing $F_x$) and $\boldsymbol{f}_{\kappa_1}$ (representing $F_{\kappa_1}$), update $\boldsymbol{f}_{\kappa_2}$ (representing $F_{\kappa_2}$) such that

$$\left[ \sum_{k=1}^{n_\kappa} c_{\kappa_2}^k \mathbf{M}_{\kappa_2}^k \right] \boldsymbol{f}_{\kappa_2} = \boldsymbol{r}_{\kappa_2} \tag{16c}$$

where the constant $c_{\kappa_2}^k$ and the vector $\boldsymbol{r}_{\kappa_2}$ are defined as

$$c_{\kappa_2}^k := (\boldsymbol{f}_x^\mathsf{T} \mathbf{K}_x^k \boldsymbol{f}_x)(\boldsymbol{f}_{\kappa_1}^\mathsf{T} \mathbf{M}_{\kappa_1}^k \boldsymbol{f}_{\kappa_1})$$

$$\boldsymbol{r}_{\kappa_2} := \sum_{l=1}^{n_s} (\boldsymbol{f}_x^\mathsf{T} \boldsymbol{q}_x^l)(\boldsymbol{f}_{\kappa_1}^\mathsf{T} \boldsymbol{q}_{\kappa_1}^l) \boldsymbol{q}_{\kappa_2}^l - \sum_{m=1}^{n-1} \sum_{k=1}^{n_\kappa} c_{\kappa_2}^{k,m} \mathbf{M}_{\kappa_2}^k \boldsymbol{f}_{\kappa_2}^m, \text{ and}$$

$$c_{\kappa_2}^{k,m} := (\boldsymbol{f}_x^\mathsf{T} \mathbf{K}_x^k \boldsymbol{f}_x^m)(\boldsymbol{f}_{\kappa_1}^\mathsf{T} \mathbf{M}_{\kappa_1}^k \boldsymbol{f}_{\kappa_1}^m).$$

Equations (16) are linear systems of equations of the size of the number of degrees of freedom each variable dimension, that is $d_x$, $d_{\kappa_1}$ and $d_{\kappa_2}$ respectively.

## 3. PGD FOR THE GEOMETRY PARAMETRIZATION

The parametrization of the geometry is essential in CAD and also in exploring the design space for shape optimization problems. A similar problem is associated with the parametric characterization of the location of internal boundaries or interfaces, which is very important for inverse problems (identification of geometrical patterns through indirect measurements, for instance).
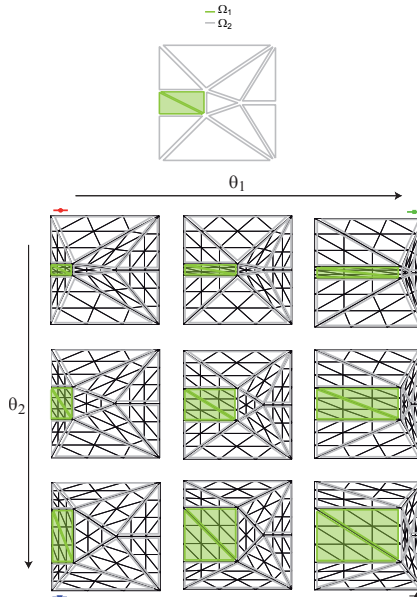
Figure 1. Example 3: Triangular macro-element mesh (left, $n_T = 11$). The two subdomains $\Omega_1$ and $\Omega_2$ are represented in two different colors. The geometry of the macro-elements varies as a function of the parameters $\boldsymbol{\theta}$ (right). Here $\theta_1$ is the horizontal size of $\Omega_1$ and $\theta_2$ its vertical size. The geometry of the (finer) finite element mesh is also mapped parametrically from a sub-mesh of the reference element $\Omega_R$.

Thus, the extension of PGD to deal also with the geometrical parametrization is of outmost practical interest because it reduces drastically the cost of sampling the design space. This idea was recently proposed in [12]. Here this approach is used and further improved to deal with a larger number of independent design parameters. Moreover, it is also extended to encompass the concept of geometrical parametrization of internal boundaries and material subdomains.

In this section, the same model problem presented in (1) is adopted. Now, the parametrization is not concerned with diffusivity $\kappa$ but rather with the geometrical definition of $\Omega$ (its boundary) and the different subdomains in which $\Omega$ is partitioned. Thus, it is assumed that there are two parameters, say $\theta_1$ and $\theta_2$, describing the geometry. Thus, now the computational domain $\Omega$ depends on $(\theta_1, \theta_2)$ and also the subdomains description depends on them, viz. $\Omega_1(\theta_1, \theta_2)$, $\Omega_2(\theta_1, \theta_2)$. The presentation is restricted to two parameters for the sake of an easer reading, there is no conceptual difficulty in increasing the number of parameters or subdomains. This is illustrated in Figure 1 for the configuration of example 3.

The statement of the problem is identical to the previous one, except for the fact that the new parameters are $\theta_1$ and $\theta_2$ instead of $\kappa_1$ and $\kappa_2$. This modifies (6) and (7), where $\kappa_1$ and $\kappa_2$ have to be directly replaced by $\theta_1$ and $\theta_2$. However, in the case of the geometrical parameters $\theta_1$ and $\theta_2$, there is no explicit parametric dependence in the equation as described in (8) for $\kappa_1$ and $\kappa_2$. This section is devoted to discuss the rationale of the techniques characterizing the effect of the geometrical parameters in the equation and, hence, in the solution.

### 3.1. Domain decomposition of the parametric geometry

In order to properly represent the problem dependence on the geometrical parameters, it is assumed that the domain $\Omega$ is divided in a set of simple (typically triangular) subdomains that are referred as
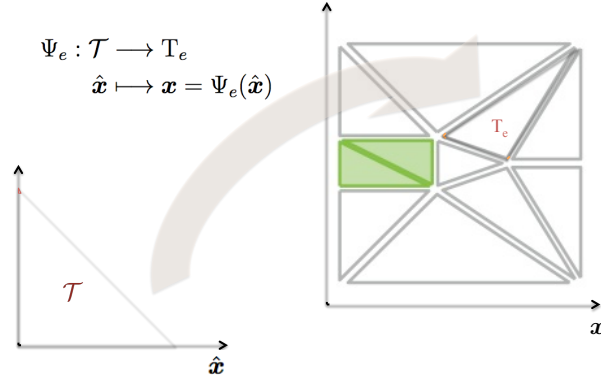
Figure 2. Each macro-element $T_e$ is the image of a reference element $\mathcal{T}$ with mapping $\Psi_e$.

macro-elements $T_1$, $T_2$ , ... , $T_{n_T}$. These macro-elements constitute a finite element mesh denoted as the geometrical mesh because it is used to characterize the geometry, both the boundary of $\Omega$ and the internal interfaces, not to introduce a functional approximation. Therefore, the macro-elements cover the complete domain, that is the closure of $\bigcup_{e=1}^{n_T} T_e$ coincides with the closure of $\Omega$.

When $\Omega$ is partitioned into subdomains, for instance $\Omega_1$ and $\Omega_2$ as shown in Figure 1, the macro-elements are nested into these subdomains. That is, the subdomains are divided in one or more macro-elements but one macro-element $T_e$ cannot intersect two or more subdomains. Thus, the parametric geometrical characterization of the macro-elements characterizes also the geometry of the subdomains.

Following the standard idea of the isogeometric mapping extensively used in FE formulations, it is assumed that each macro-element $T_e$, for $e = 1, 2, \ldots, n_T$, is the image of a reference element $\mathcal{T}$ described by the coordinates $\hat{\boldsymbol{x}}$, via a mapping denoted by $\Psi_e$, see Figure 2. That is,

$$\Psi_e : \mathcal{T} \longrightarrow T_e$$
$$\hat{\boldsymbol{x}} \longmapsto \boldsymbol{x} = \Psi_e(\hat{\boldsymbol{x}})$$

Each mapping depends on the geometry parameters; that is, $\Psi_e(\theta_1, \theta_2)$ or $\Psi_e(\boldsymbol{\theta})$ for $e = 1, 2, \ldots, n_T$. The Jacobian matrix associated with this mapping is denoted by $\mathbf{J}_e(\boldsymbol{\theta}) = \partial \Psi_e / \partial \hat{\boldsymbol{x}}$.

The functional approximation is performed in terms of these macro-elements mappings in the usual finite element fashion. A function $u(\boldsymbol{x})$ is therefore described in terms of its macro-element contributions parametrized with the coordinates $\hat{\boldsymbol{x}}$ denoted by $u_e(\hat{\boldsymbol{x}})$, $e = 1, \ldots, n_T$. This is formally written as

$$u = \sum_{e=1}^{n_T} u_e \circ \Psi_e^{-1}, \tag{17}$$

in the sense that every point $\boldsymbol{x}$ in $\Omega$ is mapped back to $\mathcal{T}$ according to the macro-element $T_e$ where it lies. In the following, any function $u$ taking values in $\Omega$ is identified with the set of local restrictions to the macro-elements, $u_1, \ldots, u_{n_T}$.

Thus, the expressions in (4), become

$$a(u, v) = \int_\Omega \nabla u \cdot (\kappa \nabla v) \, d\Omega = \sum_{e=1}^{n_T} \int_{\mathcal{T}} \nabla_{\hat{\boldsymbol{x}}} u_e \cdot (\underbrace{\kappa |\mathbf{J}_e| \mathbf{J}_e^{-\mathsf{T}} \mathbf{J}_e^{-1}}_{\mathbf{D}_e} \nabla_{\hat{\boldsymbol{x}}} v_e) \, d\hat{\boldsymbol{x}}$$

and

$$\ell(v) = \int_{\Omega} sv \, d\Omega = \sum_{e=1}^{n_T} \int_{\mathcal{T}} sv_e |\mathbf{J}_e| \, d\hat{\boldsymbol{x}}$$

where it is assumed that the gradient operator $\nabla$ is acting in the coordinates $\boldsymbol{x}$ and $\nabla_{\hat{\boldsymbol{x}}}$ is its counterpart in the reference element described by the coordinates $\hat{\boldsymbol{x}}$, $|\mathbf{J}_e|$ is the Jacobian determinant and $\mathbf{D}_e := \kappa |\mathbf{J}_e| \mathbf{J}_e^{-\mathsf{T}} \mathbf{J}_e^{-1}$ is the (fictitious) conductivity tensor associated with every macro-element $\mathrm{T}_e$.

As the integrals are resolved in the reference element $\mathcal{T}$ instead of $\Omega$, the volume element $d\hat{\boldsymbol{x}}$ replaces $d\Omega$. Both $\mathbf{J}_e$ and $|\mathbf{J}_e|$, and therefore also $\mathbf{D}_e$, depend on the geometry parameters $\boldsymbol{\theta}$, and the dependence is different for each macro-element $\mathrm{T}_e$, for $e = 1, 2, \ldots, n_T$.

When mappings $\Psi_e$ are linear (for triangular macro-elements), the Jacobian is constant inside each macro-element (it does not depend on $\hat{\boldsymbol{x}}$ but it is still different for each $e$) and therefore also $\mathbf{D}_e$ is independent of $\hat{\boldsymbol{x}}$.

The discretization space is also build on the basis of this domain decomposition. The functional approximation mesh (the standard one, associated with $V_x^h$) is constructed by assembling sub-meshes of the macro-elements of the geometry mesh, as illustrated in Figure 1. Accordingly, the space discretization functions $\boldsymbol{N}_x$ are built by assembling the local contributions from the macro-elements, that is

$$\boldsymbol{N}_x = \overset{n_T}{\underset{e=1}{\mathbf{A}}} \boldsymbol{N}_\xi \circ \Psi_e^{-1},$$

where it is assumed that the submeshes of all macro-elements are topologically identical and result of mapping a mesh of $\mathcal{T}$ associated with the vector of shape functions $\boldsymbol{N}_\xi$. The symbol $\mathbf{A}$ denotes the assembly operator, standard in finite elements, acting as a smart sum that relates local and global connectivities.

### 3.2. PGD formulation for the domain decomposition parametrization

As previously said, in the case of geometrical parametrization equation (6) is still valid replacing $\kappa_1$ and $\kappa_2$ by $\theta_1$ and $\theta_2$. Thus, the multidimensional operators read

$$A(u, v) = \int_{I_{\theta_1}} \int_{I_{\theta_2}} a(u, v) \, d\theta_2 \, d\theta_1 = \sum_{e=1}^{n_T} \int_{I_{\theta_1}} \int_{I_{\theta_2}} \int_{\mathcal{T}} \nabla_{\hat{\boldsymbol{x}}} u_e \cdot (\mathbf{D}_e \nabla_{\hat{\boldsymbol{x}}} v_e) \, d\hat{\boldsymbol{x}} \, d\theta_2 \, d\theta_1$$

and

$$L(v) = \int_{I_{\theta_1}} \int_{I_{\theta_2}} \ell(v) \, d\theta_2 \, d\theta_1 = \sum_{e=1}^{n_T} \int_{I_{\theta_1}} \int_{I_{\theta_2}} \int_{\mathcal{T}} sv_e |\mathbf{J}_e| \, d\hat{\boldsymbol{x}} \, d\theta_2 \, d\theta_1$$

Analogously to equation (11), the PGD solution $u_{\text{PGD}}^n = u_{\text{PGD}}^{n-1} + F_x F_{\theta_1} F_{\theta_2}$ is sought by using an iterative scheme that solves a nonlinear approximation problem. The problem reads: given $u_{\text{PGD}}^{n-1}$, find $F_x$, $F_{\theta_1}$ and $F_{\theta_2}$ such that

$$A(F_x F_{\theta_1} F_{\theta_2}, v_\star) = L(v_\star) - A(u_{\text{PGD}}^{n-1}, v_\star) \begin{cases} \text{for all } \delta F_\star \in V_\star^h, \\ \text{with } v_\star = \delta F_\star F_x F_{\theta_1} F_{\theta_2} / F_\star, \\ \text{for } \star = x, \theta_1 \text{ and } \theta_2. \end{cases} \quad (19)$$

The strategy is identical to the formulation for the classical (non geometrical) parametrization and the nonlinear problem (19) has to be split in three linear successive iterations analogous to equations (12).

When dealing with the geometry parametrization, the difference lies in the separability of the operators $A(\cdot, \cdot)$ and $L(\cdot)$ and the resulting matrix forms. This is mainly related with the separability of $|\mathbf{J}_e|$ and $\mathbf{D}_e$. As already mentioned, in the case of triangular macro-elements $\mathbf{J}_e$ does not depend on $\hat{\boldsymbol{x}}$. Thus, for each $e$, $|\mathbf{J}_e|$ and $\mathbf{D}_e$ depend only on $\theta_1$ and $\theta_2$. The following developments are made under this assumption.

The parametric dependence of $|\mathbf{J}_e|$ and $\mathbf{D}_e$ on $\boldsymbol{\theta}$ is readily derived from the analytical expressions giving the coordinates of the macro-elements vertices (the nodes of the geometrical mesh) as functions of $\boldsymbol{\theta}$.

Let $\boldsymbol{x}_1 = (x_1, y_1)$, $\boldsymbol{x}_2 = (x_2, y_2)$ and $\boldsymbol{x}_3 = (x_3, y_3)$ be the vertices of the macro-element $\mathrm{T}_e$. Note that $\boldsymbol{x}_1(\boldsymbol{\theta})$, $\boldsymbol{x}_2(\boldsymbol{\theta})$ and $\boldsymbol{x}_3(\boldsymbol{\theta})$ depend on $\boldsymbol{\theta}$ in a different way for each $e$, the dependence on $e$ is omitted in the notation. The dependence on $\boldsymbol{\theta}$ of the macro-element coordinates is assumed to be analytically known and given by a simple expression.

Taking as $\mathcal{T}$ the element of vertices $\hat{\boldsymbol{x}}_1 = (1, 0)$, $\hat{\boldsymbol{x}}_2 = (0, 1)$ and $\hat{\boldsymbol{x}}_3 = (0, 0)$ equipped with the standard linear shape functions,

$$N_1(\hat{x}_1, \hat{x}_2) = \hat{x}_1, \quad N_2(\hat{x}_1, \hat{x}_2) = \hat{x}_2 \text{ and } N_3(\hat{x}_1, \hat{x}_2) = 1 - \hat{x}_1 - \hat{x}_2,$$

the Jacobian reads

$$\mathbf{J}_e(\boldsymbol{\theta}) = \left[ \frac{\partial \boldsymbol{x}}{\partial \hat{\boldsymbol{x}}} \right] = \begin{pmatrix} x_1(\boldsymbol{\theta}) - x_3(\boldsymbol{\theta}) & x_2(\boldsymbol{\theta}) - x_3(\boldsymbol{\theta}) \\ y_1(\boldsymbol{\theta}) - y_3(\boldsymbol{\theta}) & y_2(\boldsymbol{\theta}) - y_3(\boldsymbol{\theta}) \end{pmatrix} = \begin{pmatrix} \Delta x_{13} & \Delta x_{23} \\ \Delta y_{13} & \Delta y_{23} \end{pmatrix},$$

where a notation for the difference of the nodal coordinates has been introduced: $\Delta x_{ij} := x_i(\boldsymbol{\theta}) - x_j(\boldsymbol{\theta})$ and $\Delta y_{ij} := y_i(\boldsymbol{\theta}) - y_j(\boldsymbol{\theta})$, for $i, j = 1, 2, 3$. Thus, the expression for the determinant is

$$|\mathbf{J}_e| = \Delta x_{13} \Delta y_{23} - \Delta x_{23} \Delta y_{13}, \tag{20}$$

the inverse of the Jacobian reads,

$$\mathbf{J}_e^{-1} = \frac{1}{|\mathbf{J}_e|} \begin{pmatrix} \Delta y_{23} & -\Delta x_{23} \\ -\Delta y_{13} & \Delta x_{13} \end{pmatrix},$$

and the expression for matrix $\mathbf{D}_e$ is

$$\mathbf{D}_e = \frac{1}{|\mathbf{J}_e|} \begin{pmatrix} (\Delta y_{23})^2 + (\Delta x_{23})^2 & -\Delta y_{23} \Delta y_{13} - \Delta x_{13} \Delta x_{23} \\ -\Delta y_{23} \Delta y_{13} - \Delta x_{13} \Delta x_{23} & (\Delta x_{13})^2 + (\Delta y_{13})^2 \end{pmatrix}. \tag{21}$$

Even if the parametric dependence of each nodal coordinate $\boldsymbol{x}_i$ in $\boldsymbol{\theta}$ is simple, it is clear from (20) and (21) that the functional dependence of $|\mathbf{J}_e|$ and $\mathbf{A}_e$ is much more involved. Recall that a separable form of $A(\cdot, \cdot)$ and $L(\cdot)$ as defined in (18) is required to carry out the PGD implementation. This requires a separable expression for $|\mathbf{J}_e|$ and $\mathbf{D}_e$ that, in most of the cases, cannot be obtained analytically from the expressions available for $\boldsymbol{x}_i$. Note that if the geometrical deformation that can

be described using very simple parametric expressions, it would be possible to obtain analytical versions of (20) and (21). The advantage of the procedure described next is its applicability to any regular deformation in 2 and 3 dimensions (the regularity assumption is preventing singular geometrical mappings).

### 3.3. Approximated separable expressions for $\mathbf{D}_e$ and $|\mathbf{J}_e|$

For a given macro-element $\mathrm{T}_e$, the dependence of $\boldsymbol{x}_i$ ($i = 1, 2, 3$) on $\boldsymbol{\theta}$ is known analytically (and presumably simple). As stated in sections 2.2 and 2.3, the parametric spaces for $\theta_1$ and $\theta_2$, $\mathcal{L}^2(I_{\theta_1})$ and $\mathcal{L}^2(I_{\theta_2})$, are readily discretized with a finite element spaces $V_{\theta_1}^h$ and $V_{\theta_2}^h$. The spaces $V_{\theta_1}^h$ and $V_{\theta_2}^h$ are associated with a grid of sampling points in $I_{\theta_1}$ and $I_{\theta_2}$ and, hence, in $I_{\theta_1} \times I_{\theta_2}$.

The expressions given in (20) and (21) are difficult to handle analytically but they are simple to evaluate in all the points of the grid in $I_{\theta_1} \times I_{\theta_2}$. Taking this set of sample points as snapshots, the POD technique provides a fair separable approximation of both $|\mathbf{J}_e|$ and $\mathbf{D}_e$. The POD approximation for $|\mathbf{J}_e|$ reads

$$|\mathbf{J}_e| \approx \sum_{l=1}^{n_J^e} T_{\theta_1}^{e,l} T_{\theta_2}^{e,l}, \tag{22}$$

where $n_J^e$ is the number of terms required to fairly represent $|\mathbf{J}_e|$ with the POD and $T_{\theta_1}^{e,l}$ and $T_{\theta_2}^{e,l}$ the corresponding modes (depending on $\theta_1$ and $\theta_2$ respectively).

Note that the symmetric $2 \times 2$ matrix $\mathbf{D}_e$ (in 2D problems) has three independent components, namely, $c_1^e = [\mathbf{D}_e]_{1,1}$, $c_2^e = [\mathbf{D}_e]_{1,2} = [\mathbf{D}_e]_{2,1}$ and $c_3^e = [\mathbf{D}_e]_{2,2}$. Thus, the fictitious conductivity matrix for each macro-element can be written as

$$\mathbf{D}_e = \sum_{a=1}^{3} c_a^e \mathbf{I}_a,$$

where

$$\mathbf{I}_1 := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{I}_2 := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ and } \mathbf{I}_3 := \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

The POD approximation of $\mathbf{D}_e$ is performed by analyzing the three components separately and reads

$$\mathbf{D}_e \approx \sum_{a=1}^{3} \sum_{m=1}^{n_D^{e,a}} G_{\theta_1}^{e,a,m} G_{\theta_2}^{e,a,m} \mathbf{I}_a, \tag{23}$$

where $n_D^{e,a}$ is the number of modes of the POD approximation of $c_a^e$, and $G_{\theta_1}^{e,a,m}$ and $G_{\theta_2}^{e,a,m}$ the corresponding functions (depending on $\theta_1$ and $\theta_2$ respectively).

According to the numerical experiments less than three terms ($n_D^{d,a} \leq 3$) are sufficient to obtain fairly accurate approximations (errors $< 1\%$). This is probably due to the smooth parametrical dependence of the nodal locations $\boldsymbol{x}(\boldsymbol{\theta})$.

If more than two geometrical dimensions are present, a High Order Singular Value Decomposition (or similar procedure) is required to obtain separable approximations of $c_a^e$ and $|\mathbf{J}_e|$. The number of terms involved in the separated expressions, in that case, could increase, depending on the dependence of the macro-elements on the parameters (not all macro-elements depend on all parameters). This raise in the number of terms will produce an increased computational effort,

nevertheless, it is usually not critical because it only affects the off-line phase of the scheme. Using the HO-SVD code provided in [15], the authors have successfully run examples involving up to 6 geometrical dimensions.

Moreover, the computation cost of evaluating the separated operators is partially mitigated by pre-computing the involved matrices. The evaluation, then, reduces to matrix times vector operations. In the case of geometrical parameters, as it is described in the following, the sum of matrices is replaced by an assembly operation with no significant increase of computational complexity.

The expression corresponding to (14) is found recalling the separability of $u$ and $v$, that is $u = F_x F_{\theta_1} F_{\theta_2}$ and $v = \tilde{F}_x \tilde{F}_{\theta_1} \tilde{F}_{\theta_2}$. Moreover, as expressed in (17), the geometrical domain decomposition via the macro-elements identifies each of the space modes $F_x(\boldsymbol{x})$ and $\tilde{F}_x(\boldsymbol{x})$ with the local contributions $F_{x,e}(\hat{\boldsymbol{x}})$ and $\tilde{F}_{x,e}(\hat{\boldsymbol{x}})$. Thus, the expression corresponding to (14) is now

$$
\begin{aligned}
A(u,v) &= \sum_{e=1}^{n_T} \sum_{a=1}^{3} \sum_{m=1}^{n_D^{e,a}} \int_{I_{\theta_1}} \int_{I_{\theta_2}} \int_{\mathcal{T}} \nabla_{\hat{\boldsymbol{x}}} F_{x,e} F_{\theta_1} F_{\theta_2} \cdot (G_{\theta_1}^{e,a,m} G_{\theta_2}^{e,a,m} \mathbf{I}_a \nabla_{\hat{\boldsymbol{x}}} \tilde{F}_{x,e} \tilde{F}_{\theta_1} \tilde{F}_{\theta_2}) \, d\hat{\boldsymbol{x}} \, d\theta_2 \, d\theta_1 \\
&= \sum_{e=1}^{n_T} \sum_{a=1}^{3} \sum_{m=1}^{n_D^{e,a}} \left[ \int_{\mathcal{T}} \nabla_{\hat{\boldsymbol{x}}} F_{x,e} \cdot (\mathbf{I}_a \nabla_{\hat{\boldsymbol{x}}} \tilde{F}_{x,e}) \, d\hat{\boldsymbol{x}} \right] \left[ \int_{I_{\theta_1}} G_{\theta_1}^{e,a,m} F_{\theta_1} \tilde{F}_{\theta_1} \, d\theta_1 \right] \left[ \int_{I_{\theta_2}} G_{\theta_2}^{e,a,m} F_{\theta_2} \tilde{F}_{\theta_2} \, d\theta_2 \right] \\
&= \sum_{e=1}^{n_T} \sum_{a=1}^{3} \sum_{m=1}^{n_D^{e,a}} a_{\mathcal{T}}^a(F_{x,e}, \tilde{F}_{x,e}) \; m_{\theta_1}^{e,a,m}(F_{\theta_1}, \tilde{F}_{\theta_1}) \; m_{\theta_2}^{e,a,m}(F_{\theta_2}, \tilde{F}_{\theta_2}),
\end{aligned}
$$

where the bilinear forms $a_{\mathcal{T}}^a(\cdot, \cdot)$, $m_{\theta_1}^{e,a,m}$ and $m_{\theta_2}^{e,a,m}$ are defined in Appendix B, as well as the matrices corresponding to their discretized forms, $\mathbf{K}_{\hat{x}}^a$, $\mathbf{M}_{\theta_1}^{e,a,m}$ and $\mathbf{M}_{\theta_2}^{e,a,m}$. Similarly,

$$
\begin{aligned}
L(v) &= \sum_{e=1}^{n_T} \int_{I_{\theta_1}} \int_{I_{\theta_2}} \int_{\mathcal{T}} s_e v_e |\mathbf{J}_e| \, d\hat{\boldsymbol{x}} \, d\theta_2 \, d\theta_1 \\
&= \sum_{e=1}^{n_T} \sum_{l=1}^{n_J^e} \int_{I_{\theta_1}} \int_{I_{\theta_2}} \int_{\mathcal{T}} s_e v_e T_{\theta_1}^{e,l} T_{\theta_2}^{e,l} \, d\hat{\boldsymbol{x}} \, d\theta_2 \, d\theta_1 \qquad (24) \\
&= \sum_{e=1}^{n_T} \sum_{l=1}^{n_J^e} \left[ \int_{\mathcal{T}} s_e \tilde{F}_{x,e} \, d\Omega \right] \left[ \int_{\theta_1} T_{\theta_1}^{e,l} \tilde{F}_{\theta_1} \, d\theta_1 \right] \left[ \int_{\theta_2} T_{\theta_2}^{e,l} \tilde{F}_{\theta_2} \, d\theta_2 \right],
\end{aligned}
$$

where it is assumed that the source term $s$ has its macro-element representations $s_e$, $e = 1, \ldots, n_T$. Moreover, here $s$ does not depend on the geometrical parameters $\theta_1$ and $\theta_2$ and therefore a separable approximation similar to (8b) is not required. The load vectors, corresponding to the different terms in the previous equation are denoted by $\boldsymbol{q}_{\hat{x}}^e$, $\boldsymbol{q}_{\theta_1}^{e,l}$ and $\boldsymbol{q}_{\theta_2}^{e,l}$, as introduced in Appendix B.

The next step requires splitting the problem into linear iterations. First, the spatial mode is iterated in the equation analogous to (12a). Here, an additional difficulty arises from the macro-element representation of the spatial modes. As mentioned above, in the discrete representations the sums over $e$ (ranging in the number of macro-elements $n_T$) are replaced in the matrix equations by the assembly operator $\mathbf{A}$.

When iterating to update the spatial mode $\boldsymbol{f}_x$, the resulting problem reads: given $\boldsymbol{f}_x^{\bar{m}}$, $\boldsymbol{f}_{\theta_1}^{\bar{m}}$ and $\boldsymbol{f}_{\theta_1}^{\bar{m}}$ for $\bar{m} = 1, \ldots, n-1$ (characterizing $u_{\text{PGD}}^{n-1}$), $\boldsymbol{f}_{\theta_1}$ (describing $F_{\theta_1}$) and $\boldsymbol{f}_{\theta_2}$ (describing $F_{\theta_2}$), update

$\boldsymbol{f}_x$ (describing $F_x$) such that

$$\mathop{\mathbf{A}}_{e=1}^{n_T} \left[ \sum_{a=1}^{3} c_x^{e,a} \, \mathbf{K}_{\hat{x}}^a \right] \boldsymbol{f}_x = \boldsymbol{r}_x \tag{25a}$$

where the constant $c_x^{e,a}$ and the vector $\boldsymbol{r}_x$ are defined as

$$c_x^{e,a} := \sum_{m=1}^{n_D^{e,a}} (\boldsymbol{f}_{\theta_1}^{\mathsf{T}} \mathbf{M}_{\theta_1}^{e,a,m} \boldsymbol{f}_{\theta_1})(\boldsymbol{f}_{\theta_2}^{\mathsf{T}} \mathbf{M}_{\theta_2}^{e,a,m} \boldsymbol{f}_{\theta_2})$$

$$\boldsymbol{r}_x := \mathop{\mathbf{A}}_{e=1}^{n_T} \left( \sum_{l=1}^{n_J^e} (\boldsymbol{f}_{\theta_1}^{\mathsf{T}} \boldsymbol{q}_{\theta_1}^{e,l})(\boldsymbol{f}_{\theta_2}^{\mathsf{T}} \boldsymbol{q}_{\theta_2}^{e,l}) \right) \boldsymbol{q}_{\hat{x}}^e - \sum_{\bar{m}=1}^{n-1} \mathop{\mathbf{A}}_{e=1}^{n_T} \left[ \sum_{a=1}^{3} c_x^{e,a,\bar{m}} \, \mathbf{K}_{\hat{x}}^a \right] \boldsymbol{f}_x^{\bar{m}}, \text{ and}$$

$$c_x^{e,a,\bar{m}} := \sum_{m=1}^{n_A^{e,a}} (\boldsymbol{f}_{\theta_1}^{\mathsf{T}} \mathbf{M}_{\theta_1}^{e,a,m} \boldsymbol{f}_{\theta_1}^{\bar{m}})(\boldsymbol{f}_{\theta_2}^{\mathsf{T}} \mathbf{M}_{\theta_2}^{e,a,m} \boldsymbol{f}_{\theta_2}^{\bar{m}})$$

As for the equation to iterate the mode corresponding to one of the geometrical parameters, say $\theta_1$, the resulting problems reads: given $\boldsymbol{f}_x^{\bar{m}}$, $\boldsymbol{f}_{\theta_1}^{\bar{m}}$ and $\boldsymbol{f}_{\theta_1}^{\bar{m}}$ for $\bar{m} = 1, \ldots, n-1$ (characterizing $u_{\mathrm{PGD}}^{n-1}$), $\boldsymbol{f}_x$ (describing $F_x$) and $\boldsymbol{f}_{\theta_2}$ (describing $F_{\theta_2}$), update $\boldsymbol{f}_{\theta_1}$ (describing $F_{\theta_1}$) such that

$$\left[ \sum_{e=1}^{n_T} \sum_{a=1}^{3} \sum_{m=1}^{n_A^{e,a}} c_{\theta_1}^{e,a,m} \, \mathbf{M}_{\theta_1}^{e,a,m} \right] \boldsymbol{f}_{\theta_1} = \boldsymbol{r}_{\theta_1} \tag{25b}$$

where $\boldsymbol{f}_{x,t}$ stands for the restriction of $\boldsymbol{f}_x$ to the nodes of the submesh discretizing macro-element $T_e$ and the constant $c_{\theta_1}^{e,a,m}$ and vector $\boldsymbol{r}_{\theta_1}$ are defined as

$$c_{\theta_1}^{e,a,m} := (\boldsymbol{f}_{x,e}^{\mathsf{T}} \mathbf{K}_{\hat{x}}^a \boldsymbol{f}_{x,e})(\boldsymbol{f}_{\theta_2}^{\mathsf{T}} \mathbf{M}_{\theta_2}^{e,a,m} \boldsymbol{f}_{\theta_2})$$

$$\boldsymbol{r}_{\theta_1} := \sum_{e=1}^{n_T} \sum_{l=1}^{n_J^e} (\boldsymbol{f}_{x,e}^{\mathsf{T}} \boldsymbol{q}_{\hat{x}}^e)(\boldsymbol{f}_{\theta_2}^{\mathsf{T}} \boldsymbol{q}_{\theta_2}^{e,l}) \boldsymbol{q}_{\theta_1}^{e,l} - \sum_{\bar{m}=1}^{n-1} \sum_{e=1}^{n_T} \sum_{a=1}^{3} \sum_{m=1}^{n_A^{e,a}} c_{\theta_1}^{e,a,m,\bar{m}} \mathbf{M}_{\theta_1}^{e,a,m} \boldsymbol{f}_{\theta_1}^{\bar{m}}, \text{ and}$$

$$c_{\theta_1}^{e,a,m,\bar{m}} := (\boldsymbol{f}_{x,e}^{\mathsf{T}} \mathbf{K}_{\hat{x}}^a \boldsymbol{f}_{x,e}^{\bar{m}})(\boldsymbol{f}_{\theta_2}^{\mathsf{T}} \mathbf{M}_{\theta_2}^{e,a,m} \boldsymbol{f}_{\theta_2}^{\bar{m}})$$

The problem corresponding to the other geometrical parameter $\theta_2$ is very similar to the one described in (25b) and therefore it is not worthy writing it here.

## 4.  GENERAL FORM FOR THE MIXED GEOMETRIC AND MATERIAL PARAMETRIZATION

Assume now that the problem is parametrized by $n_p$ material parameters $\kappa_1, \kappa_2, \ldots, \kappa_{n_p}$ and $n_g$ geometrical parameters $\theta_1, \theta_2, \ldots, \theta_{n_g}$. This section is devoted to present the PGD formulation that allows handling this complex parametrization by merging the concepts introduced in Sections 2 and 3. This is not introducing a further conceptual difficulty but the expressions become more involved.

The counterpart of (7) is now

$$u(\boldsymbol{x}, \boldsymbol{\kappa}, \boldsymbol{\theta}) \approx u_{\mathrm{PGD}}(\boldsymbol{x}, \boldsymbol{\kappa}, \boldsymbol{\theta}) = \sum_{m=1}^{n} F_x^m \prod_{p=1}^{n_p} F_{\kappa_p}^m \prod_{g=1}^{n_g} F_{\theta_g}^m \tag{26}$$

The separable approximation of the input data analogous to (8), (22) and (23) are now expressed as follows

$$\kappa \approx \sum_{k=1}^{n_\kappa} B_x^k \prod_{p=1}^{n_p} B_{\kappa_p}^k, \qquad s \approx \sum_{l=1}^{n_s} S_x^l \prod_{p=1}^{n_p} S_{\kappa_p}^l,$$

$$|\mathbf{J}_e| \approx \sum_{l=1}^{n_J^e} \prod_{g=1}^{n_g} T_{\theta_g}^{e,l}, \qquad \mathbf{D}_e \approx \sum_{a=1}^{3} \sum_{m=1}^{n_A^{e,a}} \prod_{g=1}^{n_g} G_{\theta_g}^{t,a,m} \mathbf{I}_a, \tag{27}$$

where the two last expressions have to be set for each macro-element $\mathrm{T}_e$, $e = 1, \ldots, n_T$. When needed, the spatial modes $B_x^k$ and $S_x^l$ are duly represented in terms of their descriptions in the macro-elements, that is $B_{x,e}^k$ and $S_{x,e}^l$.

Then, the iterative process to determine the spatial mode reads: given for $\bar{m} = 1, \ldots, n-1$ the vectors $\boldsymbol{f}_x^{\bar{m}}$, $\boldsymbol{f}_{\kappa_p}^{\bar{m}}$ (for $p = 1, \ldots, n_p$), and $\boldsymbol{f}_{\theta_g}^{\bar{m}}$ (for $g = 1, \ldots, n_g$), which characterize $u_{\mathrm{PGD}}^{n-1}$, as well as $\boldsymbol{f}_{\kappa_p}$ and $\boldsymbol{f}_{\theta_g}$ (for all values of $p$ and $g$), update $\boldsymbol{f}_x$ such that

$$\overset{n_T}{\underset{e=1}{\mathsf{A}}} \left[ \sum_{a=1}^{3} c_x^{e,a} \, \mathbf{K}_{\hat{x}}^a \right] \boldsymbol{f}_x = \boldsymbol{r}_x$$

where the scalar $c_x^{e,a}$ and the vector $\boldsymbol{r}_x$ are defined as

$$c_x^{e,a} := \sum_{m=1}^{n_A^{e,a}} \sum_{k=1}^{n_\kappa} \prod_{p=1}^{n_p} (\boldsymbol{f}_{\kappa_p}^\mathsf{T} \mathbf{M}_{\kappa_g}^k \boldsymbol{f}_{\kappa_p}) \prod_{g=1}^{n_g} (\boldsymbol{f}_{\theta_g}^\mathsf{T} \mathbf{M}_{\theta_g}^{e,a,m} \boldsymbol{f}_{\theta_g})$$

$$\boldsymbol{r}_x := \sum_{l=1}^{n_s} \prod_{p=1}^{n_p} (\boldsymbol{f}_{\kappa_p}^\mathsf{T} \boldsymbol{q}_{\kappa_p}^e) \boldsymbol{q}_x^l + \overset{n_T}{\underset{e=1}{\mathsf{A}}} \sum_{l=1}^{n_J^e} \prod_{g=1}^{n_g} (\boldsymbol{f}_{\theta_g}^\mathsf{T} \boldsymbol{q}_{\theta_g}^{e,l}) \boldsymbol{q}_{\hat{x}}^e - \sum_{\bar{m}=1}^{n-1} \overset{n_T}{\underset{e=1}{\mathsf{A}}} \left[ \sum_{a=1}^{3} c_x^{e,a,\bar{m}} \, \mathbf{K}_{\hat{x}}^a \right] \boldsymbol{f}_x^{\bar{m}}$$

$$c_x^{e,a,\bar{m}} := \sum_{m=1}^{n_A^{e,a}} \sum_{k=1}^{n_\kappa} \prod_{p=1}^{n_p} (\boldsymbol{f}_{\kappa_p}^\mathsf{T} \mathbf{M}_{\kappa_g}^k \boldsymbol{f}_{\kappa_p}^{\bar{m}}) \prod_{g=1}^{n_g} (\boldsymbol{f}_{\theta_g}^\mathsf{T} \mathbf{M}_{\theta_g}^{e,a,m} \boldsymbol{f}_{\theta_g}^{\bar{m}})$$

All the linear systems to iterate for one of the parametric modes have similar structures. For instance, taking parameter $\kappa_{\tilde{p}}$, the problem reads: given for $\bar{m} = 1, \ldots, n-1$ the vectors $\boldsymbol{f}_x^{\bar{m}}$, $\boldsymbol{f}_{\kappa_p}^{\bar{m}}$ (for $p = 1, \ldots, n_p$), and $\boldsymbol{f}_{\theta_g}^{\bar{m}}$ (for $g = 1, \ldots, n_g$), which characterize $u_{\mathrm{PGD}}^{n-1}$, as well as $\boldsymbol{f}_x$, and $\boldsymbol{f}_{\kappa_p}$ (for $p = 1, \ldots, \tilde{p}-1, \tilde{p}+1, \ldots, n_p$), and $\boldsymbol{f}_{\theta_g}$ (for all $g$), update $\boldsymbol{f}_{\kappa_{\tilde{p}}}$ such that

$$\sum_{e=1}^{n_T} \sum_{k=1}^{n_\kappa} c_{\kappa_{\tilde{p}}}^{e,k} \, \mathbf{M}_{\kappa_{\tilde{p}}}^k \boldsymbol{f}_{\kappa_{\tilde{p}}} = \boldsymbol{r}_{\kappa_{\tilde{p}}}$$

where the scalar $c_{\kappa_{\tilde{p}}}^{e,k}$ and the vector $\boldsymbol{r}_{\kappa_{\tilde{p}}}$ are defined as

$$c_{\kappa_{\tilde{p}}}^{e,k} := \sum_{a=1}^{3} \sum_{m=1}^{n_A^{e,a}} (\boldsymbol{f}_{x,e}^{\mathsf{T}} \mathbf{K}_{\hat{x}}^{a} \boldsymbol{f}_{x,e}) \prod_{\substack{p=1 \\ p\neq\tilde{p}}}^{n_p} (\boldsymbol{f}_{\kappa_p}^{\mathsf{T}} \mathbf{M}_{\kappa_p}^{k} \boldsymbol{f}_{\kappa_p}) \prod_{g=1}^{n_g} (\boldsymbol{f}_{\theta_g}^{\mathsf{T}} \mathbf{M}_{\theta_g}^{e,a,m} \boldsymbol{f}_{\theta_g})$$

$$\boldsymbol{r}_{\kappa_{\tilde{p}}} := \sum_{l=1}^{n_s} (\boldsymbol{f}_{x}^{\mathsf{T}} \boldsymbol{q}_{x}^{l})(\prod_{\substack{p=1 \\ p\neq\tilde{p}}}^{\tilde{n}_p} \boldsymbol{f}_{\kappa_p}^{\mathsf{T}} \boldsymbol{q}_{\kappa_p}^{l}) \boldsymbol{q}_{\kappa_{\tilde{p}}}^{l} - \sum_{\bar{m}=1}^{n-1} \sum_{e=1}^{n_T} \sum_{k=1}^{n_\kappa} c_{\kappa_{\tilde{p}}}^{e,k,\bar{m}} \, \mathbf{M}_{\kappa_{\tilde{p}}}^{k} \boldsymbol{f}_{\kappa_{\tilde{p}}}^{\bar{m}}, \text{ and}$$

$$c_{\kappa_{\tilde{p}}}^{e,k,\bar{m}} := \sum_{a=1}^{3} \sum_{m=1}^{n_A^{e,a}} (\boldsymbol{f}_{x,e}^{\mathsf{T}} \mathbf{K}_{\hat{x}}^{a} \boldsymbol{f}_{x,e}^{\bar{m}}) \prod_{\substack{p=1 \\ p\neq\tilde{p}}}^{n_p} (\boldsymbol{f}_{\kappa_p}^{\mathsf{T}} \mathbf{M}_{\kappa_p}^{k} \boldsymbol{f}_{\kappa_p}^{\bar{m}}) \prod_{g=1}^{n_g} (\boldsymbol{f}_{\theta_g}^{\mathsf{T}} \mathbf{M}_{\theta_g}^{e,a,m} \boldsymbol{f}_{\theta_g}^{\bar{m}})$$

Note that despite the complexity of the previous expressions, which involve many terms, the resulting systems are of the size expected and both matrices and force vectors are computed as linear combinations of a reduced number of different matrices that can be easily stored, as can be noted in Appendix B. The equations corresponding to any other parameter are similar to the previous one. The implementation of this methodology is much more efficient if the basic matrices and the corresponding modal vectors are properly stored. Consequently, book keeping strategies are particularly important.

## 5. NUMERICAL EXAMPLES

### 5.1. Two examples of parametric geometry

Two examples are presented next to validate the methodology proposed in Section 3 (geometry-parametrized Poisson problem) and to show its accuracy. Their simplicity allow comparing $u_{\text{PGD}}$ with other numerical or analytical solutions. The governing problem is presented in (1), having $\Omega$ divided into two subdomains, $\Omega_1(\boldsymbol{\theta})$ and $\Omega_2(\boldsymbol{\theta})$, that depend on two parameters $\theta_1$ and $\theta_2$ (diffusivities are kept constant). Boundary conditions are of Dirichlet type in the top and bottom of the model, with values of one and zero respectively, and Neumann homogeneous at the side walls.

In example 3, the shape of the subdomains is controlled by two geometrical parameters that determine the size of a rectangular interior domain as shown in Figure 1. Four sets of parameters, corresponding to the extreme values of the geometrical parameters were chosen to compare the PGD solution against the corresponding FE solution. That is, the maximum and minimum values of $\theta_1$ and $\theta_2$ describing the four corner panels in Figure 1. The convergence curves are plotted in Figure 3. The PGD solution with 40 terms has a relative error (in $L_2$ norm) of order $10^{-4}$ when compared with the four reference cases.

In example 4, two parameters $\boldsymbol{\theta} = (L, \theta)$, control the size and rotation of a small square inclusion. $L$ controls its size and $\theta$ its rotation between 0 and $\pi/4$ counterclockwise, see Figure 4.

A simple test is done by setting diffusivities of both materials to one, in order to obtain an obvious linear solution independent of the parameter for the geometry. The expected linear solution is found for all tested cases, see Figure 4. Computing 30 terms, the maximum nodal error is of order $10^{-2}$ and
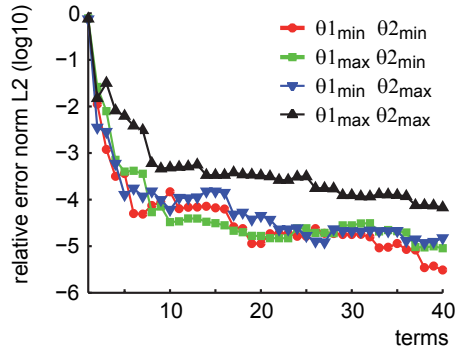
Figure 3. Example 3: Convergence of $u_{\text{PGD}}$ for the geometry-parametrized problem ($\kappa$ constant in each subdomain). Each curve corresponds to one value of the parameters that generates one of the geometries shown at the corner panels of Figure 1
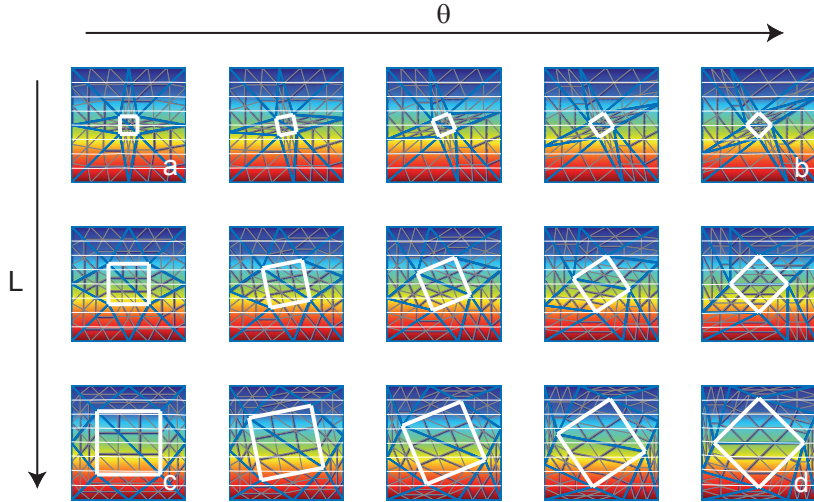


Figure 4. Example 4: Solution and meshes for different values of the geometry parameters. Colours represent the pgd solution for the case where $\kappa_1 = \kappa_2$ (linear solution expected) and thin white lines are the contour lines. Gray lines represent finite element mesh, blue lines the geometry mesh, and thick white lines indicate the internal boundary between $\Omega_1$ and $\Omega_2$.

the maximum error in $L_2$ norm is $10^{-4}$. It is important to note that similar errors are also obtained when the diffusivities of the two subdomains are different ($\kappa_1 = 1$ and $\kappa_2 = 0.01$).

The geometry parametrization as described in Section 3 is based on the parametric location of the nodes of the geometry mesh. Because the connectivity of the geometry mesh is kept constant, it is important to avoid the collapse or the flipping of the macro-elements.

### 5.2. Temperature field in a geological cross section

This last example studies the temperature distribution in the subsurface of the Earth crust; it has both scientific and practical interests. The geothermal gradient, for example, is used to estimate the thickness of the crust and tectonic plates, see for instance [21]. This gradient is also used to infer the age of the oceanic plates [22]. Moreover, knowledge on the temperature distribution is vital in
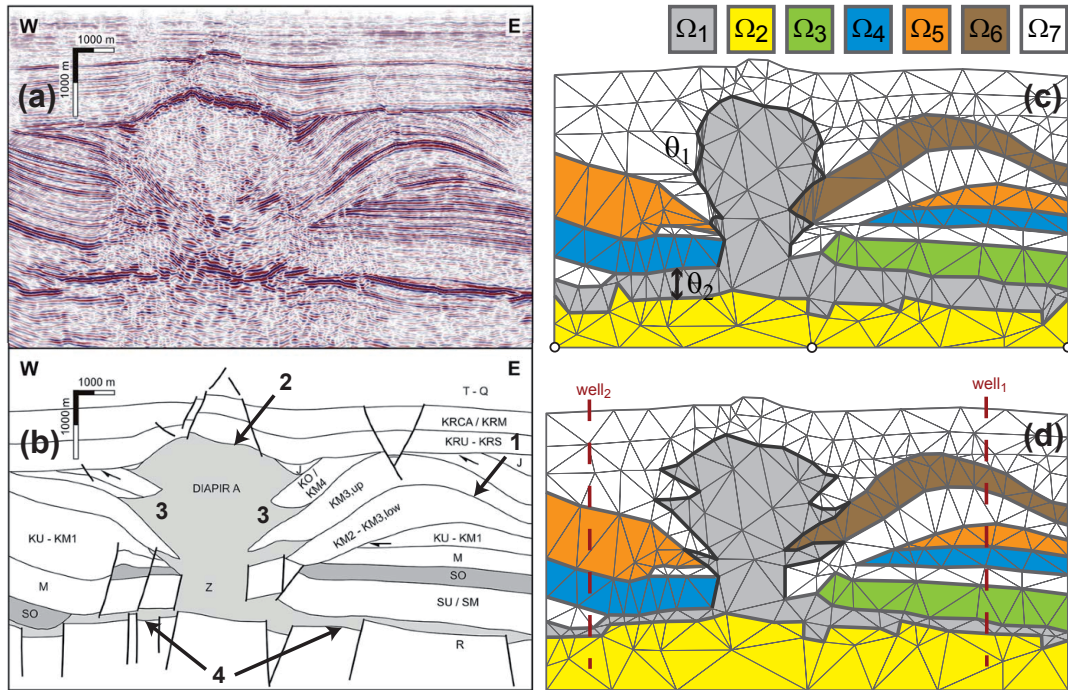
Figure 5. Example 5. Panels a and b from [24] show depth migrated seismic section and geological interpretation. Two rock contacts (interfaces) well resolved by the seismic line are indicated by labels 1 and 2 (compare the seismic data and the interpretation). Two poorly resolved features are the lateral contacts of the Diapir with the surrounding rocks (label 3) and the thickness of the Sant (gray) layer crossing the section. Panel c and d show the geometry mesh and the simplified structure used in the numerical model. They also show geometry variations produced by the geometrical parameters. Dashed lines on panel c show the approximate location of the wells 1 and 2.

regions where geothermal energy is exploited. And it is also important in the study of subsurface water flow, as convection patterns could be linked to the thermal state of the region [23].

The proposed example presents a model for the thermal state of the subsurface that depends on 8 parameters of two different kinds. First, different rock types have different physical properties such as density and thermal diffusivity. Some rocks, for example Halite (sedimentary rock composed by Salt) exhibit a wide range of thermal diffusivities depending on its composition. Therefore, the effective thermal diffusivity of each rock type in the model is controlled by its own parameter.

Second, the location of different rock bodies in the subsurface is in commonly determined using seismic studies. While many structures are very well resolved by the seismic data, others are not, leading to uncertainties in the location of the contacts (interfaces) between rock bodies. These unresolved, or noisy, patches are common in the presence of Salt, because of the very high contrast between rock properties. Therefore, the location of some contacts in the model is controlled by geometrical parameters.

This model is based in a cross section from the Central European Basin in North-West Germany where multiphase salt tectonics are present. Seismic data and its interpretation is taken from [24] and simplified to be included in this model. The transect exhibits a Salt Diapir, i.e. the central structure in Figure 5, cutting some sedimentary layers. It is important to stress that in this work the objective
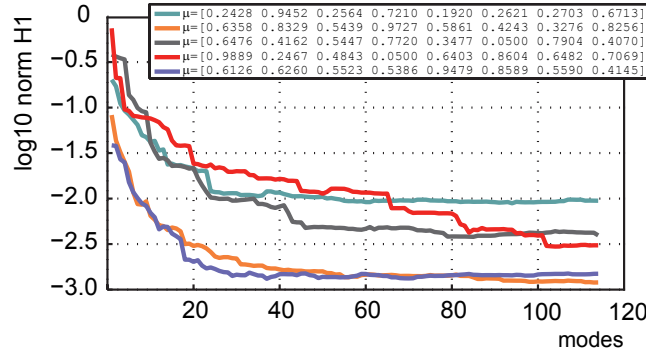
Figure 6. Example 5. Convergence in modes of $u_{\mathrm{PGD}}$ with respect to the corresponding FE solution for 5 random sets of parameters.

is not to gain geological insight on the region; the simplified section is used to show the capabilities of the numerical techniques presented so far.

Panels a and b of Figure 5 show seismic data and the geological interpretation provided by [24] (modified from their Figure 6). Examples of well resolved contacts (interfaces that can be univocally located based on the seismic data) and poorly resolved features were added to their Figure. Label 3 and 4 show two poorly resolved features: the smoothness of the lateral contact of the Salt body with surrounding rocks, and the thickness of the Salt layer that crosses the section. Two geometry parameters were chosen to investigate how these two features affect the thermal state of the section. The first parameter, $\theta_1$, determines the smoothness of the contact between the walls of the Diapir and surrounding rocks, and the second parameter, $\theta_2$, controls the thickness of the Salt layer crossing the domain.

In addition, six parameters control the thermal diffusivity of domains $\Omega_1$ to $\Omega_6$ shown in Figure 5. which correspond to different rock types. The last subdomain, $\Omega_7$, has a known constant diffusivity.

Boundary conditions are the same as used in previous examples (Dirichlet at top and bottom, homogeneous Neumann at the side walls).

Thus, the solution of the parametrized equation belongs to a 10 dimensional space (8 parameters plus space), and it has the form $u_{\mathrm{PGD}}(\boldsymbol{x}, \boldsymbol{\mu})$, where $\boldsymbol{\mu} = (\theta_1, \theta_2, \kappa_1, \ldots, \kappa_6)$. The geometrical parameters are $\theta_i \in (0, 1)$ for $i = 1, 2$ and the diffusivity parameter are $\boldsymbol{\kappa}_j \in (0.05, 1)$ for $j = 1$ to 6.

The PGD solution is compared with the corresponding FE solution for some random generated set of parameters. Convergence of $u_{\mathrm{PGD}}$ as the number of modes increases is depicted in Figure 6. Note that with 100 terms an error of order $10^{-2}$ is obtained for all tested cases.

The PGD solution is used next to solve several synthetic inverse problems. These problems simulate the practical situation where only a few observed quantities are available, for example temperature or heat flux measurements, and the parameters of the model are unknown. The goal is to recover the optimal set of parameters, using the available incomplete information.

In the synthetic problems the optimal set of parameters, $\boldsymbol{\mu}^{\mathrm{opt}}$, is chosen a priori. The complete temperature field is computed as $T^{\mathrm{opt}}(\boldsymbol{x}) = u_{\mathrm{PGD}}(\boldsymbol{x}, \boldsymbol{\mu}^{\mathrm{opt}})$. Next, the "observed" data is extracted from $T^{\mathrm{opt}}$ and, finally, using only the observed data the objective is to recover $\boldsymbol{\mu}^{\mathrm{opt}}$. This last step
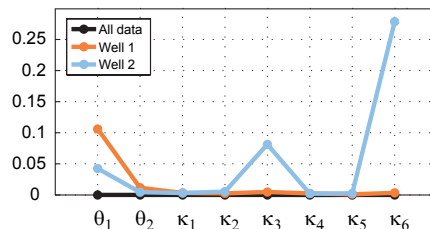
Figure 7. Example 5. Relative error of the recovered parameters, $\boldsymbol{\mu}^{\text{rec}}$, for three different data sets.
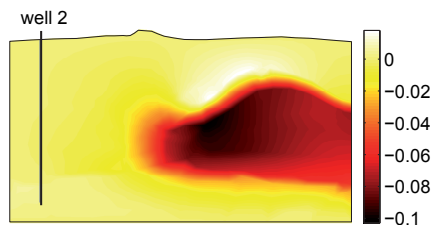


Figure 8. Example 5. The sensitivity of $\kappa_6$ in space is shown as the partial derivative of $u_{\text{PGD}}$ with respect to $\kappa_6$. Sampling values on the left part of the domain is not useful to determine the value of $\kappa_6$.

involves the solution of a constrained nonlinear optimization problem, that is implemented through the *fmincon* function of Matlab, using the Sequential Quadratic Programming algorithm.

Temperatures along two wells is used as "observed" data. The results of the inversions is summarized in Figure 7. This figure presents the relative error between recovered parameters, $\boldsymbol{\mu}^{\text{rec}}$, and the ones use to generate the synthetic example, $\boldsymbol{\mu}^{\text{opt}}$. The black curve corresponds to an inversion problem using the complete temperature field as the observed value. In that case the recovery of the parameters is almost exact, with differences of order $10^{-7}$ for all the parameters. The first geometrical parameter controlling the smoothness of the diapir, does not produce large temperature variations and therefore is is difficult to recover with large accuracy. Data from well 1 provides a very good approximation to the original parameters. In the case of well 2, most parameters are correctly recovered with the exception of $\kappa_6$; this is reasonable because the distance between the well 2 and $\Omega_6$ is large.

A major benefit of the PGD solution is that it provides the complete space of solutions and also its derivatives. Therefore, it is simple to estimate the sensitivity of $\kappa_6$ as $(\partial u_{\text{PGD}}/\partial \kappa_6)$ evaluated at $(\boldsymbol{x}, \boldsymbol{\mu}^{\text{opt}})$. Figure 8, which precisely plots this derivative, clearly indicates that temperature measurements in the left part of the domain do not provide useful information to recover parameter $\kappa_6$. Or in other words, variations of $\kappa_6$ do not produce any temperature change at well 2. This ability of easily evaluate sensitivities for any parameters can be extremely useful to decide the best location of a sampling well in order to maximize the usefulness of information.

Finally it is worth noting that in the solution of each one of the inversion problems, more that 1500 evaluations of the objective function were required. Each one of these evaluations is very efficient for $u_{\text{PGD}}$.

In this example the average number of iterations to compute each mode is three (a grand total of 152 iterations to compute 50 modes). It is difficult extracting conclusions comparing CPU time measurements, however, some figures are provided here for reference. The PGD offline phase took

$\approx$ 10 hours to complete in a standard computer. Then, once the PGD solution is available, the optimization phase involving approximately 1500 queries is solved in less than one minute.

On the other hand, a standard FEM code takes approximately 2 hours in performing these 1500 queries. Thus, for a single inverse problem and with the simple 2D FE model used here, the cost of pre-computing the PGD solution (ten hours for the PGD offline phase plus one minute online) is larger than a direct FE approach (which requires two hours online).

Consequently, in the case of solving a series of inverse problems corresponding to multiple data sets the PGD approach become competitive (in both approaches, PGD and direct FE, the online phase has to be performed many times but the offline PGD phase is performed just once). This is particularly interesting when the data sets evolve with time. Moreover, if the problem requires not only multiple queries but also fast queries (for example in control problems), the PGD solution is the only option (the online phase provides a fast response).

## 6. CONCLUSIONS

This work addresses a multiphase Poisson problem where diffusivities and interface locations are taken as parametric dimensions. Its multidimensional solution brings in the usual difficulties of high dimensional problems. The Proper Generalized Decomposition is used to circumvent these difficulties.

The parametric description of the geometry in the PGD context was first proposed by [12]. Here, this work is extended presenting a general procedure able to deal with more complex geometrical descriptions based on any number of geometrical parameters. An approximated separable expression for the inverse of the Jacobian of the mapping functions is devised to be applied in the PGD context.

The examples presented include geometrical parametrization of internal interfaces, together with parameters controlling material properties (viz. thermal diffusivity). An applied geophysical parameter-recovery problem, with its solution in a 10-dimensional space, is successfully solved using the proposed methodology.

In all the examples tested a relative small number of terms ($< 100$) and obtain a solution with an acceptable accuracy (relative errors $< 10^{-2}$). In addition, once computed, the PGD solution is evaluated extremely fast, making feasible to solve inverse problems in a short period of time with low computational resources.

### 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. *Journal of non-newtonian fluid mechanics* 2006; **139**:153–176.
2. 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 modeling of complex fluids. Part II: transient simulation using space-time separated representations. *Journal of non-newtonian fluid mechanics* 2007; **144**:98–121.

3. Chinesta F, Ammar A, Cueto E. Recent Advances and New Challenges in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models. *Archives of Computational methods in Engineering* 2010; **17**:327–350, doi:10.1007/s11831-010-9049-y.

4. Chinesta F, Leygue A, Bordeu F, Aguado JV, Cueto E, González D, Alfaro I, Ammar A, Huerta A. PGD-Based Computational Vademecum for efficient design, optimization and control. *Arch. Comput. Method Eng.* 2013; **20**:31–59.

5. Chun JS, Kim MK, Jung HK. Shape optimization of electromagnetic devices using immune algorithm. *IEEE Transactions On Magnetics* 1997; **33**(2).

6. Belytschko T, Xiao SP, Parimi C. Topology optimization with implicit functions and regularization. *International Journal For Numerical Methods In Engineering* 2003; **57**:1177–1196, doi:10.1002/nme.824.

7. Olesen LH, Okkels F, Bruus H. A high-level programming-language implementation of topology optimization applied to steady-state NavierStokes flow. *International Journal For Numerical Methods In Engineering* 2006; **65**:975–1001, doi:10.1002/nme.1468.

8. Rozza G, Veroy K. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer Methods in Applied Mechanics and . . .* 2007; .

9. Yoon GH, Sigmund O. A monolithic approach for topology optimization of electrostatically actuated devices. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**:4062–4075, doi:10.1016/j.cma.2008.04.004.

10. Nouy A, Chevreuil M, Safatly E. Fictitious domain method and separated representations for the solution of boundary value problems on uncertain parameterized domains. *Computer Methods in Applied Mechanics . . .* 2011; .

11. Iapichino L, Quarteroni A, Rozza G. A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks. *. . . methods in applied mechanics and . . .* 2012; .

12. Ammar A, Huerta A, Chinesta F, Cueto E, Leygue A. Parametric solutions involving geometry: A step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering* 2014; **268**:0178–0193, doi:10.1016/j.cma.2013.09.003.

13. González D, Ammar A, Chinesta F, Cueto E. Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering* 2010; **81**(5):637–659.

14. De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications* 2000; **21**(4):1253–1278.

15. Kolda T, Bader B. Tensor decompositions and applications. *SIAM Review* 2009; **51**(3):455–500.

16. Ammar A, Chinesta F, Díez P, Huerta A. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:1872–1880.

17. Falcó A, Nouy A. A proper generalized decomposition for the solution of elliptic problems in abstract form by using a functional eckart-young approach. *J. Math. Anal. Appl.* 2011; **376**:469–480.

18. Nouy A. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:4521–4537.

19. Chinesta F, Ammar A, Lemarchand F, Beauchene P, Boust F. Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization. *Computer Methods in Applied Mechanics and Engineering* 2008; **197**(5):400–413.

20. Niroomandi S, González D, Alfaro I, Bordeu F, Leygue A, Cueto E, Chinesta F. Real-time simulation of biological soft tissues: A PGD approach. *International Journal for Numerical Methods in Biomedical Engineering* 2013; **29**(5):586–600.

21. Zeyen H, Fernández M. Integrated lithospheric modeling combining thermal, gravity, and local isostasy analysis: Application to the NE Spanish Geotransect. *Journal of Geophysical Research: Solid Earth (1978–2012)* 1994; **99**(B9):18 089–18 102.

22. Afonso JC, Ranalli G, Fernández M. Density structure and buoyancy of the oceanic lithosphere revisited. *Geophysical Research Letters* 2007; **34**(10):10 302.

23. Fernández M, Torné M, Zeyen H. Modelling of thermal anomalies in the NW border of the Valencia Trough by groundwater convection. *Geophysical Research Letters* 1990; **17**(1):105–108.

24. Mohr M, Kukla PA, Urai JL, Bresser G. Multiphase salt tectonic evolution in NW Germany: seismic interpretation and retro-deformation. *International Journal of Earth Sciences* Nov 2005; **94**(5-6):917–940.

## A. OPERATORS AND MATRICES FOR THE DIFFUSION-PARAMETRIZED POISSON PROBLEM

The bilineal operators in Section 2.6 are defined as follows (for generic arguments $F_\star$ and $\tilde{F}_\star$)

$$a^k(F_x, \tilde{F}_x) := \int_\Omega \nabla \tilde{F}_x \cdot \left( B_x^k \nabla F_x \right) d\Omega, \qquad m_x(F_x, \tilde{F}_x) := \int_\Omega \tilde{F}_x F_x \, d\Omega,$$

$$m_{\kappa_1}^k(F_{\kappa_1}, \tilde{F}_{\kappa_1}) := \int_{I_{\kappa_1}} F_{\kappa_1} B_{\kappa_1}^k \tilde{F}_{\kappa_1} \, d\kappa_1, \qquad m_{\kappa_2}^k(F_{\kappa_2}, \tilde{F}_{\kappa_1}) := \int_{I_{\kappa_2}} F_{\kappa_2} B_{\kappa_2}^k \tilde{F}_{\kappa_2} \, d\kappa_2.$$

The matrices associated with the previous bilinear operators are the following (the vectors of shape functions $N_x$, $N_{\kappa_1}$ and $N_{\kappa_2}$ are column vectors)

$$\mathbf{K}_x^k = a^k(N_x, N_x^\top), \qquad\qquad \mathbf{M}_x = \int_\Omega N_x N_x^\top \, d\Omega,$$

$$\mathbf{M}_{\kappa_1}^k = m_{\kappa_1}^k(N_{\kappa_1}, N_{\kappa_1}^\top) = \int_{I_{\kappa_1}} N_{\kappa_1} B_{\kappa_1}^k N_{\kappa_1}^\top \, d\kappa_1, \quad \mathbf{M}_{\kappa_2}^k = m_{\kappa_2}^k(N_{\kappa_2}, N_{\kappa_2}^\top) = \int_{I_{\kappa_2}} N_{\kappa_2} B_{\kappa_2}^k N_{\kappa_2}^T \, d\kappa_2,$$

where index $k$ runs from 1 to the number of terms in the decomposition of $\kappa$.

The vectors associated with (15) read

$$\boldsymbol{q}_x^l = \int_\Omega S_x^l N_x \, d\Omega, \quad \boldsymbol{q}_{\kappa_1}^l = \int_{I_{\kappa_1}} S_{\kappa_1}^l N_{\kappa_1} \, d\kappa_1, \quad \boldsymbol{q}_{\kappa_2}^l = \int_{I_{\kappa_2}} S_{\kappa_2}^l N_{\kappa_2} \, d\kappa_2.$$

## B. OPERATORS AND MATRICES FOR THE GEOMETRICALLY PARAMETRIZED POISSON PROBLEM

The bilineal operators in Section 3.3 are defined as follows (for generic arguments $F_\star$ and $\tilde{F}_\star$)

$$a_{\mathcal{T}}^a(F_{x,e}, \tilde{F}_{x,e}) := \int_{\mathcal{T}} \nabla_{\hat{\boldsymbol{x}}} \tilde{F}_{x,e} \cdot (\mathbf{I}_a \nabla_{\hat{\boldsymbol{x}}} F_{x,e}) \, d\hat{\boldsymbol{x}},$$

$$m_{\theta_1}^{e,a,m}(F_{\theta_1}, \tilde{F}_{\theta_1}) := \int_{I_{\theta_1}} G_{\theta_1}^{e,a,m} F_{\theta_1} \tilde{F}_{\theta_1} \, d\theta_1,$$

and

$$m_{\theta_2}^{e,a,m}(F_{\theta_2}, \tilde{F}_{\theta_2}) := \int_{I_{\theta_2}} G_{\theta_2}^{e,a,m} F_{\theta_2} \tilde{F}_{\theta_2} \, d\theta_2.$$

They induce the following matrices

$$\mathbf{K}_{\hat{x}}^a = a_{\mathcal{T}}^a(N_{\hat{x}}, N_{\hat{x}}^\top), \quad \mathbf{M}_{\theta_1}^{e,a,m} = m_{\theta_1}^{e,a,m}(N_{\theta_1}, N_{\theta_1}^\top), \text{ and } \mathbf{M}_{\theta_2}^{e,a,m} = m_{\theta_2}^{e,a,m}(N_{\theta_2}, N_{\theta_2}^\top),$$

where indices range as follows: $e = 1, \ldots, n_T$, $a = 1, 2, 3$, and $m = 1, \ldots, n_A^{e;a}$. Thus, the number of matrices to be stored are:

- 3 matrices $\mathbf{K}_{\hat{x}}^a$ of size the number of nodes of the macro-element sub mesh

- $\sum_{e,a} n_A^{e,a}$ matrices $\mathbf{M}_{\theta_1}^{e,a,m}$ and $\mathbf{M}_{\theta_1}^{e,a,m}$ of size the dimension of number of nodes discretizing $I_{\theta_1}$ and $I_{\theta_1}$. If all macro-elements and components of $\mathbf{D}_e$ have the same number of terms $n_A$ in POD approximation (23), the number of matrices is $3n_T n_A$. In practice, the number of matrices to be precomputed and stored is much lower because for different values of indices $e$ and $a$, the value of $n_A^{e,a}$ is small (often equal to 1) if the parameters have nearly no influence in the component $a$ of element $e$.

Finally, the vectors associated with (24) read

$$\boldsymbol{q}_{\hat{x}}^e = \int_{\Omega_R} s_e \boldsymbol{N}_{\hat{x}} \, d\hat{\boldsymbol{x}}, \quad \boldsymbol{q}_{\theta_1}^{e,l} = \int_{\theta_1} T_{\theta_1}^{e,l} \boldsymbol{N}_{\theta_1} \, d\theta_1, \;\; \text{and} \; \boldsymbol{q}_{\theta_2}^{e,l} = \int_{\theta_2} T_{\theta_2}^{e,l} \boldsymbol{N}_{\theta_2} \, d\theta_2.$$