Parametric solutions involving geometry: A step towards efficient shape optimization ☆

Amine Ammar [a], Antonio Huerta [b,c], Francisco Chinesta [d], Elías Cueto [e,*], Adrien Leygue [d]

[a] *Arts et Métiers ParisTech 2 Boulevard du Ronceray, BP 93525, F-49035 Angers cedex 01, France*
[b] *Laboratori de Calcul Numeric (LaCaN), Universitat Politecnica de Catalunya, BarcelonaTech, 08034 Barcelona, Spain*
[c] *Civil & Computational Engineering Centre, College of Engineering, Swansea University, Swansea SA2 8PP, UK*
[d] *GEM UMR CNRS – Ecole Centrale de Nantes, 1 rue de la Noë, BP 92101, F-44321 Nantes cedex 3, France*
[e] *Aragon Institute of Engineering Research (I3A), Universidad de Zaragoza, Maria de Luna 3, E-50018 Zaragoza, Spain*

A B S T R A C T

Optimization of manufacturing processes or structures involves the optimal choice of many parameters (process parameters, material parameters or geometrical parameters). Usual strategies proceed by defining a trial choice of those parameters and then solving the resulting model. Then, an appropriate cost function is evaluated and its optimality checked. While the optimum is not reached, the process parameters should be updated by using an appropriate optimization procedure, and then the model must be solved again for the updated process parameters. Thus, a direct numerical solution is needed for each choice of the process parameters, with the subsequent impact on the computing time. In this work we focus on shape optimization that involves the appropriate choice of some parameters defining the problem geometry. The main objective of this work is to describe an original approach for computing an off-line parametric solution. That is, a solution able to include information for different parameter values and also allowing to compute readily the sensitivities. The curse of dimensionality is circumvented by invoking the Proper Generalized Decomposition (PGD) introduced in former works, which is applied here to compute geometrically parametrized solutions.

1. Introduction

The main objective of this work is to propose an original methodological approach to perform efficient numerical modeling and optimization. More specifically, the aim is to obtain approximations of the solution for models with parametrized geometries. These parametric approximations could then be employed for performing efficient shape optimization for example.

Optimization problems rely usually on iterative approaches. Optimal parameters (for instance, geometrical parameters describing the family of possible shapes) are obtained as extrema of a *cost function*. In general, for computational mechanics problems and, more particularly, in shape optimization, the evaluation of the cost function implies the resolution of a boundary value problem (BVP). Therefore, at each iteration, given a set of trial parameters, the feasibility and the optimality (value

of the cost function) require the resolution of a non trivial problem. Thus, until convergence a large number of, possibly costly, problems must be solved.

There are also a large variety of techniques to determine the updates on the trial parameters. If available, the derivatives of the cost function with respect to the design parameters are an important asset to improve the performance of convergence. The evaluation of these derivatives is, in general, not an easy task. They require to evaluate sensitivities of the solution of the BVP with respect to the design parameters and, although many techniques are available, this can have a computational cost non-negligible. In some cases the cost is of the order of the resolution of the BVP or even higher when the derivatives are evaluated using finite difference techniques, for instance.

The focus here is not to discuss particular optimization strategies, but pointing out that standard optimization strategies need numerous direct solutions of the problem, at least, one solution for each tentative geometry. The solution of such models is a tricky task that demands important computational resources and usually implies extremely large computing times, although good examples of commercial software exist nowadays.

Other works in the field such as [1–7] present efficient developments for shape and/or topology optimization techniques. Particularly noteworthy is the work covered in [3,4] where in order to characterize the topology of the domain, the design variables define a field (a function) which describes the geometry.

In this paper a radically different approach is proposed and, to the authors' knowledge, never previously explored. The main contribution of the suggested approach is to determine the solution of the BVP for any value of the unknown geometrical parameters (i.e. any value of the design variables). To this end, the design variables are viewed as new coordinates of the model and a separated representation is used to describe this general solution. In fact, coordinates, or space dimensions, represent the (non-necessarily physical) locations at which the solution is to be represented. Thus, strictly speaking, one could compute the solution of the problem for any value of the unknown parameters (in a bounded interval). This transforms the parameters in new dimensions of the space in which the model is defined.

How to determine this general function for any value of the geometrical parameters is discussed later. Note however, that determining this general function is done only once. Then, given its analytical expression (as a separated representation) determining the solution of the BVP for any set of trial design parameters is a simple post-process (very fast computation). Moreover, determining the sensibility of the solution of the BVP to any design parameter (geometry parameter) is also easy and fast because of the separated representation. The shape optimization consists then into the evaluation of an objective function (and, probably, some restrictions). Note that such an evaluation is now also very fast and efficient since the solution of the BVP is known for any design parameter and thus does not require to solve the BVP. Moreover, the general solution of the BVP is independent of the particular objective function.

To illustrate these ideas, consider a simple model, viz. the steady heat equation defined in the domain sketched in Fig. 1, whose parametric space reduces to the horizontal and vertical displacement of the upper right corner, μ_1 and μ_2. The traditional optimization procedures based on the minimization of a cost function $C_{ost}(\mu_1, \mu_2)$ can be summarized as follows:

- *Until a minimum of $C_{ost}(\mu_1, \mu_2)$ is reached, proceed to:*
  1. Compute the unknown field related to the trial choice of the geometry, i.e. $u(\mathbf{x}; \mu_1, \mu_2)$.
  2. Compute the cost function $C_{ost}(\mu_1, \mu_2)$ from the just calculated thermal field.
  3. Check the optimality: while the optimum is not reached, update the geometry, i.e. modify μ_1 and μ_2, by using an appropriate strategy and go back to step 1 to solve again the model in the newly updated geometry.

The methodology proposed here is substantially different. It is as follows:

- Determine a general solution of the thermal field for any possible geometry (here the location of the upper-right corner plays the same role that the space coordinates), the problem becoming multidimensional. This general solution is written as a sum of separable functions.
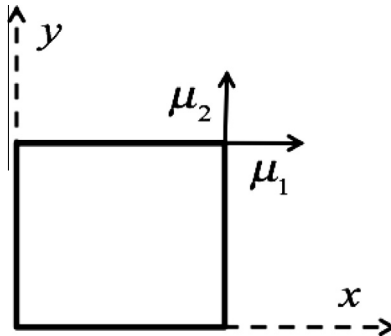

Fig. 1. Parametric domain.

- *Until a minimum of $C_{ost}(\mu_1, \mu_2)$ is reached, proceed to:*
  1. Particularize the general parametric solution at the considered values of the geometrical parameters (very fast).
  2. Compute the cost function $C_{ost}(\mu_1, \mu_2)$.
  3. Check the optimality: while the optimum is not reached, update the geometry by using an appropriate strategy (if sensitivities are needed use the known separated expression of the general solution to efficiently evaluate derivatives of the thermal field with respect to design parameters) and go back to step 1 to particularize again the parametric solution.

Thus, in the methodology proposed here, the model is solved only once and then it is particularized for any choice of the geometry. The price to pay is to solve a multidimensional thermal model that now has as coordinates the usual physical space x and all the geometrical parameters, in the present example the two extra-coordinates μ_1 and μ_2 defining the location of the upper-right corner.

Obviously, the solution of the resulting multidimensional model is a tricky task if one considers a standard mesh based discretization strategy because the number of degrees of freedom increases exponentially with the dimensionality of the model. Thus, for a hypercubic domain, the number of degrees of freedom scales with the number of nodes along each spatial direction to the power of the number of dimensions. For instance, in 2D if 100 nodes are used along each direction with a single degree of freedom per node, the resulting number of degrees of freedom is 100^2. In 3D, the number of degrees of freedom rises to 100^3 and so on. This exponential increase of the number of degrees of freedom can be literally out of reach for todays computers even if the number of dimensions increases only moderately. This phenomenon is known as *curse of dimensionality*. Although efficient techniques exist for moderate number of spatial dimensions, such as sparse grid methods, they fail when the dimensionality increases.

To circumvent this serious difficulty, the Proper Generalized Decompositions (PGD) is used. It considers a separated representation of the unknown field and was originally introduced in [8,9] for addressing multidimensional steady state and transient models, respectively. The interested reader can refer to [10,11] and the references therein for a complete review of PGD techniques. PGD techniques construct an approximation of the solution by means of a sequence of products of separable functions, circumventing the curse of dimensionality. These functions are determined "on the fly", as the method proceeds, with no initial assumption on their structure.

As can be readily noticed, the potential of the technique for inverse identification, optimization, among others, seems to be huge. This approach has been applied to optimization of structures and processes as well as for identification and simulation based control [12–16]. An alternative approach combining PGD and an efficient exploration of the parametric domain was considered in [17] within the non-linear-non-incremental LATIN framework [18–20]. Other related works in the field include [21,19]. On a related basis, [22] includes the management of high-dimensional experimental data in a PGD framework in order to construct surface response approaches to a given problem.

In Section 2 the basic ideas to construct a PGD separated representation, are revisited. In Section 3 the procedure and difficulties to include the geometrical parameters as extra-coordinates, are described for a model problem. This methodology is generalized and formalized in Section 4. Finally, Section 5 presents some numerical examples for illustrating the potentialities of the proposed approach.

2. Rationale of Proper Generalized Decomposition in a generic linear parametric model

Suppose the standard elliptic problem

$$
\begin{cases}
-\nabla \cdot \boldsymbol{K}\nabla u = f & \text{in } \Omega, \\
\boldsymbol{n} \cdot \boldsymbol{K}\nabla u = t & \text{on } \Gamma, \\
u = 0 & \text{on } \partial\Omega \setminus \Gamma,
\end{cases}
\tag{1}
$$

where the source term and the Neumann conditions are assumed constant to simplify the presentation, and the conductivity (diffusivity/permeability) matrix $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{\mu})$ has a spatial variation characterized by parameters $\boldsymbol{\mu} \in \mathbb{R}^m$. More precisely, $\boldsymbol{\mu} \in I_1 \times I_2 \times \cdots \times I_m$ where I_j is the range of variation of parameter μ_j.

A major contribution of the PGD approach is to view these parameters $\boldsymbol{\mu}$ characterizing conductivity $\boldsymbol{K}$ as new coordinates. Thus, instead of solving an excessively large number of thermal models for each different discrete value of these parameters, the objective is to solve at once a more general problem with $\boldsymbol{\mu}$ as extra coordinates. The price to pay being an increase of the problem dimensionality, since now $\boldsymbol{\mu}$ play the role of new coordinates in the model. However, as the complexity of the PGD scales only linearly (and not exponentially) with the space dimension, consideration of $\boldsymbol{\mu}$ as extra coordinates does not preclude to efficiently obtain an accurate solution.

The weak problem equivalent to (1) is obtained using a weighted residual argument, namely, find u for all δu in the selected appropriate functional space such that

$$
A(u, \delta u) = L(\delta u)
\tag{2a}
$$

with

$$A(u, \delta u) := \int_{I_1} \int_{I_2} \cdots \int_{I_m} (\boldsymbol{K}\nabla u, \nabla \delta u)_\Omega \, d\mu_1 d\mu_2 \ldots d\mu_m, \tag{2b}$$

$$L(\delta u) := \int_{I_1} \int_{I_2} \cdots \int_{I_m} [(f, \delta u)_\Omega + \langle t, \delta u \rangle_\Gamma] \, d\mu_1 d\mu_2 \ldots d\mu_m, \tag{2c}$$

where $(u, v)_\Omega$ and $\langle u, v \rangle_\Gamma$ denote, respectively, the $\mathcal{L}^2$ scalar product of functions u and v in Ω and its traces over Γ.

The PGD approach assumes, see [23], that the solution of (2) can be approximated by a rank-n separable approximation of the solution, $u(\boldsymbol{x}, \boldsymbol{\mu})$, namely,

$$u(\boldsymbol{x}, \boldsymbol{\mu}) \approx u^n(\boldsymbol{x}, \boldsymbol{\mu}) = \sum_{s=1}^n v^s(\boldsymbol{x}) \prod_{j=1}^m \omega_j^s(\mu_j),$$

$$= u^{n-1}(\boldsymbol{x}, \boldsymbol{\mu}) + v(\boldsymbol{x}) \prod_{j=1}^m \omega_j(\mu_j), \tag{3}$$

where v and $v^s \in \mathcal{H}^1_{\Gamma_D}$ while ω_j and $\omega_j^s \in \mathcal{L}^2(I_j)$ for $j = 1, \ldots, m$ and $s = 1, \ldots, n$, with

$$\mathcal{H}^1_{\Gamma_D} := \{v \in \mathcal{H}^1(\Omega) : v = 0 \text{ on } \Gamma_D\}. \tag{4}$$

Note that each ω_j^s can be normalized in the natural norm of its space.

Assume $u^{n-1}(\boldsymbol{x}, \boldsymbol{\mu})$ already known, then, the n-enrichment requires the evaluation of v and ω_j for $j = 1, \ldots, m$. Note that PGD is an a priori reduced order model where the separable functions are evaluated from the weak problem. The final number of terms n for convergence requires an error estimate, see, for instance, [24]. In practice, convergence is fast enough in elliptic problems to avoid the need for implementing an error estimator.

After substitution of (3) into (2), the weak problem becomes

$$A\left(v \prod_{j=1}^m \omega_j, \delta u\right) = L(\delta u) - A(u^{n-1}, \delta u), \tag{5a}$$

with the test functions are also separated as

$$\delta u = \delta v \prod_{j=1}^m \omega_j + \sum_{k=1}^m v \delta \omega_k \prod_{\substack{j=1 \\ j \neq k}}^m \omega_j. \tag{5b}$$

Note that (5) defines a non-linear problem that must be solved by means of a suitable iterative scheme. Newton's method is a straightforward alternative, see, for instance, [8,9]. However, simpler linearization strategies can also be applied, see [11]. The simplest one is an alternating direction, fixed-point algorithm, which was found remarkably robust in the present context. Each iteration consists of as many stages as separated functions, $m + 1$ in this case, that are repeated until convergence.

For simplicity in the exposition suppose $m = 2$. Thus, Eq. (5b) becomes

$$\delta u = \delta v \omega_1 \omega_2 + v \delta \omega_1 \omega_2 + v \omega_1 \delta \omega_2,$$

and, consequently, Eq. (5a) is transformed in the following three stages

1. Find $v \in \mathcal{H}^1 \Gamma_D$ for all $\delta v \in \mathcal{H}^1_{\Gamma_D}$ (ω_1 and ω_2 assumed known) such that $A(v \omega_1 \omega_2, \delta v \omega_1 \omega_2) = L(\delta v \omega_1 \omega_2) - A(u^{n-1}, \delta v \omega_1 \omega_2)$.
2. Find $\omega_1 \in \mathcal{L}^2(I_1)$ for all $\delta \omega_1 \in \mathcal{L}^2(I_1)$ (v and ω_2 assumed known) such that $A(v \omega_1 \omega_2, v \delta \omega_1 \omega_2) = L(v \delta \omega_1 \omega_2) - A(u^{n-1}, v \delta \omega_1 \omega_2)$.
3. Find $\omega_2 \in \mathcal{L}^2(I_2)$ for all $\delta \omega_2 \in \mathcal{L}^2(I_2)$ (v and ω_1 assumed known) such that $A(v \omega_1 \omega_2, v \omega_1 \delta \omega_2) = L(v \omega_1 \delta \omega_2) - A(u^{n-1}, v \omega_1 \delta \omega_2)$.

These three stages are iterated until convergence. The first stage is at most 3D (size of the spatial dimension), and all other stages, for each parameter, are 1D. Moreover, it is interesting to note, see Eq. (2b), that each equation for function ω_j associated to parameter μ_j is algebraic (there are no derivatives with respect to the parameters μ_j).

In summary, for each enrichment step of the separable approximation, see Eq. (3), this nonlinear three-stage procedure is required. In general, the number of iterations for each enrichment does not exceed ten. Although the exact n-value (rank of the approximation) needed to accurately approximate the solution depends on the solution separability and regularity, numerical evidence in elliptic problems reveals that n ranges between a few tens and a few hundreds. Thus, at most, the complexity of the PGD procedure is a few hundreds spatial problems (the cost of each 1D algebraic problem being negligible compared to the 3D one). This is, in general, orders of magnitude less expensive that the full dimensional problem $(3 + m)$ dimensions or solving each spatial (3D) problem for a given set of parameters. Note that in the previous example, with $m = 2$,

sampling for ten values of μ_1 and μ_2 is already equivalent to a hundred 3D problems. Clearly, the CPU time savings by applying the PGD can be of several orders of magnitude (see [9]).

Finally, it is important to note that if the structure of the operator $A(\cdot, \cdot)$ is separable, as it is the case many practical examples, see [11], each stage in of the nonlinear process can be drastically simplified.

3. Model problem introducing geometrical parameters as extra-coordinates

This section presents in a simple problem the inherent difficulties associated to the use parameters modifying the geometry. It also illustrates for this elementary problem depicted in Fig. 1 the proposed solution. The model problem described in (1) is further simplified with K as the identity matrix and $\Gamma = \emptyset$, namely

$$\begin{cases} -\nabla \cdot \nabla u = f & \text{in } \Omega(\boldsymbol{\mu}) \subset \mathbb{R}^2, \\ u = 0 & \text{on } \Gamma_D(\boldsymbol{\mu}) := \partial\Omega(\boldsymbol{\mu}), \end{cases} \tag{6}$$

where it is explicitly indicated the dependence of the computational domain and its boundary in the parameters $\boldsymbol{\mu} \in [-a, a] \times [-a, a] \subset \mathbb{R}^2$, with $a < 1$, in fact, in the example below $a = 0.7$. Thus, the weak problem is a particularization of (2), whose solution in the proper finite dimensional spaces gives an approximation of the general solution $u(\boldsymbol{x}, \boldsymbol{\mu})$ where $\boldsymbol{\mu}$ are treated as extra-coordinates.

In order to solve the problem using $\boldsymbol{\mu}$ as extra-coordinates a mapping relating the spatial domain $\Omega(\boldsymbol{\mu})$ to a reference one Ω_ξ is required. Note that while the mapping is obviously dependent on $\boldsymbol{\mu}$, Ω_ξ is independent on these parameters. In fact, $\Omega(\boldsymbol{\mu})$ is the image of Ω_ξ for a given mapping $\mathcal{M}_{\boldsymbol{\mu}}$, namely

$$\Omega(\boldsymbol{\mu}) := \mathcal{M}_{\boldsymbol{\mu}}(\Omega_\xi) = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} = \mathcal{M}_{\boldsymbol{\mu}}(\xi), \ \forall \xi \in \Omega_\xi\}.$$

In what follows a strategy to define a convenient mapping is proposed. In any case, this mapping allows writing the weak problem (2) over the reference domain with no dependence on $\boldsymbol{\mu}$, namely

$$A(u, \delta u) = L(\delta u) \tag{7a}$$

with

$$A(u, \delta u) := \int_{I_1} \int_{I_2} \cdots \int_{I_m} (\boldsymbol{J}^{-1}\nabla_\xi u, \boldsymbol{J}^{-1}\nabla_\xi \delta u \det \boldsymbol{J})_{\Omega_\xi} d\mu_1 d\mu_2 \ldots d\mu_m, \tag{7b}$$

$$L(\delta u) := \int_{I_1} \int_{I_2} \cdots \int_{I_m} (f, \delta u \det \boldsymbol{J})_{\Omega_\xi} d\mu_1 d\mu_2 \ldots d\mu_m, \tag{7c}$$

where $\boldsymbol{J} = [\partial \boldsymbol{x}/\partial \xi]$ is the Jacobian matrix of the mapping and $\det \boldsymbol{J}$ its determinant, white ∇_ξ is the gradient in reference coordinates.

Remark 1. Note the similarity between (7b) and (2b) with the following definition of $\boldsymbol{K} = \det \boldsymbol{J} [\boldsymbol{J}^{-T}\boldsymbol{J}^{-1}]$.

3.1. A first tentative geometrical transformation

In order to explicit the dependence of the model on both geometrical parameters, consider a first tentative geometrical transformation from the parameter space $\xi \in]0, 1[\times]0, 1[$ to the physical spatial domain $\boldsymbol{x} \in \Omega(\boldsymbol{\mu})$:

$$\boldsymbol{x} = \mathcal{M}_{\boldsymbol{\mu}}(\xi) := \xi + \frac{1}{2}\left(\xi^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \xi\right)\boldsymbol{\mu},$$

which entails a linear Jacobian matrix

$$\boldsymbol{J} = \left[\frac{\partial \boldsymbol{x}}{\partial \xi}\right] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \boldsymbol{\mu}^T \xi = \begin{pmatrix} 1 + \xi_2\mu_1 & \xi_2\mu_2 \\ \xi_1\mu_1 & 1 + \xi_1\mu_2 \end{pmatrix}.$$

The determinant of $\boldsymbol{J}$ and its inverse are respectively,

$$\det \boldsymbol{J} = 1 + \xi^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \boldsymbol{\mu},$$

$$\boldsymbol{J}^{-1} = \left[\frac{\partial \xi}{\partial \boldsymbol{x}}\right] = \frac{1}{\det \boldsymbol{J}} \begin{pmatrix} 1 + \xi_1\mu_2 & -\xi_2\mu_2 \\ -\xi_1\mu_1 & 1 + \xi_2\mu_1 \end{pmatrix}.$$

Replacing these definitions into (7) allows to apply the PGD methodology developed in Section 2. It is important to note that with this mapping most of the terms in (7) can be expressed in a separated form, implying a finite sum of products of func-

tions of ξ, functions of μ_1 and functions of μ_2. However, the inverse of the Jacobian, i.e. $\boldsymbol{J}^{-1}$ see (7b), introduces $\det \boldsymbol{J}$ in the denominator. This is a key issue because: (i) the expression of the inverse of a separated function in a separated form can involve many terms, and (ii) in order to build-up such separated representation

$$\frac{1}{\det \boldsymbol{J}} \approx \sum_{i=1}^{N} J_i^{\xi}(\xi) J_i^{\mu_1}(\mu_1) J_i^{\mu_2}(\mu_2)$$

a high-order singular value decomposition (HOSVD) is required. In a multidimensional space involving many extra-coordinates, the implementation of HOSVD becomes delicate and in any case non optimal.

Remark 2 (*Small perturbations*). The particular case of small perturbations allows for a separated representation of the inverse of the Jacobian with a reduced number of terms. Only for the assumption of small perturbation a general mapping induces a weak problem, see (7), where a PGD methodology produces approximations with a reasonable number of terms, see [5]. In this work, however, general perturbations are considered and consequently more efficient transformations are proposed.

3.2. Looking for simpler transformations

To facilitate a separate representation of the inverse of $\det \boldsymbol{J}$ even in the presence of large perturbations, an alternative mapping is proposed. As in the previous section, the same quadrilateral domain depicted in Fig. 1 is studied. And, again, the unit square is distorted perturbing the position of the upper left corner with parameters $\boldsymbol{\mu} = (\mu_1, \mu_2)^T$. That is, the coordinates of the four vertices P_i are $\boldsymbol{x}_i$, for $i = 1, \ldots, 4$ defined as

$$\begin{cases} \boldsymbol{x}_1 = (0,0)^T, \\ \boldsymbol{x}_2 = (1,0)^T, \\ \boldsymbol{x}_3 = (1 + \mu_1, 1 + \mu_2)^T, \\ \boldsymbol{x}_4 = (0,1)^T. \end{cases} \tag{8}$$

The new mapping requires a partition of the original geometry in non-overlapping triangles. In this case, two triangles Ω_1 and Ω_2 are defined; the first one is characterized by points (P_1, P_3, P_4) and the second one by (P_1, P_2, P_3).

The reference triangle $\mathcal{T}$ is defined in the reference domain of coordinates ξ and its vertices are $T_1 = (0,0)$, $T_2 = (1,0)$, and $T_3 = (0,1)$. The mapping of the parametric spatial domain $\Omega(\boldsymbol{\mu}) = \Omega_1(\boldsymbol{\mu}) \cup \Omega_2(\boldsymbol{\mu})$ into a reference configuration independent of $\boldsymbol{\mu}$ is described by two geometrical transformations $\mathcal{T} \rightarrow \Omega_1$ and $\mathcal{T} \rightarrow \Omega_2$. That is,

$$\Omega(\boldsymbol{\mu}) = \bigcup_{i=1}^{2} \Omega_i(\boldsymbol{\mu}) \quad \text{with } \Omega_i(\boldsymbol{\mu}) = \mathcal{M}_{\boldsymbol{\mu}}^i(\mathcal{T}) = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} = \mathcal{M}_{\boldsymbol{\mu}}^i(\xi), \forall \xi \in \mathcal{T}\} \quad \text{for } i = 1, 2.$$

In this case, the mapping for each element follows the well-known finite element strategy for a linear triangle. In particular, the interpolation of any function $u(\xi)$ defined in $\mathcal{T}$ is prescribed by

$$u(\xi) = u_1 N_1(\xi) + u_2 N_2(\xi) + u_3 N_3(\xi)$$

where u_i, for $i = 1, 2, 3$, denote the values of the field u at vertices T_1, T_2 and T_3, respectively, and where the well-known shape functions $N_i(\xi)$,

$$\begin{cases} N_1(\xi_1, \xi_2) = 1 - \xi_1 - \xi_2, \\ N_2(\xi_1, \xi_2) = \xi_1, \\ N_3(\xi_1, \xi_2) = \xi_2, \end{cases}$$

verify the Kroenecker's delta property.

Thus, the two mappings prescribing the domain transformation are

$$\begin{cases} \boldsymbol{x}(\xi, \boldsymbol{\mu}) = \mathcal{M}_{\boldsymbol{\mu}}^1(\xi) := \boldsymbol{x}_1(\boldsymbol{\mu}) N_1(\xi) + \boldsymbol{x}_3(\boldsymbol{\mu}) N_2(\xi) + \boldsymbol{x}_4(\boldsymbol{\mu}) N_3(\xi) & \text{for } \Omega_1, \\ \boldsymbol{x}(\xi, \boldsymbol{\mu}) = \mathcal{M}_{\boldsymbol{\mu}}^2(\xi) := \boldsymbol{x}_1(\boldsymbol{\mu}) N_1(\xi) + \boldsymbol{x}_2(\boldsymbol{\mu}) N_2(\xi) + \boldsymbol{x}_3(\boldsymbol{\mu}) N_3(\xi) & \text{for } \Omega_2, \end{cases}$$

where $\boldsymbol{x}_i$, coordinates of points P_i for $i = 1, \ldots, 4$, are defined in (8). Note, that, in this case, only $\boldsymbol{x}_3$ depends on the parameters $\boldsymbol{\mu}$.

Consequently, the Jacobian related to $\mathcal{M}_{\boldsymbol{\mu}}^1(\mathcal{T}) = \Omega_1(\boldsymbol{\mu})$ becomes:

$$\boldsymbol{J}_1 = \begin{pmatrix} x_3 - x_1 & y_3 - y_1 \\ x_4 - x_1 & y_4 - y_1 \end{pmatrix} = \begin{pmatrix} 1 + \mu_1 & 1 + \mu_2 \\ 0 & 1 \end{pmatrix},$$

and analogously for the transformation $\mathcal{M}_{\boldsymbol{\mu}}^2(\mathcal{T}) = \Omega_2(\boldsymbol{\mu})$

$$J_2 = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 + \mu_1 & 1 + \mu_2 \end{pmatrix}.$$

Finally, for this particular case, $\det J_1 = 1 + \mu_1$ and $\det J_2 = 1 + \mu_2$, whose inverse can be written from a single functional product.

These expressions for the Jacobian are substituted in (7), taking into account that the spatial forms are now integrated in each element, for instance

$$(J^{-1} \nabla_\xi u, J^{-1} \nabla_\xi \delta u \, \det J)_{\Omega_\xi} = \sum_{i=1}^{2} (J_i^{-1} \nabla_\xi u, J_i^{-1} \nabla_\xi \delta u \, \det J_i)_{\mathcal{T}}.$$

3.3. Separated form approximation

Following Section 2 and more particularly Eq. (3), for this example, the model solution is approximated in the separated form as

$$u(\boldsymbol{x}) \approx u^n(\boldsymbol{x}(\xi, \boldsymbol{\mu})) = \sum_{s=1}^{n} v^s(\xi) \, \omega_1^s(\mu_1) \, \omega_1^s(\mu_2) = u^{n-1}(\boldsymbol{x}(\xi, \boldsymbol{\mu})) + v(\xi) \, \omega_1(\mu_1) \, \omega_1(\mu_2),$$

where the approximation $u^n(\boldsymbol{x}(\xi, \boldsymbol{\mu}))$ is defined piecewise by its restriction to Ω_1 and Ω_2. As usual an incremental process is designed where u^{n-1} is assumed known while v, ω_1 and ω_2 are the unknown functions to be determined with the standard PGD approach. They are evaluated iterating with the three stages described in Section 2.

It is important to note that finite dimensional subspaces of $\mathcal{H}_{\Gamma_D}^1$, $\mathcal{L}^2(I_1)$ and $\mathcal{L}^2(I_2)$ must be chosen to solve numerically the problem. While any 1D discretization is possible for $\mathcal{L}^2(I_i)$, $i = 1, 2$, for the spatial discretization two alternatives are possible. The first option to determine function v is to create a nested mesh of finite elements inside the triangles Ω_1 and Ω_2 used to describe the geometry. That is the submesh for computations, which in this case is also composed by triangles, has no elements crossing the interface $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$. This is the best option because it induces optimal rates of convergence. Whereas suboptimal rates of convergence, see [25], are induced by the second alternative, which consists in generating a mesh independent of the macro triangles used to define the geometry and then assign to each Gauss point the corresponding Jacobian of the transformation.

Here $I_i := [-0.7, 0.7]$ for $i = 1, 2$ and, consequently, an approximation of the solution is obtained for any geometry perturbing the upper-right corner of the unit square in this range along the horizontal and vertical directions. Fig. 2 depicts the limit solutions obtained from $(\mu_1, \mu_2) = (-0.7, -0.7)$ and $(\mu_1, \mu_2) = (0.7, 0.7)$. Both solutions were compared with the ones computed by using finite elements in both geometries and both agree with two significant digits; the $\mathcal{L}^2$ error over the domain is $0.91 \cdot 10^{-3}$ and $0.29 \cdot 10^{-2}$.

4. General framework for domain dependence on PGD coordinates

To present the general approach for PGD problems where the geometry is described with parameters a simplified model problem is studied. Without loss of generality, the problem statement is restricted to 2D with homogenous Dirichlet boundary conditions, namely

$$\begin{cases} -\nabla \cdot \nabla u = f & \text{in } \Omega(\boldsymbol{\mu}) \subset \mathbb{R}^2, \\ \boldsymbol{n} \cdot \nabla u = t & \text{on } \Gamma(\boldsymbol{\mu}), \\ u = 0 & \text{on } \Gamma_D(\boldsymbol{\mu}) := \partial\Omega(\boldsymbol{\mu}) \setminus \Gamma(\boldsymbol{\mu}), \end{cases} \tag{9}$$

where it is explicitly indicated the dependence of the computational domain and its boundary in the parameters $\boldsymbol{\mu} \in \mathbb{R}^m$. The classical weak problem equivalent to (9) is: find $u \in \mathcal{H}_{\Gamma_D}^1$ such that

$$(\nabla u, \nabla \delta u)_\Omega = (f, \delta u)_\Omega + \langle t, \delta u \rangle_\Gamma \quad \forall \delta u \in \mathcal{H}_{\Gamma_D}^1. \tag{10}$$

Recall the definition of $\mathcal{H}_{\Gamma_D}^1$ in Eq. (4).

Assume for simplicity that $\partial\Omega(\boldsymbol{\mu})$ is defined piecewise linearly, that is, $\partial\Omega(\boldsymbol{\mu})$ is a polygon. Moreover, assume that parameters $\boldsymbol{\mu}$ correspond to the (perturbation of the) coordinates of a subset of the vertices of that polygon. Note that no restriction is imposed on this polygon (convexity, holes, etc.). Under these assumptions there exists a partition on n_{tg} non overlapping triangles covering the complete domain. Denote by Ω_e a general triangle, $e = 1, \ldots, n_{tg}$.

Remark 3. It is important to observe that perturbations in the position of one vertex only affect triangles connected to this vertex. This locality of the perturbation will facilitate the separability among parameters $\boldsymbol{\mu}$. It is obvious, that parameters associated to vertices belonging to different triangles are independent.
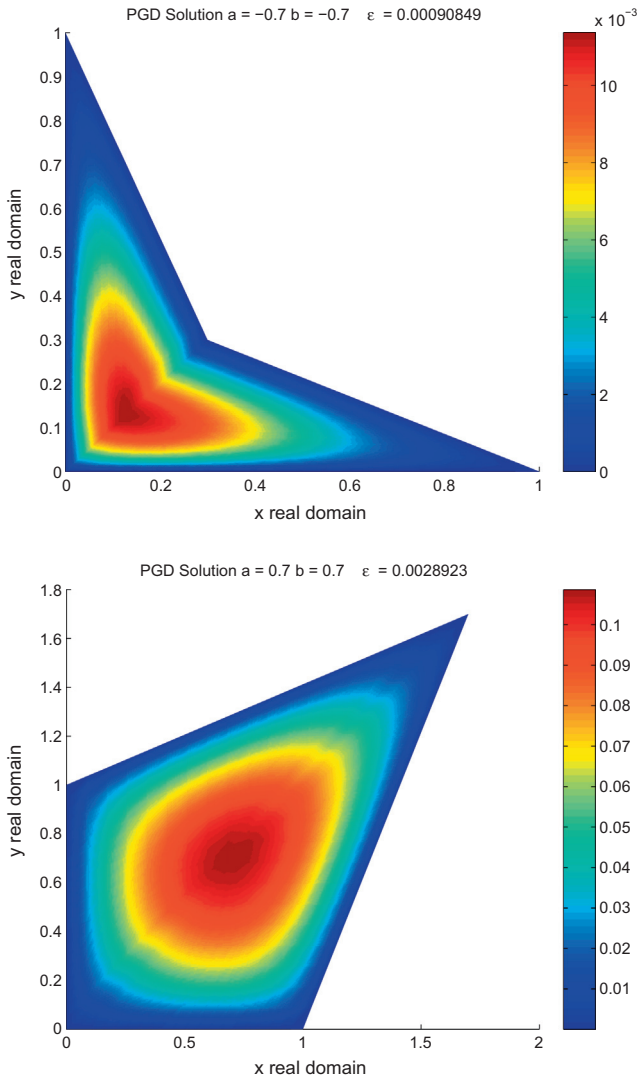
Fig. 2. Solution of the thermal model for: (top) $(\mu_1, \mu_2) = (-0.7, -0.7)$ and (bottom) $(\mu_1, \mu_2) = (0.7, 0.7)$.

Following the same rationale as in standard finite elements based on triangular discretization, for each triangle Ω_e, there is a linear mapping between the reference coordinates $\xi \in \mathcal{T}$ and the physical ones $\boldsymbol{x} \in \Omega_e$, i.e. $\boldsymbol{x} = \mathcal{M}_{\boldsymbol{\mu}}^e(\xi)$ for $e = 1, \ldots, \mathrm{n}_{\mathrm{tg}}$. It is crucial to note that, in each triangle, the Jacobian matrix, $\boldsymbol{J}_e := [\partial \boldsymbol{x}/\partial \xi]$ for $e = 1, \ldots, \mathrm{n}_{\mathrm{tg}}$, is "constant" (i.e. it does not depend on ξ) and it is only dependent on $\boldsymbol{\mu}$.

With these definitions, the weak problem (10), can be rewritten as

$$\sum_e^{\mathrm{n}_{\mathrm{tg}}} (\nabla u, \nabla \delta u)_{\Omega_e} = \sum_e^{\mathrm{n}_{\mathrm{tg}}} (f, \delta u)_{\Omega_e} + \sum_e^{\mathrm{n}_{\mathrm{tg}}} \langle t, \delta u \rangle_{\Gamma \cap \partial \Omega_e} \quad \forall \delta u \in \mathcal{H}_{\Gamma_D}^1,$$

or more explicitly

$$\sum_e^{\mathrm{n}_{\mathrm{tg}}} (\boldsymbol{K}_e \nabla_\xi u_e, \nabla_\xi \delta u_e)_{\mathcal{T}} \det(\boldsymbol{J}_e) = \sum_e^{\mathrm{n}_{\mathrm{tg}}} (f_e, \delta u_e)_{\mathcal{T}} \det(\boldsymbol{J}_e) + \sum_e^{\mathrm{n}_{\mathrm{tg}}} \langle t_e, \delta u_e \rangle_{\Gamma \cap \partial \mathcal{T}} \det(\boldsymbol{J}_e^\Gamma), \tag{11}$$

where $\boldsymbol{K}_e = \boldsymbol{J}_e^{-T} \boldsymbol{J}_e^{-1}$, $u_e = u|_{\Omega_e}$, ∇_ξ is the gradient with respect to the reference coordinates in the reference triangle $\mathcal{T}$, $\Gamma \cap \partial \mathcal{T}$ represent (symbolically) the edges of $\mathcal{T}$ where Neumann conditions are applied, and $\det(\boldsymbol{J}_e^\Gamma)$ the Jacobian along these edges. Note that u_e for $e = 1, \ldots, \mathrm{n}_{\mathrm{tg}}$ cannot be approximated independently for each triangle because continuity must be ensured.

This new expression of the weak problem, see (11), hints a separable approximation of the solutions. Because, for every $e = 1, \ldots, \mathrm{n}_{\mathrm{tg}}$, u_e, δu_e, f_e, and t_e only depend on ξ whereas $\boldsymbol{J}_e^{-1}$ (and consequently $\boldsymbol{K}_e$), $\det(\boldsymbol{J}_e)$ and $\det(\boldsymbol{J}_e^\Gamma)$ only depend on $\boldsymbol{\mu}$.

Under these circumstances, the major hypothesis of the PGD approach, which assumes that the solution of (10) can be approximated by a rank-n separable approximation, seems reasonable. This rank-n separable approximation is explicitly written as

$$
\begin{aligned}
u(\boldsymbol{x}) \approx u^n(\boldsymbol{x}(\xi, \boldsymbol{\mu})) &= \sum_{s=1}^{n} v^s(\xi) \prod_{j=1}^{m} \omega_j^s(\mu_j), \\
&= u^{n-1}(\boldsymbol{x}(\xi, \boldsymbol{\mu})) + v(\xi) \prod_{j=1}^{m} \omega_j(\mu_j),
\end{aligned}
\tag{12}
$$

where the approximation $u^n(\boldsymbol{x}(\xi, \boldsymbol{\mu})) \in \mathcal{H}_{\Gamma_D}^1$ is defined by each restriction to Ω_e for $e = 1, \ldots, n_{tg}$, and, in the following, u^{n-1} is assumed known while v and ω_j (with $j = 1, \ldots, m$) are the unknown functions.

Since the new variables $\boldsymbol{\mu}$ prescribing the geometry are now seen as extra coordinates, the weak problem, see (10) and (11), is now defined over a larger set of coordinates. In particular, (10) can be rewritten as

$$
\int_{I_1} \int_{I_2} \cdots \int_{I_m} (\nabla u, \nabla \delta u)_\Omega d\mu_1 d\mu_2 \ldots d\mu_m = \int_{I_1} \int_{I_2} \cdots \int_{I_m} (f, \delta u)_\Omega d\mu_1 d\mu_2 \ldots d\mu_m + \int_{I_1} \int_{I_2} \cdots \int_{I_m} \langle t, \delta u \rangle_\Gamma d\mu_1 d\mu_2 \ldots d\mu_m,
$$

where I_j is the range of parameter μ_j, for $j = 1, \ldots, m$. More precisely, replacing (12) in (11) gives

$$
\int_{I_1} \omega_1(\mu_1) \int_{I_2} \omega_2(\mu_2) \cdots \int_{I_m} \omega_m(\mu_m) \det(\boldsymbol{J}_e) \sum_e (\boldsymbol{K}_e \nabla_\xi v_e, \nabla_\xi \delta u_e)_T \, d\mu_1 d\mu_2 \ldots d\mu_m = r(\delta u),
\tag{13a}
$$

where $r(\delta u)$ is the residual evaluated for the $n - 1$ separation and defined as

$$
\begin{aligned}
r(\delta u) = &\int_{I_1} \int_{I_2} \cdots \int_{I_m} \det(\boldsymbol{J}_e) \sum_e (f_e, \delta u_e)_T \, d\mu_1 d\mu_2 \ldots d\mu_m + \int_{I_1} \int_{I_2} \cdots \int_{I_m} \det(\boldsymbol{J}_e^\Gamma) \sum_e \langle t_e, \delta u_e \rangle_{\Gamma \cap \partial T} \, d\mu_1 d\mu_2 \ldots d\mu_m \\
&- \int_{I_1} \int_{I_2} \cdots \int_{I_m} \det(\boldsymbol{J}_e) \sum_e (\boldsymbol{K}_e \nabla_\xi u^{n-1}, \nabla_\xi \delta u_e)_T \, d\mu_1 d\mu_2 \ldots d\mu_m.
\end{aligned}
\tag{13b}
$$

Note that once the test functions are also separated, recall (5b)

$$
\delta u = \delta v \prod_{j=1}^{m} \omega_j + \sum_{k=1}^{m} v \, \delta \omega_k \prod_{\substack{j=1 \\ j \neq k}}^{m} \omega_j,
$$

the rationale of PGD exposed in Section 2 can be readily applied and the iterative process to determine v and $\boldsymbol{\omega}$ is automatically determined.

5. Numerical results

In this section, the proposed approach is applied to more complex scenarios.

5.1. Quadrilateral with one parametrized edge

The first numerical experiment concerns problem (6) with a unitary source term on a domain $\Omega(\boldsymbol{\mu})$ obtained by perturbing the rectangular domain $\Omega(\mathbf{0}) =]0, 3[\times]0, 1[$. For this purpose, assume that $\partial\Omega(\boldsymbol{\mu})$ is described by the position of eight control points P_i, $i = 1, \ldots, 8$. Moreover, four of those control point are allowed to move vertically and the parameters describing this perturbation are $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4)^T$. In this example $\mu_i \in I_i := [-0.3, 0.3]$, for $i = 1, \ldots, 4$. See Table 1 for the coordinates of these points prescribing $\partial\Omega(\boldsymbol{\mu})$.

In order to apply the procedure previously described consider the six triangles: $\Omega_1 = (P_1, P_7, P_8)$, $\Omega_2 = (P_1, P_2, P_7)$, $\Omega_3 = (P_2, P_6, P_7)$, $\Omega_4 = (P_2, P_3, P_6)$, $\Omega_5 = (P_3, P_5, P_6)$ and $\Omega_6 = (P_3, P_4, P_5)$, as depicted in Fig. 3.

The resulting solution separated representation involves 40 terms

$$
u(\boldsymbol{x}(\xi, \boldsymbol{\mu})) \approx \sum_{s=1}^{40} v^s(\xi) \prod_{j=1}^{4} \omega_j^s(\mu_j).
\tag{14}
$$

Functions $\omega_j^s(\mu_j)$ were approximated using a 1D discretization consisting of 13 nodes uniformly distributed in the interval $[-0.3, 0.3]$. Functions $v^s(\xi)$ were approximated using a nested mesh of linear finite element depicted in Fig. 4.

Fig. 5 depicts the first three functions $v^s(\boldsymbol{x})$ (i.e. $s = 1, 2, 3$) and Fig. 6 also presents the first three functions $\omega_j^s(\mu_j)$, for $j = 1, \ldots, 4$.

Fig. 7 compares the finite element solution obtained directly over a domain perturbed by $\mu_1 = -0.15$, $\mu_2 = 0.3$, $\mu_3 = -0.3$, and $\mu_4 = 0.3$ and the results of a particularization of the PGD parametric solution for the same perturbation. The difference between both solution was, using a $\mathcal{L}^2$ norm, of around 10^{-2}. Moreover, Fig. 8 shows the evolution or the residual as a func-

Table 1
Coordinates of the points describing the boundary of the perturbed domain.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
x	0	1	2	3	3	2	1	0
y	0	0	0	0	$1 + \mu_1$	$1 + \mu_2$	$1 + \mu_3$	$1 + \mu_4$
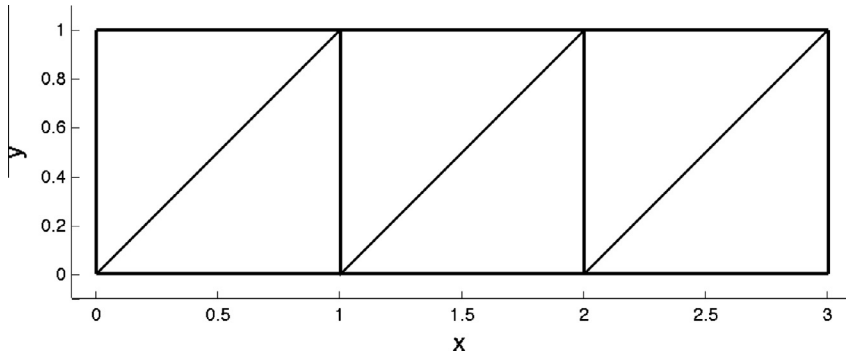
Fig. 3. Triangulation defining the geometrical transformation.
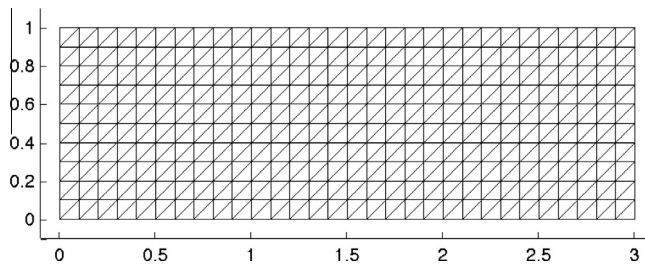
Fig. 4. Finite element nested mesh used to approximate the spatial functions $v^s(\xi)$.

tion of the number of terms involved in the separated representation. Recall the definition of the residual in (13b). Notice that Fig. 8 goes beyond the 40 terms used in all the comparisons presented here and also shows the residue up to 80. This is done to illustrate that as the number of terms increases the error decreases depending on the optimality of the method and the separability of the solution. Moreover, as also depicted in Fig. 8, the error between the PGD approximation and the direct finite element (FE) approximation decreases with the number of terms. Note that for $s = 40$ the difference between a post processed PGD approximation and a FE solution is already below 10^{-4}. Obviously, as the number of terms in the PGD approximation is increased the error can be reduced. Note that, this off-line solution is the general thermal solution for any point in space and any design parameter (geometric parameters). That is, a six-dimensional problem with 341 nodes in space and 13 nodes for each geometric parameter. Since four geometry parameters are used, only 45 s are needed in PGD to approximate the solution of $13^4 = 28\,561$ configurations that would imply (when proceeding with standard techniques) the solution of $28\,561$ two-dimensional spatial problems, each with 341 nodes, or, equivalently, a 6D problem with $9\,739\,301$ number of degrees of freedom, and the error compared to FE is below 10^{-4}.

Finally, Fig. 9 depicts the finite element solutions and the particularized parametric PGD solutions for two other different geometries. Again, both solutions agree to a great level of precision.

Obviously, adding more parameters as extra-coordinates is not a major issue, the strategy for building-up the separated representation proceeds in the same manner, but because the solution is now richer, the separated representation involves more terms. Assume now that $\Omega(\boldsymbol{\mu})$ is defined by 12 control points, and that, as previously, the position of the top ones, 6 in this case is perturbed. See Table 2 for the coordinates of these points prescribing $\partial\Omega(\boldsymbol{\mu})$ and note that $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6)^T$ is such that $\mu_i \in I_i := [-0.3, 0.3]$, for $i = 1, \ldots, 6$. Fig. 10 depicts the triangulation for $\Omega(\mathbf{0})$.

The resulting solution separated representation involves 70 terms

$$u(\boldsymbol{x}(\xi, \boldsymbol{\mu})) \approx \sum_{s=1}^{70} v^s(\xi) \prod_{j=1}^{6} \omega_j^s(\mu_j). \tag{15}$$

Fig. 11 compares the particularization of the general PGD approximation (15) when considering the geometry perturbed by $\mu_1 = -0.3$, $\mu_2 = 0.3$, $\mu_3 = 0.3$, $\mu_4 = -0.3$, $\mu_5 = 0.3$, and $\mu_6 = 0.3$ with the finite element solution computed directly on the
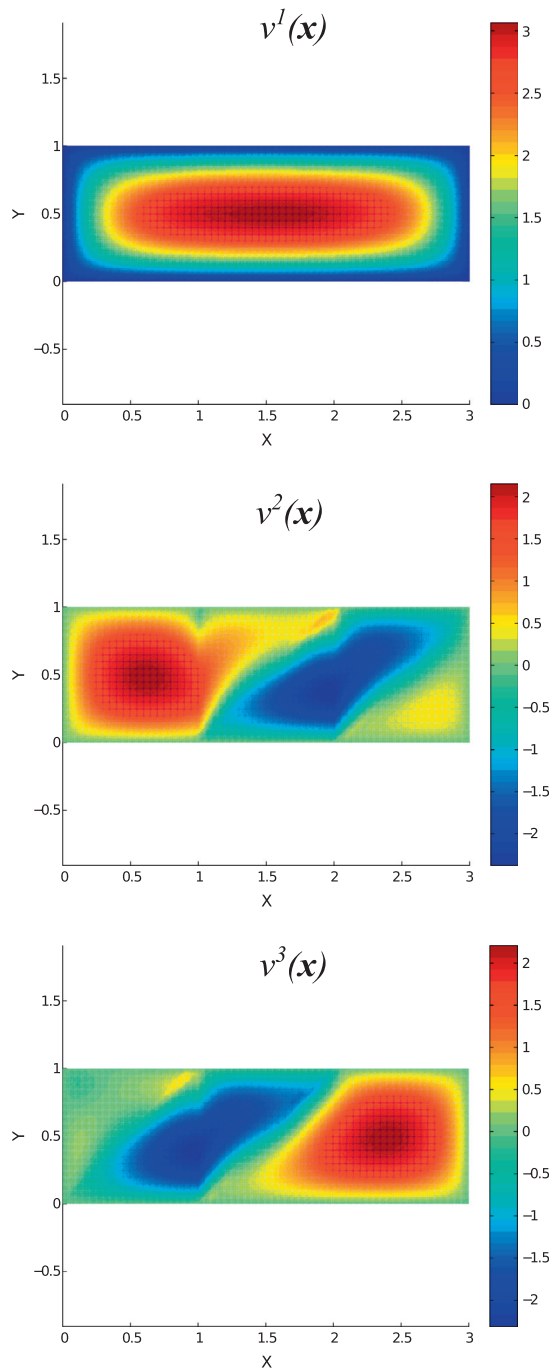
Fig. 5. Three most significant spatial modes.

perturbed domain. Again, both solutions are in perfect agreement. It is important to notice that also in this case all the intervals $I_i = [-0.3, 0.3]$ for $i = 1, \ldots, 6$ are discretized with 13 uniformly distributed nodes. Thus, separated representation in Eq. (15) represents the solution for 13^6 different geometries, that is, for 4 826 809 possible domain geometries.

5.2. First steps towards multi-criteria efficient shape optimization

This section is dedicated to show, in an academic problem, the advantages of a generalized (on-line) solution to address optimization problems. As noted in the introduction, optimization problems rely on iterative procedures that require the evaluation of an objective function (and/or multiple criteria). An efficient optimization scheme would reduce the number
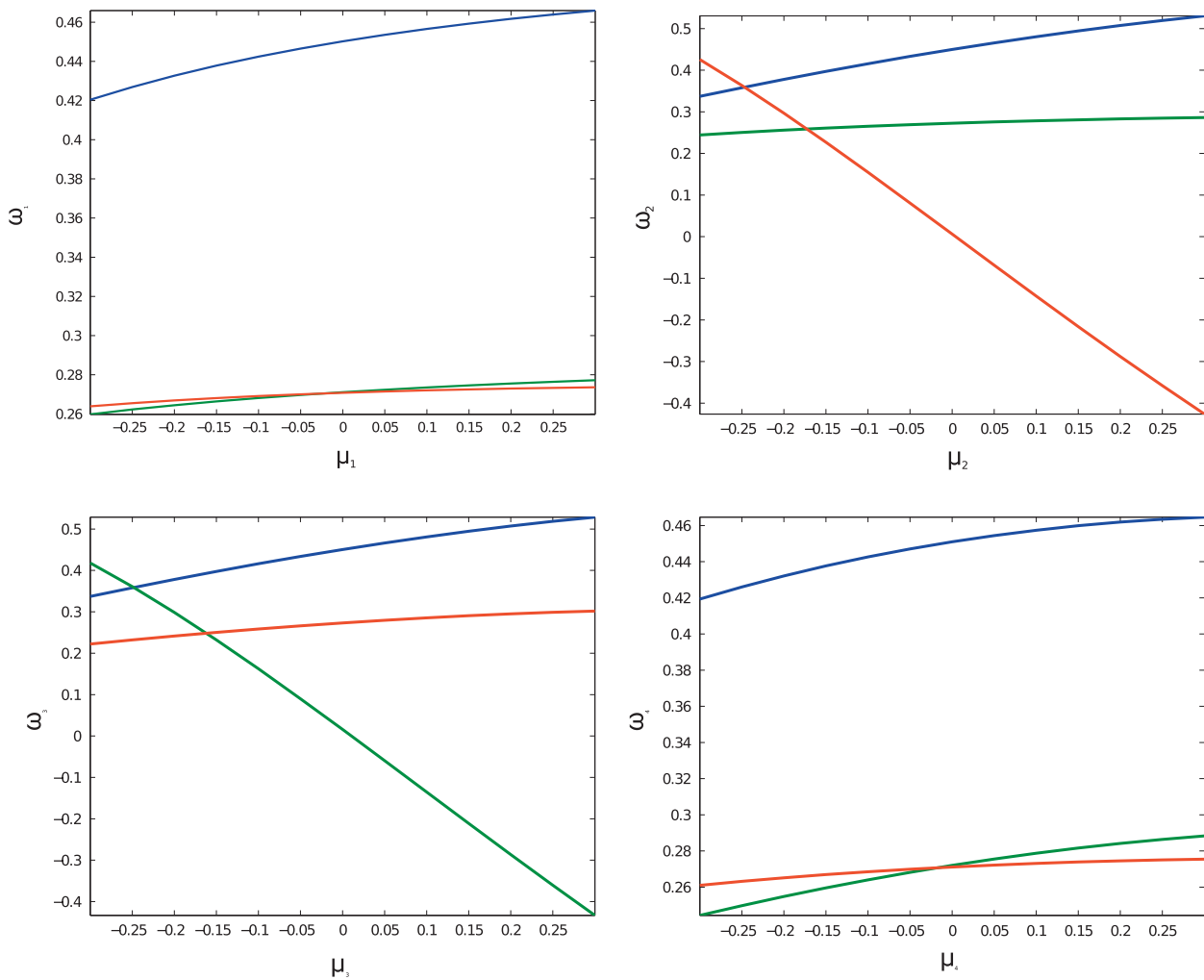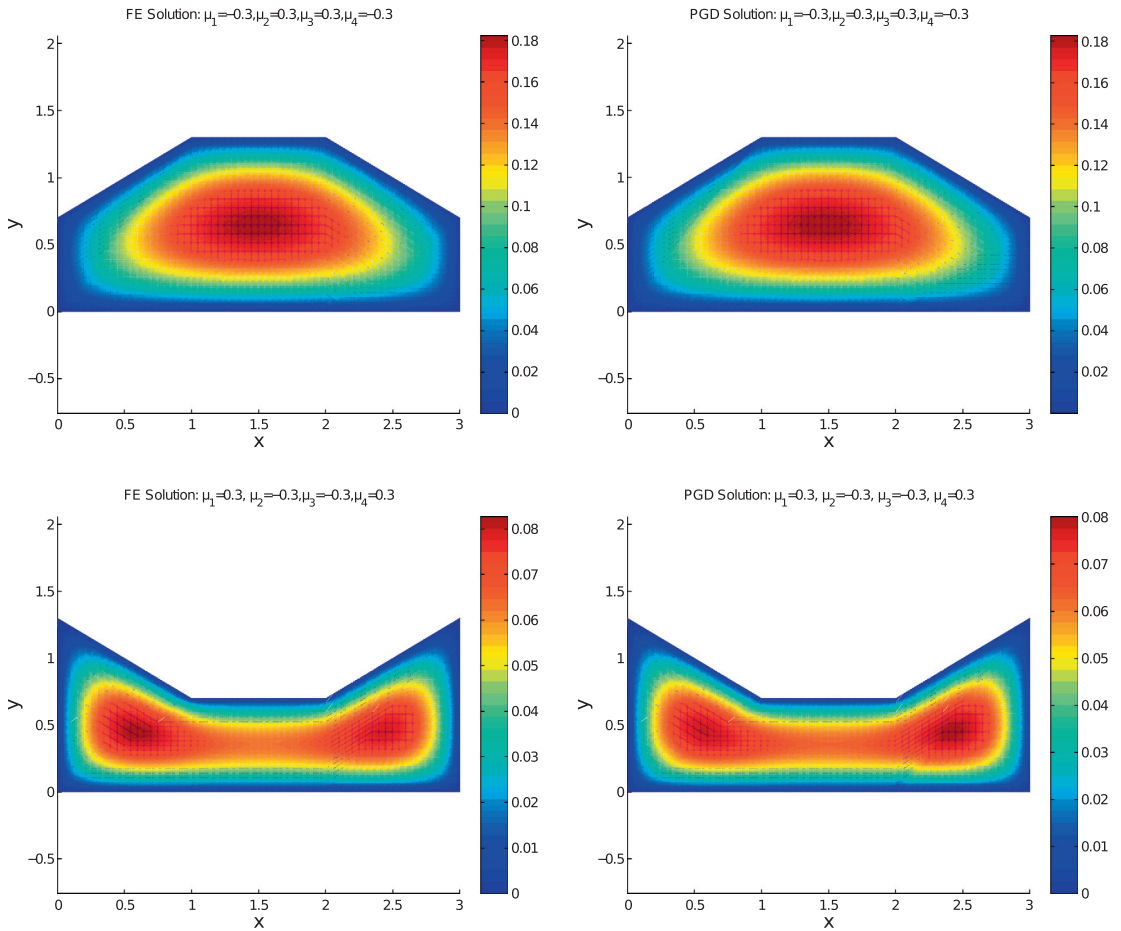
Fig. 6. Three most significant parametric modes $\omega_j^s(\mu_j)$, for $j = 1, \ldots, 4$.
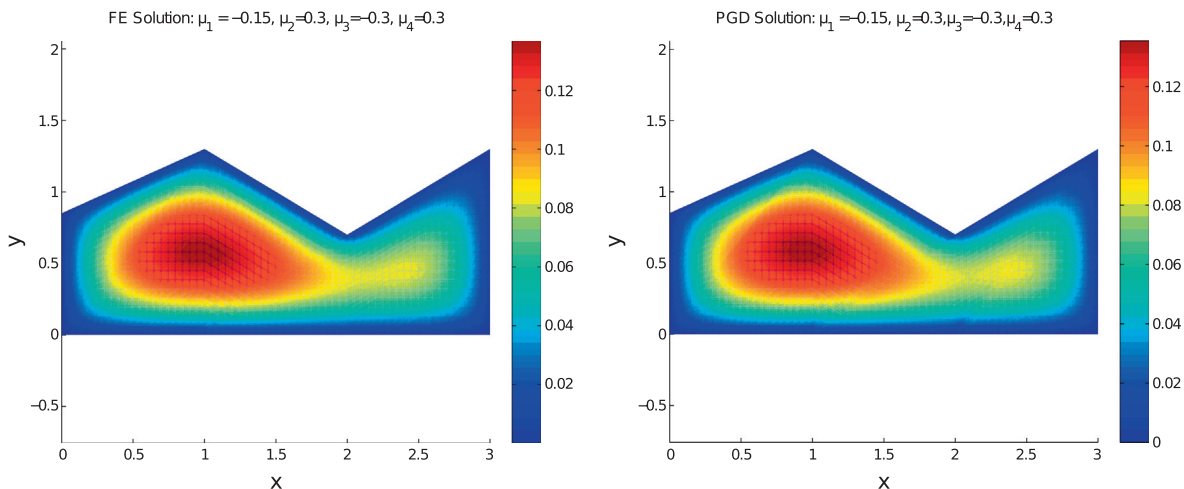
Fig. 7. Comparing the finite element solution (left) with the particularized PGD parametric solution (right).
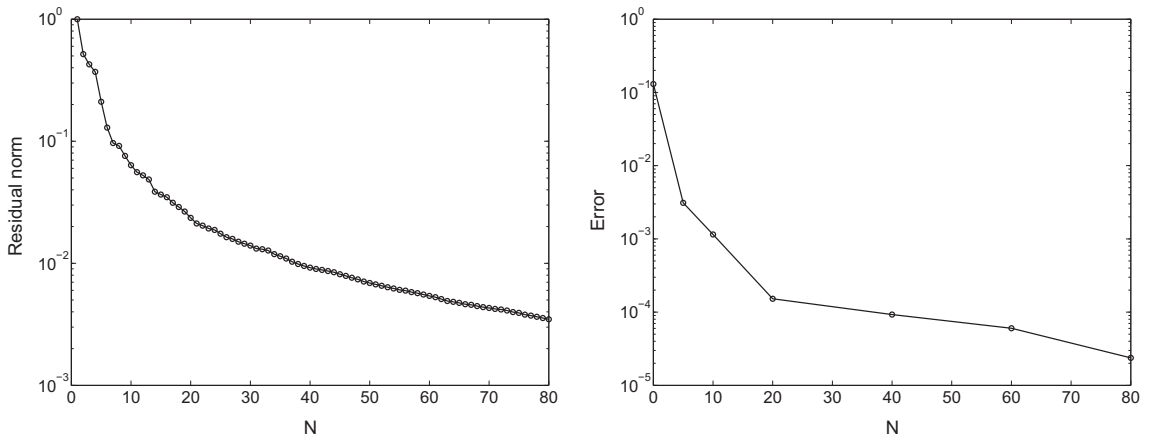
Fig. 8. Evolution of the total residual (left) and the error between PGD and FE (right) with the number of terms involved in the separated representation.

Fig. 9. Finite element (left) versus particularized PGD (right) solutions in two other geometries.
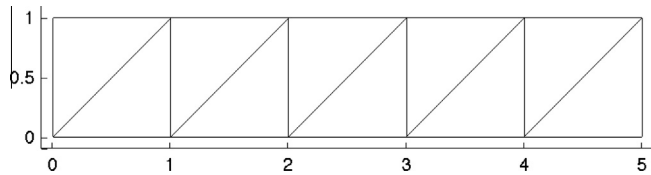
of iterations, however as the number of parameters and criteria increases the optimization processes becomes more and more time-consuming. Obviously, such a methodology is extremely sensitive to the cost of evaluation a solution for a set of parameters.

Suppose that the problem at hand requires to minimize the volume and maximize the heat flux along the top surface of the quadrilateral problem described in Section 5.1. Consequently, the objective is to optimize the shape of the parametrized

Table 2
Coordinates of the points describing the boundary of the perturbed domain.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
x	0	1	2	3	4	5	5	4	3	3	1	0
y	0	0	0	0	0	0	$1+\mu_1$	$1+\mu_2$	$1+\mu_3$	$1+\mu_4$	$1+\mu_5$	$1+\mu_6$

Fig. 10. Triangulation defining the geometrical transformation.
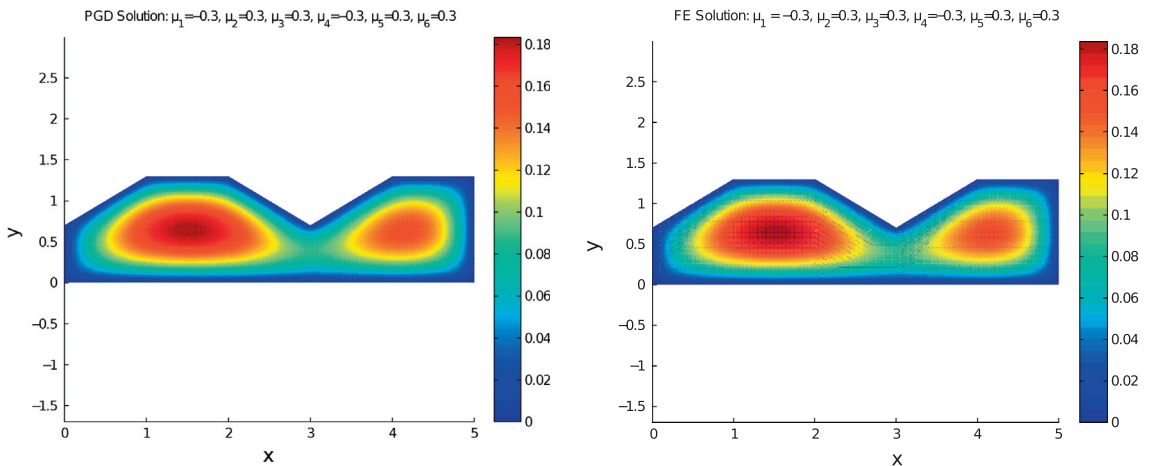
Fig. 11. Comparison of the PGD and finite element approximations for $\Omega(\boldsymbol{\mu})$ perturbed by $\mu_1 = -0.3$, $\mu_2 = 0.3$, $\mu_3 = 0.3$, $\mu_4 = -0.3$, $\mu_5 = 0.3$, and $\mu_6 = 0.3$.

quadrilateral domain minimizing volume $V(\boldsymbol{\mu})$ and maximizing heat flux through the upper surface, Γ^{Up}, defined by the segments joining points P_5, P_6, P_7 and P_8, namely

$$\Phi(\boldsymbol{\mu}) := \int_{\Gamma^{\mathrm{Up}}(\boldsymbol{\mu})} \boldsymbol{n} \cdot \nabla u(\boldsymbol{x}(\boldsymbol{\xi}, \boldsymbol{\mu})) \, d\Gamma$$

Recall that $\boldsymbol{\mu} \in I_1 \times \cdots \times I_4$ with $I_i = [-0.3, 0.3]$ for $i = 1, \ldots, 4$. The optimization iterative process can be performed solving for each iteration (for each set of parameters) problem (6) or evaluating for each iteration the approximated PGD solution (14).

A first alternative to optimize the shape defined by $\boldsymbol{\mu}$ is to combine both objectives (volume and head flux) in one single objective function. Consider for instance an objective function defined as

$$C_{\mathrm{ost}}(\boldsymbol{\mu}, \lambda) = \lambda V(\boldsymbol{\mu}) + (1 - \lambda)(1/\Phi(\boldsymbol{\mu})), \tag{16}$$

where the relative weight between both objectives, controlled by λ, must be user-defined a priori.

This particular optimization problem does not present major difficulties and can be readily solved with a large number of techniques for any given $\lambda \in]0, 1[$. Nevertheless, thanks to the PGD parametric solution, see the expression (14), a direct particularization of the parametric solution is readily obtained for each tentative value of the parameters at every iteration. This not the case of standard procedures, they would require to solve problem (6) at every iteration with the new set of tentative parameters. Moreover, with the PGD solution it is trivial to implement Newton's method to find the stationary point of (16). Whereas, classical approaches requiring the resolution of problem (6) at each iteration necessitate an extra effort to evaluate sensitivities in order to construct the tangent matrix, see for instance [1].

For instance, for $\lambda = 0.5$, after some four Newton iterations the optimal values are obtained, namely

$$(-0.6, -0.2, -0.6, -0.6) = \arg \min_{\boldsymbol{\mu} \in I_1 \times \cdots \times I_4} C_{\mathrm{ost}}(\boldsymbol{\mu}, 0.5).$$
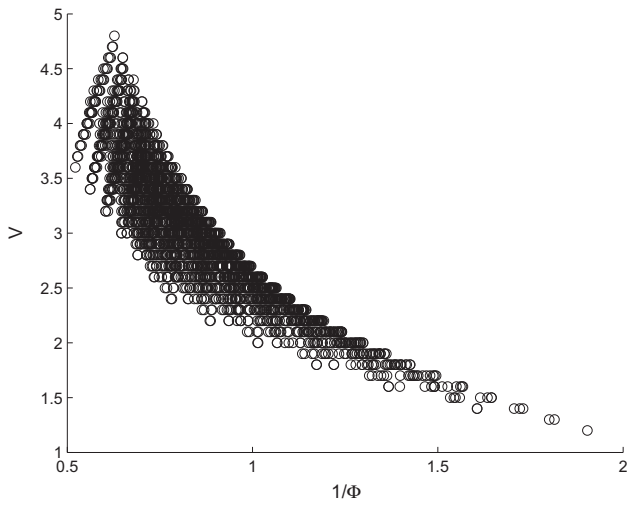
Fig. 12. Cloud of 10^4 design parameters in the multiobjective plane: $V(\boldsymbol{\mu})$ versus $1/\Phi(\boldsymbol{\mu})$.
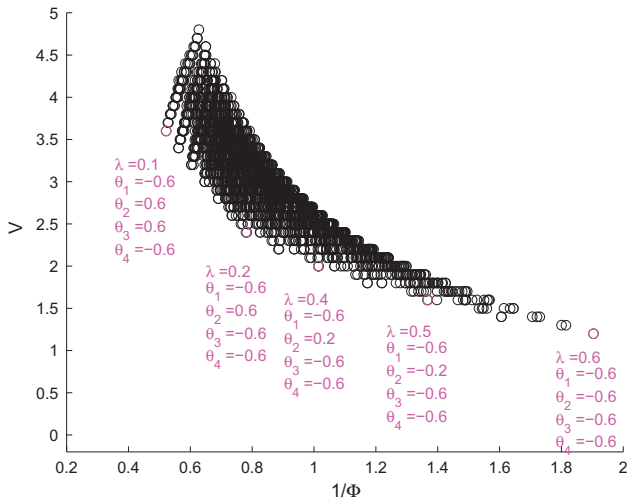
Fig. 13. Five optimal solutions in the Pareto's sense.

The advantages of having a parametric solution of a problem, such as (6), are more evident in multiobjective optimization, in particular, with the construction of Pareto's fronts. Suppose, that for each parameter μ_i, in this case, for $i = 1, \ldots, 4$ ten values are used to sample the range I_i. The total number of possible geometries is 10^4. Particularizing the solution, see (14), and then computing each objective, 10^4 times is not at all an expensive procedure. Thus for each possible geometry defined by a given $\boldsymbol{\mu}_i$ for $i = 1, \ldots, 10^4$ the volume, $V(\boldsymbol{\mu}_i)$, and the inverse of the flux $1/\Phi(\boldsymbol{\mu}_i)$ are readily evaluated. The resulting 10^4 points are depicted in Fig. 12. The convexity of the resulting cloud indicates that, in this case, the Pareto's front can be calculated by minimizing the cost function $C_{\mathrm{ost}}(\boldsymbol{\mu}, \lambda)$, see (16), for any of $\lambda \in]0, 1[$. This methodology is extremely easy and fast when the parametric solution is available. Fig. 13 depicts five points on the Pareto's front.

6. Conclusions

This work is a first attempt at considering parametric models in which the parameters controlling the geometry are treated as extra-coordinates. An approximation of the solution is obtained invoking a Proper Generalized Decomposition. This solution circumvents the curse of dimensionality do to the increased number of coordinates and induces an expression that is readily used in any optimization procedure. Moreover, Pareto fronts are also efficiently computed in multi-objective optimization.

A key ingredient is to overcome the difficulties related to the separated representation of the inverse of the Jacobian of the transformation. They have been alleviated by considering simpler mappings, as the ones associated with linear triangles that in many cases allows for exact separated representations of the inverse of the Jacobian.

The consideration of geometrical parameters coming for a CAD description or even the ones related to a isogeometric description of the domain boundary constitutes some of the works in progress.

References

[1] F. Navarrina, M. Casteleiro, A general methodological analysis for optimum design, Int. J. Numer. Methods Eng. 31 (1) (1991) 85–111.
[2] T. Belytschko, S.P. Xiao, C. Parimi, Topology optimization with implicit functions and regularization, Int. J. Numer. Methods Eng. 57 (8) (2003) 1177–1196.
[3] L.H. Olesen, F. Okkels, H. Bruus, A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow, Int. J. Numer. Methods Eng. 65 (7) (2006) 975–1001.
[4] G.H. Yoon, O. Sigmund, A monolithic approach for topology optimization of electrostatically actuated devices, Comput. Methods Appl. Mech. Eng. 197 (45–48) (2008) 4062–4075.
[5] C. Ghnatios, A. Ammar, A. Cimetiere, A. Hamdouni, A. Leygue, F. Chinesta, First steps in the space separated representation of models defined in complex domains, in: 11th Biennial Conference on Engineering Systems Design and Analysis, ASME, Nantes, 2012, ESDA2012-82489.
[6] F. van Keulen, R.T. Haftka, N.H. Kim, Review of options for structural design sensitivity analysis. I. Linear systems, Comput. Methods Appl. Mech. Eng. 194 (30–33) (2005) 3213–3243.
[7] D. Wang, Sensitivity analysis and shape optimization of a hole in a vibrating rectangular plate for eigenfrequency maximization, J. Eng. Mech. – ASCE 138 (6) (2012) 662–674.
[8] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, 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. 139 (3) (2006) 153–176.
[9] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, 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, J. Non-Newtonian Fluid Mech. 144 (2–3) (2007) 98–121.
[10] F. Chinesta, A. Ammar, E. Cueto, Recent advances and new challenges in the use of the Proper Generalized Decomposition for solving multidimensional models, Arch. Comput. Methods Eng. 17 (4) (2010) 327–350.
[11] F. Chinesta, P. Ladevèze, E. Cueto, A short review on model order reduction based on Proper Generalized Decomposition, Arch. Comput. Methods Eng. 18 (4) (2011) 395–404.
[12] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition, Math. Comput. Simul. 81 (4) (2010) 791–810.
[13] C. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, A. Poitou, P. Breitkopf, P. Villon, Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion, Compos. Pt. A – Appl. Sci. Manuf. 42 (9) (2011) 1169–1178.
[14] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, F. Chinesta, Proper Generalized Decomposition based dynamic data-driven control of thermal processes, Comput. Methods Appl. Mech. Eng. 213–216 (2012) 29–41.
[15] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, A. Poitou, Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity, Comput. Methods Appl. Mech. Eng. 201–204 (2012) 1–12.
[16] D. González, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, F. Chinesta, Proper Generalized Decomposition based dynamic data driven inverse identification, Math. Comput. Simul. 82 (9) (2012) 1677–1695.
[17] C. Heyberger, P.-A. Boucard, D. Néron, Multiparametric analysis within the Proper Generalized Decomposition framework, Comput. Mech. 49 (3) (2012) 277–289.
[18] P. Ladevèze, Nonlinear Computational Structural Mechanics. New Approaches and Non-incremental Methods of Calculation, Springer, New York, NY, 1999. translated from the French by James G. Simmonds.
[19] J.-C. Passieux, P. Ladevèze, D. Néron, A scalable time–space multiscale domain decomposition method: adaptive time scale separation, Comput. Mech. 46 (4) (2010) 621–633.
[20] P. Ladevèze, J.-C. Passieux, D. Néron, The LATIN multiscale computational method and the proper generalized decomposition, Comput. Methods Appl. Mech. Eng. 199 (21–22) (2010) 1287–1296.
[21] D. Néron, P. Ladevèze, Proper Generalized Decomposition for multiscale and multiphysics problems, Arch. Comput. Methods Eng. 17 (4) (2010) 351–372.
[22] F.E. Halabi, D. González, A. Chico-Roca, M. Doblaré, Multiparametric response surface construction by means of proper generalized decomposition: an extension of the parafac procedure, Comput. Methods Appl. Mech. Eng. 253 (2013) 543–557.
[23] A. Nouy, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations, Comput. Methods Appl. Mech. Eng. 199 (23–24) (2010) 1603–1626.
[24] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Comput. Methods Appl. Mech. Eng. 199 (25–28) (2010) 1872–1880.
[25] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing (Arch. Elektron. Rechnen) 5 (1970) 207–213.