Model reduction, data-based and advanced discretization in computational mechanics

Wavelet-based multiscale proper generalized decomposition

Angel Leon [a], Anais Barasinski [a], Emmanuelle Abisset-Chavanne [b], Elias Cueto [c], Francisco Chinesta [d,*]

[a] GeM Institute, École centrale de Nantes, 1, rue de la Noë, BP 92101, 44321 Nantes cedex 3, France
[b] High Performance Computing Institute & ESI GROUP Chair, École centrale de Nantes, 1, rue de la Noë, BP 92101, 44321 Nantes cedex 3, France
[c] I3A, University of Zaragoza, Maria de Luna s/n, 50018 Zaragoza, Spain
[d] PIMM Laboratory & ESI GROUP Chair, ENSAM ParisTech, 151, boulevard de l'Hôpital, 75013 Paris, France

ARTICLE INFO

Article history:
Received 4 September 2017
Accepted 7 February 2018
Available online 4 May 2018

Keywords:
Wavelets
Proper Generalized Decomposition
Multi-resolution
Multi-scale PGD

ABSTRACT

Separated representations at the heart of Proper Generalized Decomposition are constructed incrementally by minimizing the problem residual. However, the modes involved in the resulting decomposition do not exhibit a clear multi-scale character. In order to recover a multi-scale description of the solution within a separated representation framework, we study the use of wavelets for approximating the functions involved in the separated representation of the solution. We will prove that such an approach allows separating the different scales as well as taking profit from its multi-resolution behavior for defining adaptive strategies.

1. Introduction

Model Order Reduction – MOR – techniques allow nowadays solving, under real-time constraints, complex models. Intense research activities allowed reaching at present a certain maturity in the domain of model order reduction. Among the numerous references, the interested reader can refer to some review papers and books [1–4], covering three major MOR technologies: POD (Proper Orthogonal Decomposition), RB (Reduced Basis), and PGD (Proper Generalized Decomposition).

Proper Orthogonal Decomposition (POD) is a general technique for extracting the most significant characteristics of a system's behavior and representing them in a set of "POD basis vectors". These basis vectors then provide an efficient (typically low-dimensional) representation of the key system behavior, which proves useful in a variety of ways. The most common use is to project the system-governing equations onto the reduced-order subspace defined by the POD basis vectors. This yields an explicit POD reduced model that can be solved in place of the original system. The POD basis can also provide a low-dimensional description in which to perform parametric interpolation, infill missing or "gappy" data, perform model adaptation, or define hyper-reduction procedures [5]. There is an extensive literature and POD has seen broad application across fields. Some review of POD and its applications can be found in [6,7].

Another family of model reduction techniques lies in the use of Reduced Basis constructed by combining a greedy algorithm and "a posteriori" error indicators. As for the POD, the Reduced Basis method requires some amount of offline work, but then the reduced basis model can be used online for solving different models with control of the solution

* Corresponding author.
E-mail addresses: Angel.Leon-Collado@ec-nantes.fr (A. Leon), anais.barasinski@ec-nantes.fr (A. Barasinski), Emmanuelle.Abisset-chavanne@ec-nantes.fr (E. Abisset-Chavanne), ecueto@unizar.es (E. Cueto), Francisco.CHINESTA@ensam.edu (F. Chinesta).

accuracy, because of the availability of error bounds. When the error is unacceptably high, the reduced basis can be enriched by invoking a greedy adaption strategy [8,9].

Separated representations, at the heart of the so-called Proper Generalized Decomposition methods, are considered when solving at-hand partial differential equations by employing procedures based on the separation of variables. Then they were considered in quantum chemistry for approximating multidimensional quantum wave-function. In the 1980s, Pierre Ladevèze proposed the use of space-time separated representations of transient solutions involved in strongly nonlinear models, defining a non-incremental integration procedure [10,11]. Later, separated representations were employed for solving multidimensional models suffering the so-called curse of dimensionality [12,13], as well as in the context of stochastic modeling [14]. Then, they were extended to the separation of space coordinates, making possible the solution to models defined in degenerated domains, e.g., plate and shells [15], as well as for addressing parametric models where model parameters were considered as model extra-coordinates, making possible the offline calculation of the parametric solution, which can be viewed as a metamodel or a computational vademecum to be used online for real-time simulation, optimization, inverse analysis, and simulation-based control [2,16].

1.1. Separated representations

Within the PGD framework, four kinds of separated representations have been widely considered.

(i) *Space-time separated representations* that allowed the construction of efficient incremental and non-incremental integrators.
Within the standard finite element method, a space-time solution $u(\mathbf{x},t)$, $\mathbf{x}\in\Omega\subset\mathbb{R}^3$ and $t\in\mathcal{I}\subset\mathbb{R}$, of a transient problem is approximated from

$$u(\mathbf{x},t)\approx\sum_{i=1}^{\mathtt{M}}u(\mathbf{x}_i,t)N_i(\mathbf{x}) \tag{1}$$

where $\mathtt{M}$ is the number of nodes employed for interpolating the unknown field, located at positions $\mathbf{x}_i$, and $N_i(\mathbf{x})$ the so-called shape functions. Because of the interpolative property of the shape functions, the approximation coefficients correspond to the nodal value of the approximated field, $u(\mathbf{x}_i,t)$. Thus, in general, when solving a nonlinear problem, at least a linear system of size $\mathtt{M}$ must be solved at each time step. When considering $\mathtt{P}$ time steps ($\mathtt{P}$ can reach several millions), the complexity grows very fast.
When considering POD-based model order reduction, the solution is projected into the reduced basis composed of functions $\{\phi_1(\mathbf{x}),\cdots,\phi_{\mathtt{R}}(\mathbf{x})\}$ extracted from some collected snapshots of the problem solution, with in general $\mathtt{R}\ll\mathtt{M}$, and consequently the solution approximation reads

$$u(\mathbf{x},t)\approx\sum_{i=1}^{\mathtt{R}}\xi_i(t)\phi_i(\mathbf{x}) \tag{2}$$

which requires the resolution of linear systems of size $\mathtt{R}$ instead of the ones of size $\mathtt{M}$ characteristic of finite element solutions. The use of a reduced basis implies in many cases impressive computing-time savings.
Approximations (1) or (2) imply a finite sum of time-dependent coefficients and space functions. The last are assumed known; they consist of the usual finite element shape functions or the modes extracted by applying, for example, Proper Orthogonal Decomposition – POD. A step forward could consist in assuming space functions to be also unknown and in computing both time and space functions on the fly. In this case, the approximation reads

$$u(\mathbf{x},t)\approx\sum_{i=1}^{\mathtt{N}}T_i(t)X_i(\mathbf{x}) \tag{3}$$

Because both functions involved in approximation (3) are unknown, it defines a nonlinear problem whose solution requires an appropriate linearization strategy. The interested reader can refer to [17] and the references therein for additional details on the separated representation constructor.
Expression (3) evidences that the solution procedure requires the resolution of about $\mathtt{N}$ problems, with $\mathtt{N}\ll\mathtt{M}$ and $\mathtt{N}\sim\mathtt{R}$ (in fact a bit more because of the nonlinearity induced by separated representations) involving the space coordinates (in general three-dimensional – 3D – and whose associated discrete systems are of size $\mathtt{M}$) for computing the space functions $X_i(\mathbf{x})$ and about $\mathtt{N}$ one-dimensional – 1D – problems for calculating the time functions $T_i(t)$. Due to the fact that the computing cost related to the solution of 1D problems is negligible with respect to the solution of 3D problems, the resulting computational complexity reduces drastically, scaling with $\mathtt{N}$ instead of $\mathtt{P}$.

(ii) *Space separation* allowed addressing multi-physics problems defined in degenerated geometries in which at least one of its dimensions remains much smaller that the other ones (e.g., beams, plates, shells, laminates...) or processes involving additive layers (e.g., automated tape placement, 3D printing, or additive manufacturing).

If domain Ω can be decomposed as $\Omega = \Omega_x \times \Omega_y \times \Omega_z$, the solution $u(x, y, z)$ could be approximated by using the separated representation

$$u(x, y, z) \approx \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) \tag{4}$$

which allows calculating the 3D solution from a sequence of 1D problems.
For some geometries, as the ones associated with plates or shells, in-plane–out-of-plane separated representation becomes specially suitable,

$$u(x, y, z) \approx \sum_{i=1}^{N} X_i(x, y) Z_i(z) \tag{5}$$

where the 3D complexity is reduced to the one characteristic of 2D problems, the ones related to the calculation of in-plane functions $X_i(x, y)$.

(iii) *Space–time–parameter separated representations* allowed constructing the so-called computational vademecums (also known as abacus, virtual charts, nomograms...) efficiently considered for multiple purposes: simulation, optimization, inverse analysis, uncertainty propagation and simulation-based control, all of them under the real-time constraint.
When the unknown field involves space, time, and a series of parameters $\mu_1, \ldots, \mu_Q$, its associated separated representation reads

$$u(\mathbf{x}, t, \mu_1, \ldots, \mu_Q) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) T_i(t) \prod_{j=1}^{Q} \Gamma_i^j(\mu_j) \tag{6}$$

(iv) *Separated representations of intrinsically multidimensional models* involving differential operators applying on time, space, and a series of conformation coordinates $c_1, \ldots, c_C$. In this case, the solution is approximated according to

$$u(\mathbf{x}, t, c_1, \ldots, c_C) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) T_i(t) \prod_{j=1}^{C} C_i^j(c_j) \tag{7}$$

1.2. Paper motivation

This work focuses on the separated representations previously discussed and contributes in two different ways. First, in general the calculation of functions involved in the finite sum representations is performed on a given approximation basis. Thus, the possible multi-scale character of the individual modes (functions involved in the finite sum) and/or the reconstructed solution (the finite sum itself) is not addressed.

Even if today it is accepted within the PGD community that both the number of terms in the finite sum, and the number of nodes used for discretizing each function involved in the former are of major relevance for ensuring accuracy, and even if different error estimators have been proposed in this context [18–20], the multi-scale nature of both, the terms in the sum and the sum itself, remain almost unexplored. An adaptive progressive PGD strategy was proposed in [21].

This work considers the use of a naturally multi-scale approach, a wavelet-based approximation technique, for approximating the different functions involved in the separated representation. Then, recombining the different scales involved in this approximation allows one to capture the multi-scale character of the solution. Even if, in our knowledge, this kind of approximations was employed in a Galerkin setting, as referred to later, its consideration within the PGD framework remains unexplored.

The next section revisits the PGD constructor and the wavelet-based multi-resolution analysis that will be applied in Section 3 within the PGD constructor described above. Then Section 4 will illustrate some potential applications of the proposed numerical technique.

2. Methods

2.1. PGD constructor at a glance

Consider the solution to the Poisson equation

$$\Delta u(x, y) = f(x, y) \tag{8}$$

in a two-dimensional rectangular domain $\Omega = \Omega_x \times \Omega_y = (0, L) \times (0, H)$. We specify homogeneous Dirichlet boundary conditions for the unknown field $u(x, y)$, i.e. $u(x, y)$ vanishes at the domain boundary Γ. Furthermore, we assume that the source term f is constant over the domain Ω.

For all suitable test functions u^*, the weighted residual form of (8) reads

$$\int_{\Omega_x \times \Omega_y} u^* \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - f \right) \mathrm{d}x \, \mathrm{d}y = 0 \tag{9}$$

Our goal is to obtain a PGD approximate solution to (8) in the separated form

$$u(x, y) = \sum_{i=1}^{\mathrm{N}} X_i(x) \cdot Y_i(y) \tag{10}$$

We shall do so by computing each term of the expansion one at a time, thus enriching the PGD approximation until a suitable convergence criterion is satisfied. Thus, at each enrichment step n ($n \geq 1$), we have already computed the $n-1$ first terms of the PGD approximation

$$u^{n-1}(x, y) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) \tag{11}$$

We now wish to compute the next term $X_n(x) \cdot Y_n(y)$ to obtain the enriched PGD solution

$$u^n(x, y) = u^{n-1}(x, y) + X_n(x) \cdot Y_n(y) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) + X_n(x) \cdot Y_n(y) \tag{12}$$

Both functions $X_n(x)$ and $Y_n(y)$ are unknown at the current enrichment step n, and they appear in the form of a product. The resulting problem is thus non-linear, and a suitable iterative scheme is required. We shall use the index p to denote a particular iteration. At enrichment step n, the PGD approximation $u^{n,p}$ obtained at iteration p thus reads

$$u^{n,p}(x, y) = u^{n-1}(x, y) + X_n^p(x) \cdot Y_n^p(y) \tag{13}$$

and the algorithm proceeds by (i) calculating $X_n^p(x)$ from $Y_n^{p-1}(y)$, then (ii) updating $Y_n^p(y)$ from the just-computed $X_n^p(x)$; and finally (iii) checking the convergence by evaluating $\| X_n^p(x) \cdot Y_n^p(y) - X_n^{p-1}(x) \cdot Y_n^{p-1}(y) \|$.

In the first step, by introducing the approximate

$$u^{n,p}(x, y) = u^{n-1}(x, y) + X_n^p(x) \cdot Y_n^{p-1}(y) \tag{14}$$

as well as the weight function

$$u^*(x, y) = X_n^*(x) \cdot Y_n^{p-1}(y) \tag{15}$$

into the problem integral form (9), after integrating over Ω_y, one finds

$$\int_{\Omega_x} X_n^* \cdot \left(\alpha^x \frac{\mathrm{d}^2 X_n^p}{\mathrm{d}x^2} + \beta^x X_n^p \right) \mathrm{d}x = - \int_{\Omega_x} X_n^* \cdot \sum_{i=1}^{n-1} \left(\gamma_i^x \frac{\mathrm{d}^2 X_i}{\mathrm{d}x^2} + \delta_i^x X_i \right) \mathrm{d}x + \int_{\Omega_x} X_n^* \cdot \xi^x \, \mathrm{d}x \tag{16}$$

where the coefficients result from the integration over Ω_y of functions depending on the y-coordinate. For more details, the interested reader can refer to [17]. The above one-dimensional problem is then discretized by using a standard mesh-based discretization technique, as for example the finite element method.

Having thus computed $X_n^p(x)$, we are now ready to proceed with the second step of iteration p, which updates $Y_n^p(y)$ from the just-computed $X_n^p(x)$. The procedure exactly mirrors what we have done above. Indeed, we simply exchange the roles played by all relevant functions of x and y, starting from the solution approximate

$$u^{n,p}(x, y) = \sum_{i=1}^{n-1} X_i(x) \cdot Y_i(y) + X_n^p(x) \cdot Y_n^p(y) \tag{17}$$

which finally leads to the one-dimensional problem involving the unknown field $Y_n^p(y)$:

$$\int_{\Omega_y} Y_n^* \cdot \left(\alpha^y \frac{\mathrm{d}^2 Y_n^p}{\mathrm{d}y^2} + \beta^y Y_n^p \right) \mathrm{d}y = - \int_{\Omega_y} Y_n^* \cdot \sum_{i=1}^{n-1} \left(\gamma_i^y \frac{\mathrm{d}^2 Y_i}{\mathrm{d}y^2} + \delta_i^y Y_i \right) \mathrm{d}y + \int_{\Omega_y} Y_n^* \cdot \xi^y \, \mathrm{d}y \tag{18}$$

It is important to realize that the original two-dimensional Poisson equation defined over $\Omega = \Omega_x \times \Omega_y$ has been transformed within the PGD framework into a series of *decoupled one-dimensional problems* formulated in Ω_x and Ω_y. The procedure can be generalized when considering D-dimensional problems.

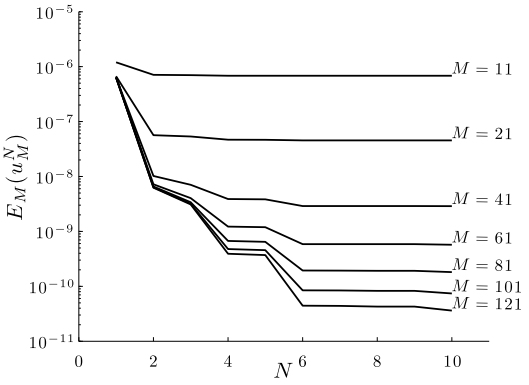

Fig. 1. PGD error $E_M(u_M^N)$ as a function of the number of enrichment steps N for different numbers M of discretization points.

As previously indicated, the functions $X_n(x)$ and $Y_n(y)$ are approximated using standard finite element approximations or orthogonal polynomials (spectral approximations). Thus, in first approximation one could expect that the accuracy of the computed solution should depend on the number of terms involved in the finite sum $\mathtt{N}$ and the number of nodes $\mathtt{M}$ (or the polynomial degree) considered for approximating the different functions involved in the finite sum, $X_n(x)$ and $Y_n(y)$ in the previous case study.

In [17], the authors considered the Poisson equation (8) on a two-dimensional rectangular domain $\Omega = \Omega_x \times \Omega_y = (0, 2) \times (0, 1)$ with $f = 1$, which has as exact solution

$$u_{\text{ex}}(x, y) = \sum_{m,n \text{ odd}} \frac{64}{\pi^4 \left(4n^2 + m^2\right)} \sin\left(\frac{m\pi x}{2}\right) \sin(n\pi y) \tag{19}$$

Functions $X_i(x)$ and $Y_i(y)$ were sought on a uniform one-dimensional grid with $\mathtt{M}$ points. The solution convergence (with respect to the exact one) was evaluated by using the error

$$E_{\mathtt{M}}(u_{\mathtt{M}}^{\mathtt{N}}) = \int_0^1 \int_0^2 \left(u_{\text{ex}}(x, y) - u_{\mathtt{M}}^{\mathtt{N}}(x, y)\right)^2 \, \mathrm{d}x \, \mathrm{d}y \tag{20}$$

where integrals were performed numerically and $u_{\mathtt{M}}^{\mathtt{N}}$ refers to the solution obtained using a finite sum composed of $\mathtt{N}$ terms and the involved one-dimensional functions were approximated by using $\mathtt{M}$ nodes. Fig. 1 depicts the PGD error $E_{\mathtt{M}}(u_{\mathtt{M}}^{\mathtt{N}})$ as a function of the number of enrichment steps $\mathtt{N}$ for different numbers $\mathtt{M}$ of discretization points. This figure proves that both $\mathtt{M}$ and $\mathtt{N}$ are relevant to ensure convergence, and that both must be considered together, because, for a given mesh, and independently of the number of terms considered in the finite sum, the error reaches a plateau. In the other sense, even a very fine mesh is unable to capture the solution if the number of terms in the finite sum is insufficient.

2.2. Wavelet-based multi-resolution analysis and adaptive approximations

Multi-resolution analysis is based on the construction of a series of embedded subspaces $V_j \subset V_{j+1}$, $\forall j \in \mathbb{Z}$, i.e.

$$\{\emptyset\} \cdots \subset V_{j-1} \subset V_j \subset V_{j+1} \cdots \subset L^2(\mathbb{R}) \tag{21}$$

where each subspace is spanned by the integer translation of a single function, the scaling function $\phi(x)$ [22].

If $\phi(x) \in V_0$, the functions $\phi_{0,k} = \phi(x - k)$ constitute a basis of V_0, and the functions $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$ define a basis of subspace V_j. Thus, the projection of the function $f(x)$ in V_j reads

$$\mathcal{P}_j f(x) = \sum_{k=-\infty}^{k=+\infty} c_{j,k}\phi_{j,k}(x) \tag{22}$$

The fact that $V_0 \subset V_1$ allows writing

$$\phi(x) = \sum_{k=-\infty}^{k=+\infty} a_k \phi(2x - k) \tag{23}$$

and because $V_j \subset V_{j+1}$ we can define the orthogonal complement W_j, such that

$$V_{j+1} = V_j \oplus W_j \tag{24}$$

with $V_j \perp W_j$. An important consequence is that

$$\bigoplus_{j\in\mathbb{Z}} W_j = L^2(\mathbb{R}) \tag{25}$$

or

$$V_0 \oplus \bigoplus_{j\in\mathbb{N}} W_j = L^2(\mathbb{R}) \tag{26}$$

Spaces W_j are also spanned by the integer translation of a single function, the wavelet function $\psi(x)$, defining the bases at each scale: $\psi_{j,k} = 2^{j/2}\psi(2^j x - k)$. Because $W_0 \subset V_1$, we can write

$$\psi(x) = \sum_{k=-\infty}^{k=+\infty} b_k \phi(2x - k) \tag{27}$$

In what follows we are considering Daubechies wavelets [23] for approximating the different functions involved in the separated representation of the problem solution. This choice is motivated by their compact support and regularity (approximation consistency). In opposition to usual approximation techniques, its use allows space and frequency localization, the latter because of its multiscale character and the former due to its compact support (in opposition to approximations based on Fourier polynomials). The approximation is fully defined by the coefficients a_k in Eq. (23). The coefficients b_k depend on a_k from $b_k = (-1)^k a_{1-k}$.

The coefficients a_k result from the following conditions:

(i) the normalization condition,

$$\int_{-\infty}^{\infty} \phi(x)\, \mathrm{d}x = 1 \tag{28}$$

which, taking into account Eq. (23), leads to

$$\sum_{k=-\infty}^{k=\infty} a_k = 2 \tag{29}$$

(ii) the orthogonality condition between $\phi(x)$ and its integer translates into

$$\int_{-\infty}^{\infty} \phi(x)\phi(x-l)\, \mathrm{d}x = \delta_{0l} \tag{30}$$

with δ the Kroenecker delta function. Taking into account Eq. (23), the previous equation results in

$$\sum_{k=-\infty}^{k=\infty} a_k a_{k+2l} = 2\delta_{0l}, \quad \forall l \in \mathbb{Z} \tag{31}$$

(iii) if we consider a choice with $\mathcal{N}$ coefficients, the previous conditions only provide $\mathcal{N}/2 + 1$ equations. Thus, other $\mathcal{N}/2 - 1$ conditions should be added in order to compute the $\mathcal{N}$ coefficients. One common route consists in enforcing that the scaling function exactly represents polynomials of order $\mathcal{M} = \mathcal{N}/2$, i.e.

$$f(x) = a_0 + a_1 x + \cdots + a_{\mathcal{M}-1} x^{\mathcal{M}-1} \tag{32}$$

from which, by using

$$f(x) = \sum_{k=-\infty}^{k=\infty} c_k \phi(x - k) \tag{33}$$

and the fact that the wavelet function $\psi(x)$ is orthogonal to the scaling translates $\phi(x-k)$, it results

$$\langle f(x), \psi(x)\rangle = \sum_{k=-\infty}^{k=\infty} c_k \langle \phi(x-k), \psi(x)\rangle = 0 \tag{34}$$

with

$$\langle g(x), h(x)\rangle = \int_{-\infty}^{\infty} g(x)\, h(x)\, \mathrm{d}x \tag{35}$$

By replacing $f(x)$ by its expression (32) into Eq. (34) leads to

$$a_0\langle \psi(x), 1\rangle + a_1\langle \psi(x), x\rangle + a_{\mathcal{M}-1}\left\langle \psi(x), x^{\mathcal{M}-1}\right\rangle = 0 \tag{36}$$

which applies for any value of coefficients $a_0, \cdots, a_{\mathcal{M}-1}$. Thus, by choosing $a_l = 1$ and $a_m = 0,\ \forall m \neq l$, it results

$$\left\langle \psi(x), x^l\right\rangle = 0, \quad l = 0, 1, \cdots, \mathcal{M}-1 \tag{37}$$

implying that the $\mathcal{M}$ first moments of the wavelet function vanish. This condition can be proved to be equivalent to

$$\sum_{k=-\infty}^{k=\infty} (-1)^k a_k k^l = 0, \quad l = 0, 1, \cdots, \mathcal{M}-1 \tag{38}$$

Now, all the conditions above allow determining the $\mathcal{N}$ coefficients that determine completely the approximation.

In order to discretize the weak form of a given problem, one must have access to the scaling and wavelet function and their derivatives at any point (in particular at the integration points considered for evaluating the integrals involved in the problem weak form). For that purpose, we start by considering

$$\phi(x) = a_0\phi(2x) + a_1\phi(2x-1) + \cdots + a_{\mathcal{N}-1}\phi(2x-\mathcal{N}+1) \tag{39}$$

or, taking into account that the support of the scaling function is $[0, N]$, it results

$$\begin{cases} \phi(0) = a_0\phi(0) \\ \phi(1) = a_0\phi(2) + a_1\phi(1) + a_2\phi(0) \\ \phi(2) = a_0\phi(4) + a_1\phi(3) + a_2\phi(2) + a_3\phi(1) + a_4\phi(0) \\ \vdots \end{cases} \tag{40}$$

which defines an eigenvalue problem giving access to the value of the scaling function at the integers. Now, in order to obtain its value at the so-called dyadic points, we make use of

$$\phi(x/2) = \sum_{k=-\infty}^{k=\infty} a_K \phi(x-k) \tag{41}$$

and from those we can compute $\phi(x/4)$ and so on.

Now, for the derivatives, we proceed in a similar manner, by taking the derivative of Eq. (39), which allows us, by following the same rationale and solving the associated eigenproblem, to calculate the derivative at the integer points, and from this the one at the dyadic points, and so on.

Approximations based on wavelets have been successfully considered for discretizing partial differential equations, most of the time by using Galerkin formulations [22,24–27].

If, for a while, we consider the one-dimensional problem involving the unknown field $u(x)$

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(K(x)\frac{\mathrm{d}u}{\mathrm{d}x}\right) = 0 \tag{42}$$

with $x \in [0, 1]$, $u(x=0) = 0$, $u(x=1) = 1$ and the model parameter $K(x)$ defined from

$$K(x) = \begin{cases} K_{\max} & \text{if } x \in [0.15, 0.85] \\ K_{\min} & \text{elsewhere} \end{cases} \tag{43}$$

it is expected that the continuous field $u(x)$ exhibits a discontinuity in its derivative at points $x = 0.15$ and $x = 0.85$, because of the continuity of the fluxes, i.e.

$$K(0.15-\epsilon)\left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x=0.15-\epsilon} = K(0.15+\epsilon)\left.\frac{\mathrm{d}u}{\mathrm{d}x}\right|_{x=0.15+\epsilon} \tag{44}$$

and similarly for $x = 0.85$, with ϵ a small enough constant.

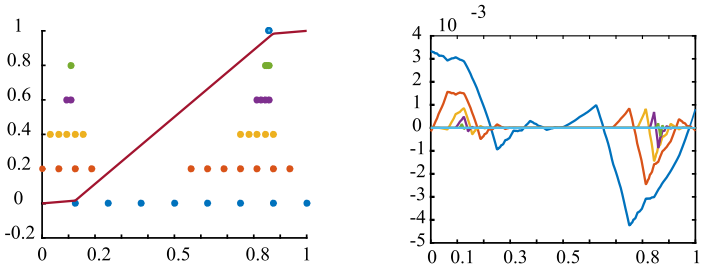

Fig. 2. Adaptive approximation strategy.

When using wavelet-based approximations, the unknown field can be expressed at the lowest level by using the scaling functions

$$u^0(x) = \sum_k c_{0,k}\phi(x-k) \tag{45}$$

The problem solution (described later) consists of the set of coefficients $c_{0,k}$. The solution at the next level is written by combining both the scaling and wavelet function at the lowest level, i.e.

$$u^1(x) = \sum_k c_{0,k}\phi(x-k) + \sum_k d_{0,k}\psi(x-k) \tag{46}$$

whose solution consists now of the coefficients $c_{0,k}$ and $d_{0,k}$.

Because of the multi-resolution property of the employed approximation, one expects that the coefficients $d_{0,k}$ become higher at locations k where the lowest-level approximation $u^0(x)$ based on the use of the lowest level scaling functions is not able to represent the solution with sufficient accuracy. Thus, we could define a threshold value κ, allowing us to identify two sets $\mathcal{S}_1^-$ and $\mathcal{S}_1^+$, composed respectively by integers k such that $d_{0,k} \leq \kappa$ and $d_{0,k} > \kappa$.

Now, the next approximation level $u^2(x)$ is defined by adding to the previous one $u^1(x)$ only the wavelet functions related to points in $\mathcal{S}_1^+$ and the ones having a natural neighbor in $\mathcal{S}_1^+$. The resulting extended set is denoted by $\tilde{\mathcal{S}}_1^+$, and now the approximation $u^2(x)$ reads

$$u^2(x) = u^1(x) + \sum_{k\in\tilde{\mathcal{S}}_1^+} d_{1,k}\psi_{1,k}(x) \tag{47}$$

After calculating coefficients $d_{1,k}$, they are classified in sets $\mathcal{S}_2^-$ and $\mathcal{S}_2^+$, and the adaptive approximation continues.

Fig. 2 depicts the evolution of the approximation basis on the left (with the red line being the computed solution), where the points at the lowest level (blue ones) are related to the uniform lowest-level approximation (based on the use of the scaling functions). Then, the second level, red points, are the ones in $\tilde{\mathcal{S}}_1^+$, and so on. On the right (Fig. 2), the different contributions to the solution are depicted: $u^1(x)$ (blue curve) as well as the adaptive contributions (for example, the red curve represents $u^2(x) - u^1(x)$, whose support is defined by $\tilde{\mathcal{S}}_1^+$).

In the next section, the wavelet approximations will be considered for discretizing PDEs within the separated representation format characteristic of PGD methods.

3. Multi-scale Proper Generalized Decomposition

For the sake of simplicity, we consider the heat equation

$$\nabla\left(K(x,y)\nabla u(x,y)\right) = f(x,y) \tag{48}$$

where $K(x,y)$ represents the material conductivity, assumed isotropic.

Problem (48) is defined in $\Omega = \Omega_x \times \Omega_y = (0,L)\times(0,H)$ and where without loss of generality the homogeneous Dirichlet boundary condition is enforced on the domain boundary $\Gamma \equiv \partial\Omega$.

The weak form related to Eq. (48) reads

$$\int_{\Omega_x\times\Omega_y} \nabla u^* \cdot \left(K\nabla u(x,y) + f(x,y)\right) \mathrm{d}x\,\mathrm{d}y = 0 \tag{49}$$

whose solution, as described in Section 1, is searched under the separated form

$$u^n(x,y) = \sum_{i=1}^{n} X_i(x)\cdot Y_i(y) \tag{50}$$

which implies the alternative resolution of two one-dimensional problems, the first one for calculating $X_i(x)$, and the second one for calculating $Y_i(y)$, similarly to Eqs. (16) and (18).

When considering wavelet-based approximations, the functions $X_i(x)$ and $Y_i(y)$ at the enrichment iteration n and the nonlinear iteration p, involved in the one-dimensional problems of the PGD separated representation constructor, are approximated according to:

$$\begin{cases} X_n^p(x) = 2^{j_0/2}\sum_k X_{j_0,k}^{p,n}\phi\left(2^{j_0}x-k\right) + \sum_{j=j_0}^{J} 2^{j/2}\sum_k X_{j,k}^{p,n}\psi\left(2^{j}x-k\right) \\ Y_n^p(y) = 2^{j_0/2}\sum_k Y_{j_0,k}^{p,n}\phi\left(2^{j_0}y-k\right) + \sum_{j=j_0}^{J} 2^{j/2}\sum_k Y_{j,k}^{p,n}\psi\left(2^{j}y-k\right) \\ X_n^*(x) = 2^{j_0/2}\sum_k X_{j_0,k}^{*,n}\phi\left(2^{j_0}x-k\right) + \sum_{j=j_0}^{J} 2^{j/2}\sum_k X_{j,k}^{*,n}\psi\left(2^{j}x-k\right) \\ Y_n^*(y) = 2^{j_0/2}\sum_k Y_{j_0,k}^{*,n}\phi\left(2^{j_0}y-k\right) + \sum_{j=j_0}^{J} 2^{j/2}\sum_k Y_{j,k}^{*,n}\psi\left(2^{j}y-k\right) \\ X_i(x) = 2^{j_0/2}\sum_k X_{j_0,k}^{i}\phi\left(2^{j_0}x-k\right) + \sum_{j=j_0}^{J} 2^{j/2}\sum_k X_{j,k}^{i}\psi\left(2^{j}x-k\right) \\ Y_i(y) = 2^{j_0/2}\sum_k Y_{j_0,k}^{i}\phi\left(2^{j_0}y-k\right) + \sum_{j=j_0}^{J} 2^{j/2}\sum_k Y_{j,k}^{i}\psi\left(2^{j}y-k\right) \end{cases} \tag{51}$$

where j_0 refers to the lowest level and where the coefficients affecting the different scaling and wavelet functions are the unknowns.

It is important to note that, even if the approximates (51), as written, involve all the translations k at each level j, the multi-resolution analysis allows considering, as previously discussed, only the wavelet functions located in the regions in which they contribute to the solution improvement. Moreover, even if the largest scale is assumed to be the same for all the functions involved in the separated representation, J for all them, in general each function at each level could involve a different number of scales.

As previously described, all these functions, as well as their derivatives, can be computed at dyadic points, until approaching sufficiently the considered point. Thus, by injecting the previous approximates into the weak form of the one-dimensional problems resulting from the PGD discretization of Eq. (49) and using an integration quadrature, the approximation coefficients can be obtained.

It is important to mention that Eq. (49) also involves the conductivity $K(x, y)$, which should be separated. For this purpose, we proceed as described in our former works (e.g., [17]) by constructing its separated representation

$$K(x,y) \approx \sum_{i=1}^{\mathrm{K}} \mathcal{K}_i^x(x)\mathcal{K}_i^y(y) \tag{52}$$

The separation could be performed by using a standard SVD – Singular Value Decomposition. Another possibility consists in using the PGD (see Chapter 3 in [17]), which consists in solving the problem

$$\tilde{K}(x,y) - K(x,y) = 0 \tag{53}$$

with

$$\tilde{K}(x,y) = \sum_{i=1}^{\mathrm{K}} \mathcal{K}_i^x(x)\mathcal{K}_i^y(y) \tag{54}$$

whose integral form reads

$$\int_{\Omega_x\times\Omega_y} \tilde{K}^*(x,y)\left(\tilde{K}(x,t) - \sum_{i=1}^{\mathrm{K}} \mathcal{K}_i^x(x)\mathcal{K}_i^y(y)\right)\mathrm{d}x\,\mathrm{d}y = 0 \tag{55}$$

Again, this equation is solved by using the PGD constructor and now, because no regularity is needed in the representation of the conductivity, the Haar's wavelet is chosen, i.e. used for approximating functions involved in the conductivity separated representation $\mathcal{K}_i^x(x)$ and $\mathcal{K}_i^y(y)$.

The main issue when considering the Daubechies and Haar representations of the temperature field $u(x, y)$ and the conductivity $K(x, y)$ is that the integrals in Eq. (49) involve the product of two or three scaling and wavelet functions, of different nature (Daubechies and Haar) at different scales. This difficulty has been circumvented by considering the technique proposed in [28] and summarized in Appendix A.

After solving the problem, that is, after computing the coefficients affecting the different scaling and wavelet functions appearing in the approximation of functions involved in the solution separated representation, one realizes that each mode $n \leq \mathrm{N}$, $X_n(x) \cdot Y_n(y)$, involves different levels. The Galerkin PGD constructor does not allow associating modes with levels; each mode contains many levels of description, and each level of description is present in most of the PGD modes.

However, in order to extract the multi-scale features of the solution, after having solved the problem, one could proceed with re-ordering the modes by enforcing the first mode to contain the lowest-level contribution (zero level), the second mode the first level contribution, and so on.

To better describe the re-ordering procedure, we consider the solution approximation

$$u^{\mathrm{N}}(x, y) = \sum_{i=1}^{\mathrm{N}} X_i(x) Y_i(y) \tag{56}$$

with

$$\begin{cases} X_i(x) = 2^{j_0/2} \sum_k X^i_{j_0,k} \phi\left(2^{j_0} x - k\right) + \sum_{j=j_0}^{J} 2^{j/2} \sum_k X^i_{j,k} \psi\left(2^j x - k\right) \\ Y_i(y) = 2^{j_0/2} \sum_k Y^i_{j_0,k} \phi\left(2^{j_0} y - k\right) + \sum_{j=j_0}^{J} 2^{j/2} \sum_k Y^i_{j,k} \psi\left(2^j y - k\right) \end{cases} \tag{57}$$

Solution (56) can be rewritten in the form

$$u^{\mathrm{N}}(x, y) = \sum_{i=0}^{J} Z_i(x, y) \tag{58}$$

with

$$Z_0(x, y) = \sum_{i=1}^{\mathrm{N}} \left\{ 2^{j_0/2} \sum_k X^i_{j_0,k} \phi\left(2^{j_0} x - k\right) \cdot 2^{j_0/2} \sum_k Y^i_{j_0,k} \phi\left(2^{j_0} x - k\right) \right\} \tag{59}$$

$$\begin{aligned} Z_1(x, y) = \sum_{i=1}^{\mathrm{N}} \Bigg\{ & 2^{j_0/2} \sum_k X^i_{j_0,k} \phi\left(2^{j_0} x - k\right) \cdot 2^{j_0/2} \sum_k Y^i_{j_0,k} \psi\left(2^{j_0} y - k\right) \\ & + 2^{j_0/2} \sum_k X^i_{j_0,k} \psi\left(2^{j_0} x - k\right) \cdot 2^{j_0/2} \sum_k Y^i_{j_0,k} \phi\left(2^{j_0} y - k\right) \Bigg\} \end{aligned} \tag{60}$$

and so on. Here, each term $Z_i(x, y)$ contains a level of description, from the coarsest one to the finest one.

It is important to note that each term $Z_i(x, y)$ has a separated representation, a finite sum of functional products, with one of the involved functions depending on the x-coordinate, and the other one on the y-coordinate.

4. Numerical examples

4.1. The Poisson problem: convergence analysis

In order to check the proposed strategy, we consider the problem discussed in Section 1, Eq. (8), with $f = 1$ and homogeneous boundary conditions whose exact solution, as previously discussed, is available.

Fig. 3 depicts the reconstructed solution and the associated modes in each direction. It was noticed that the approximation of functions involved in higher modes (e.g., $X_{\mathrm{N}}(x)$ and $Y_{\mathrm{N}}(y)$) required higher levels. The difference between the reconstructed and the exact solution is depicted in Fig. 4.

The convergence analysis considered increasing the number of models involved in the separated representation for different levels in the mode approximation and the inverse, increasing the number of levels for a fixed number of modes. The error is calculated using the L^2-norm. As expected, a strategy increasing either the number of modes or the number of levels rapidly reaches a plateau, as Fig. 5 attests to. Only by increasing simultaneously both of them, the convergence rate can be maintained.

Fig. 6 compares the first three PGD models $X_i(x) \cdot Y_i(y)$ and the re-ordered multi-scale modes, referred to as WPGD modes, $Z_i(x, y)$, according to the strategy described in the previous section.

Finally, Fig. 7 illustrates the contribution of different levels to each PGD mode. It can be noticed that the first PGD modes involve mainly lower levels, whereas higher levels contribute mainly to the last PGD modes.

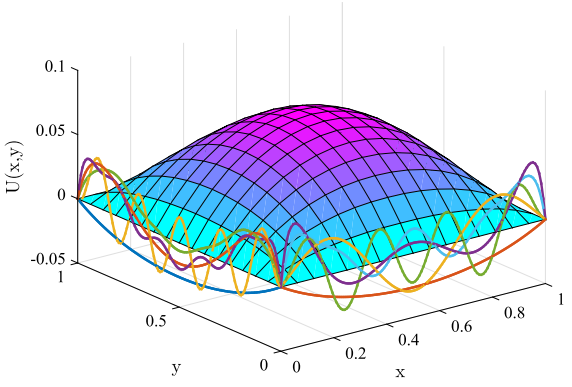

Fig. 3. Reconstructed solution and involved modes in each direction.

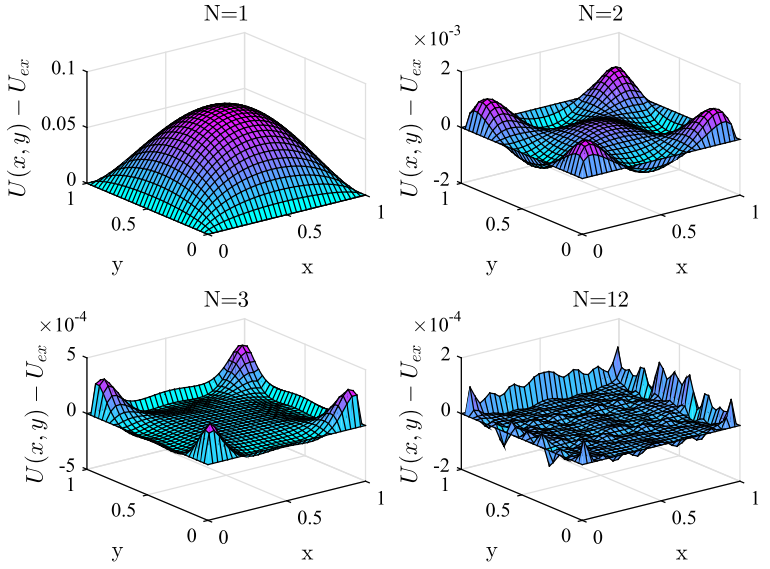

Fig. 4. Difference between the reconstructed solution and the exact one for different number of models considered in the solution separated representation.

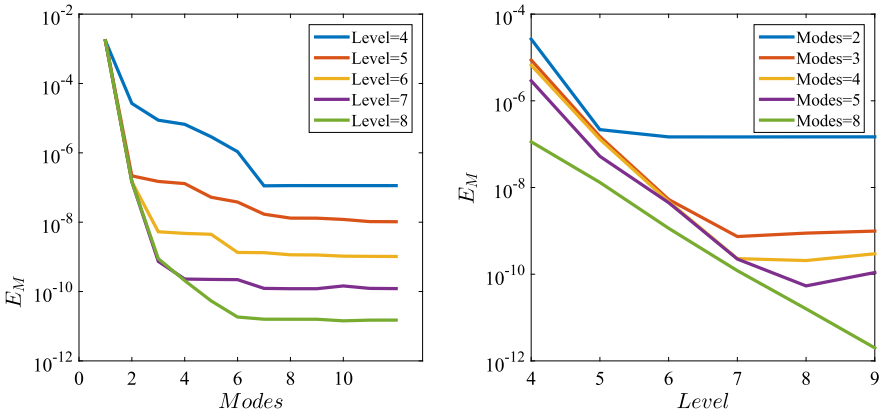

Fig. 5. Convergence analysis with respect to the number of modes and levels.

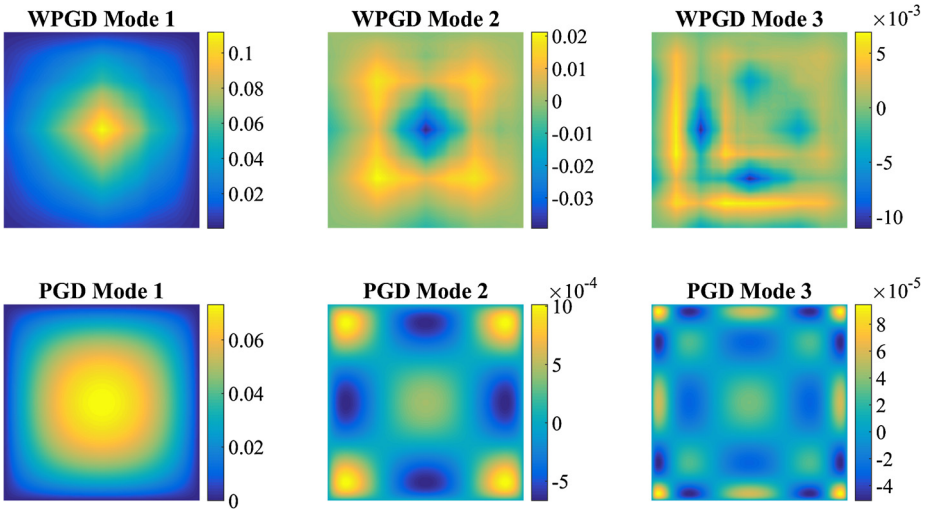

Fig. 6. PGD modes $X_i(x) \cdot Y_i(y)$ versus WPGD modes $Z_i(x, y)$.

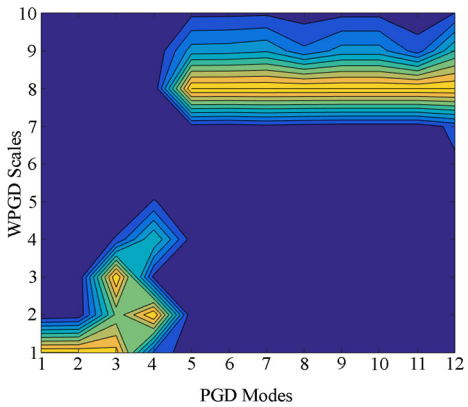

Fig. 7. Contribution of the different levels to each PGD mode.

4.2. Multi-scale analysis

In this case study, we consider a thermal problem (heat transfer equation) in the domain depicted in Fig. 8, consisting of two phases with different thermal conductivities (both of them assumed isotropic). As discussed before, Daubechies wavelets with $\mathcal{N} = 6$ are considered for approximating the functions composing the different solution modes, i.e. the functions $X_i(x)$ and $Y_i(y)$, while the separated representation of the conductivity uses a Haar wavelet representation.

First, the conductivity is approximated at a fine enough level by using a Haar wavelet representation. Then, the solution is approximated at different levels using Daubechies wavelets. When considering fine scales, as fine as the microstructure representation, the solution becomes locally representative of the exact solution, whereas when considering coarser representations with respect to the microstructure length scale, the latter is considered in an averaged sense in the low-scale solution approximations. Fig. 9 represents the solution when considering different representation levels of the solution.

Finally, Figs. 10 and 11 depict respectively the PGD modes $X_i(x) \cdot Y_i(y)$ and their associated multi-scale re-ordered modes $Z_i(x, y)$ described in the previous section, where the multi-scale character is pointed out, with higher frequencies appreciated at higher modes.

4.3. Parametric solutions

Until now, separated representations were associated with the space coordinates x and y. However, as discussed in Section 1, one appealing feature of separated representations concerns the solution to parametric models. In [2], the solution to parametric models within the PGD framework was widely described. Thus, if one is interested in calculating the temperature in the domain Ω for any value of the material conductivity $K \in \mathcal{I} \subset \mathbb{R}^+$, the solution separated representation reads

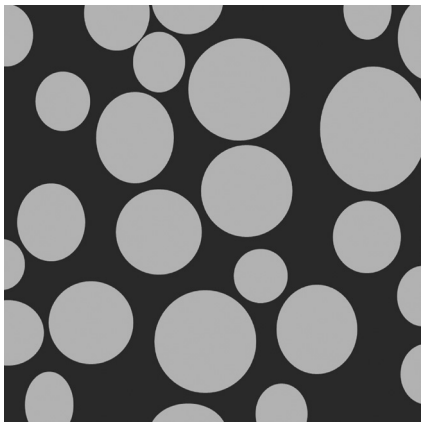

Fig. 8. Domain exhibiting two phases with different thermal conductivities.

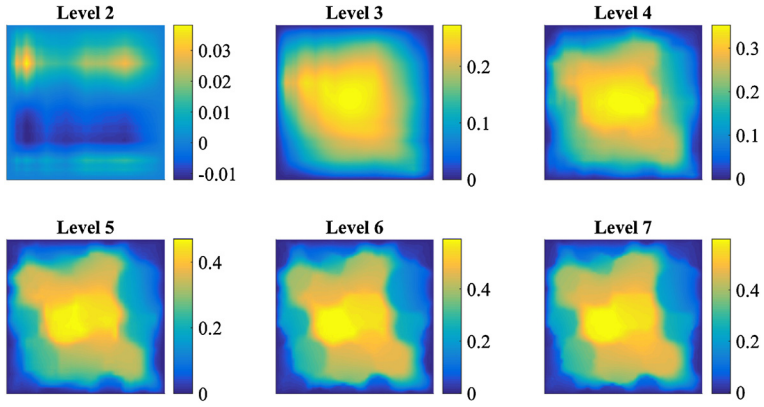

Fig. 9. Reconstructed solution for different approximation levels.

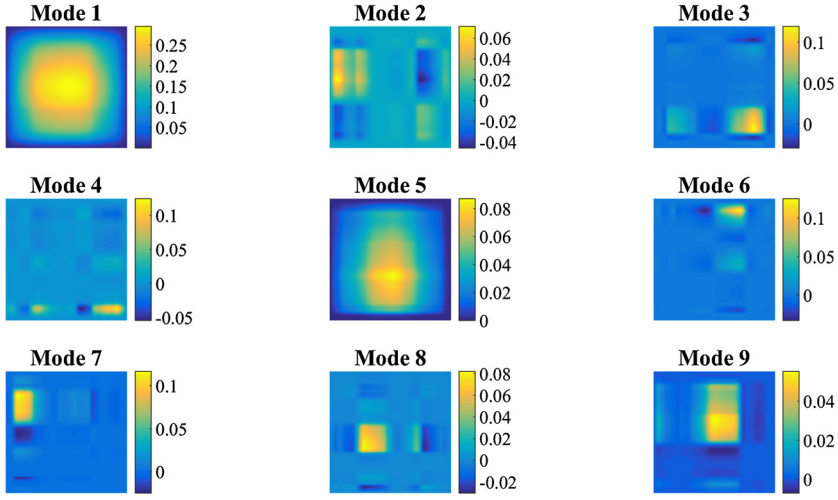

Fig. 10. PGD modes: $X_i(x) \cdot Y_i(y)$.

$$u(\mathbf{x}, K) \approx \sum_{i=1}^{\mathrm{N}} X_i(\mathbf{x})\, C_i(K) \tag{61}$$

which, injected in the problem integral form, allows calculating the functions $X_i(\mathbf{x})$ and $C_i(K)$. However, such a procedure is too intrusive to be considered for calculating a parametric solution when using an external simulation code able to provide the temperature field for a given value of the conductivity, that is, $u(\mathbf{x}; K)$.

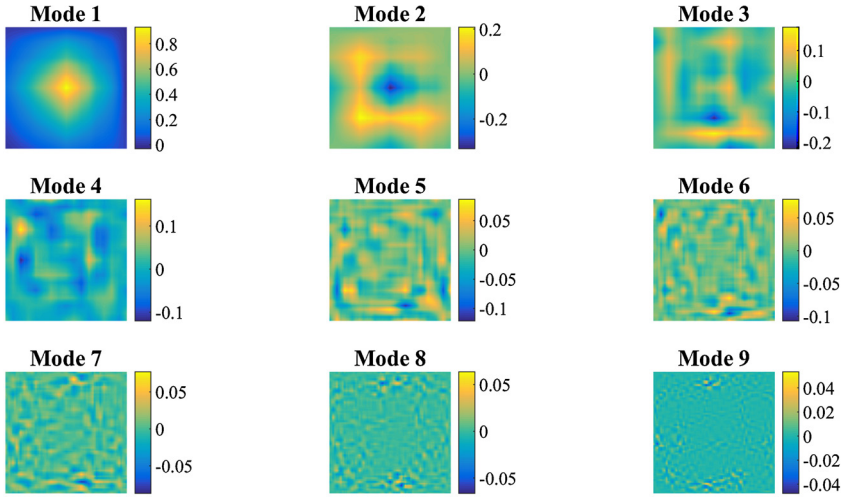

Fig. 11. Multi-scale modes $Z_i(x, y)$ constructed from the wavelet-based PGD separated representation.

The last approach is based on an appropriate sampling of the parametric domain by choosing particular values of the conductivity grouped in the set $\mathcal{K} = \{K_1, K_2, \cdots, K_{\mathrm{K}}\}$ and solving the problem for all these values in order to obtain $u(\mathbf{x}; K_i)$, $\forall i \in \mathcal{K}$. Then, all these solutions can be adequately interpolated for defining the solution for any conductivity value $K \in [K_1, K_{\mathrm{K}}]$. This is the key idea behind surrogate models, response surfaces, etc.

If we imagine for a while that we solved the problem at hand for the minimum and maximum values of the conductivities, leading to $u(\mathbf{x}, K_{\min})$ and $u(\mathbf{x}, K_{\max})$, the conductivity for any other value of the conductivity $K \in [K_{\min}, K_{\max}]$ can be linearly approximated according to

$$u(\mathbf{x}, K) = u(\mathbf{x}, K_{\min}) N_1(K) + u(\mathbf{x}, K_{\max}) N_2(K) \tag{62}$$

where the approximation functions in the parametric space $N_1(K)$ and $N_2(K)$ read

$$N_1(K) = \frac{K - K_{\min}}{K_{\max} - K_{\min}} \tag{63}$$

and

$$N_2(K) = \frac{K_{\max} - K}{K_{\max} - K_{\min}} \tag{64}$$

Finer approximations require finer samplings and the subsequent interpolation. The main difficulty is how choosing these points to keep the accuracy under control. In [29], the authors proposed a non-intrusive sparse subspace learning approach using hierarchical approximation bases.

Another possibility consists in taking advantage of the multi-resolution property of wavelet-based approximations for controlling the sampling richness, that is, the number of elements in the set $\mathcal{K}$. Thus, one can control the convergence by analyzing the value of the associated coefficients related to the wavelet functions. This appealing property is absent in usual polynomial approximations.

Using a wavelet representation of the parametric evolution employing spline-wavelets,

$$\phi(x) = \frac{1}{2}\phi(2x) + \phi(2x - 1) + \frac{1}{2}\phi(2x - 2) \tag{65}$$

has the advantage of recovering coefficients that correspond to the value of the approximated function at its location.

Then, as soon as the solution is computed at two consecutive levels by approximating the parametric functions by using the scaling functions $\phi_{j,k}$ and $\phi_{j+1,k}$, respectively, their difference represents the wavelet contribution that can be expressed in the wavelet basis $\psi_{j,k}$, and whose higher coefficients indicate the locations at which sampling must be refined.

This procedure allowed us to reduce significantly the number of sampling points, of some orders of magnitude in some of our numerical experiments.

5. Conclusions

In this work, we proved that separated representations involved in PGD solution procedures can be combined with wavelet-based functional approximations for extracting the multi-scale behavior present in the solutions.

Moreover, we proved that the multi-resolution property inherent to wavelet representations allows defining simple adaptive procedures with multiple applications: model refinement or efficient sampling for calculating parametric solutions.

The connection between multi-scale solutions in complex microstructures and homogenization constitutes a work in progress.

Appendix A. Calculating integrals involving three functions at the same and at different levels

We consider the integral

$$\Gamma_{l,m}^{d_1,d_2,d_3} \equiv \int_{-\infty}^{\infty} \phi^{d1}(x)\,\phi^{d2}(x-l)\,\phi^{d3}(x-m)\,\mathrm{d}x \tag{66}$$

where $d1, d2$ and $d3$ refers to the derivative order. Integral (66) can be expressed as

$$\Gamma_{l,m}^{d_1,d_2,d_3} = 2^{d_1}2^{d_2}2^{d_3} \int_{-\infty}^{\infty} \sum_k a_k \phi^{d_1}(2x-k) \sum_r a_r \phi^{d_2}(2x-2l-r) \sum_p a_p \phi^{d_3}(2x-2m-p)\,\mathrm{d}x \tag{67}$$

By considering the change of variable $y = 2x - k$, the preceding expression reads

$$\Gamma_{l,m}^{d_1,d_2,d_3} = \frac{2^{d_1}2^{d_2}2^{d_3}}{2} \sum_k a_k \sum_r a_r \sum_p a_p \int_{-\infty}^{\infty} \phi^{d_1}(y)\,\phi^{d_2}(y+k-2l-r)\,\phi^{d_3}(y+k-2m-p)\,\mathrm{d}y \tag{68}$$

which can be rewritten as

$$\Gamma_{l,m}^{d_1,d_2,d_3} = \frac{2^{d_1}2^{d_2}2^{d_3}}{2} \sum_k a_k \sum_r a_r \sum_p a_p \Gamma_{2l+r-k,2m+p-k}^{d_1,d_2,d_3} \tag{69}$$

which defines a rank-deficient eigenproblem. The rank deficiency is circumvented by adding the required number of extra equations [28].

Now we consider that the integral involves functions at different scales,

$$\Omega_{im,l}^{d_1,d_2,d_3} \equiv \int_{-\infty}^{\infty} \phi^{d1}(y)\,\phi^{d2}(y-m)\,\phi^{d3}(2y-l)\,\mathrm{d}y \tag{70}$$

which can be rewritten as

$$\Omega_{m,l}^{d_1,d_2,d_3} = \int_{-\infty}^{\infty} \sum_k a_k 2^{d_1} \phi^{d_1}(2y-k) \sum_r a_r 2^{d_2} \phi^{d_2}(2y-2m-r)\,\phi^{d_3}(2y-l) \tag{71}$$

or

$$\begin{aligned}\Omega_{m,l}^{d_1,d_2,d_3} &= 2^{d_1}2^{d_2} \sum_k a_k \sum_r a_r \int_{-\infty}^{\infty} \phi^{d_1}(z)\,\phi^{d_2}(z+k-2m-r)\,\phi^{d_3}(z+k-l) \\ &= 2^{d_1}2^{d_2} \sum_k a_k \sum_r a_r \Gamma_{2m+r-k,l-k}\end{aligned} \tag{72}$$

which results again in an eigenproblem.

References

[1] F. Chinesta, P. Ladevèze, E. Cueto, A short review in model order reduction based on Proper Generalized Decomposition, Arch. Comput. Methods Eng. 18 (2011) 395–404.

[2] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, A. Huerta, Parametric PGD based computational vademecum for efficient design, optimization and control, Arch. Comput. Methods Eng. 20 (1) (2013) 31–59.

[3] F. Chinesta, P. Ladevèze (Eds.), Separated Representations and PGD Based Model Reduction: Fundamentals and Applications, CISM–Springer, 2014.

[4] F. Chinesta, A. Huerta, G. Rozza, K. Willcox, Model order reduction, in: Encyclopedia of Computational Mechanics, 2nd edition, Wiley, 2016.

[5] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, J. Comput. Phys. 202 (2005) 346–366.

[6] S. Volkwein, Model Reduction Using Proper Orthogonal Decomposition, Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz, Austria, 2011.

[7] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (4) (2015) 483–531.
[8] A.T. Patera, G. Rozza, Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations, MIT Pappalardo Monographs in Mechanical Engineering, 2007, online at http://augustine.mit.edu.
[9] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics, Arch. Comput. Methods Eng. 15 (3) (2008) 229–275.
[10] P. Ladevèze, The large time increment method for the analyze of structures with nonlinear constitutive relation described by internal variables, C. R. Acad. Sci. Paris, Ser. IIb 309 (1989) 1095–1099.
[11] P. Ladevèze, Nonlinear Computational Structural Mechanics. New Approaches and Non-incremental Methods of Calculation, Springer-Verlag, 1999.
[12] 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, J. Non-Newton. Fluid Mech. 139 (2006) 153–176.
[13] 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 representation, J. Non-Newton. Fluid Mech. 144 (2007) 98–121.
[14] A. Nouy, Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms, Comput. Methods Appl. Mech. Eng. 197 (2008) 4718–4736.
[15] B. Bognet, A. Leygue, F. Chinesta, A. Poitou, F. Bordeu, Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity, Comput. Methods Appl. Mech. Eng. 201 (2012) 1–12.
[16] 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.
[17] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations. A Primer, Springerbriefs, Springer, 2014.
[18] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Comput. Methods Appl. Mech. Eng. 199 (2010) 1872–1880.
[19] P. Ladevèze, L. Chamoin, On the verification of model reduction methods based on the proper generalized decomposition, Comput. Methods Appl. Mech. Eng. 200 (2011) 2032–2047.
[20] E. Nadal, A. Leygue, F. Chinesta, M. Beringhier, J.J. Rodenas, F.J. Fuenmayor, A separated representation of an error indicator for the mesh refinement process under the Proper Generalized Decomposition framework, Comput. Mech. 55 (2) (2015) 251–266.
[21] A. Falcó, A. Nouy, Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces, Numer. Math. 121 (3) (2012) 503–530.
[22] S. Gopalakrishnan, M. Mitra, Wavelet Methods for Dynamical Problems with Application to Metallic, Composite, and Nano-Composite Structures, CRC Press, Taylor & Francis, 2010.
[23] I. Daubechies, Orthonormal basis of compactly supported wavelets, Commun. Pure Appl. Math. 41 (7) (1988) 909–996.
[24] A. Avudainayagam, C. Vani, Wavelet-Galerkin method for integro-differential equations, Appl. Numer. Math. 32 (3) (2000) 247–254.
[25] S. Jones, M. Legrand, The wavelet-Galerkin method for solving PDE's with spatially dependent variables, in: 19th International Congress on Sound and Vibration, ICSV19, Vilnius, Lithuania, 2012, p. R33, Numer. Methods Acoust. Vib. 326 (2012).
[26] B.V.R. Kumar, M. Mehra, A three-step wavelet Galerkin method for parabolic and hyperbolic partial differential equations, Int. J. Comput. Math. 83 (1) (2006) 143–157.
[27] Y. Mahmoudi, Wavelet Galerkin method for numerical solution of nonlinear integral equations, Appl. Math. Comput. 167 (2) (2005) 1119–1129.
[28] A. Latto, H.L. Resnikoff, E. Tannenbaum, The evaluation of connection coefficients of compactly supported wavelets, in: Proceedings of the French–USA Workshop on Wavelets and Turbulence, Springer, 1991.
[29] D. Borzacchiello, J.V. Aguado, F. Chinesta, Non-intrusive sparse subspace learning for parametrized problems, Arch. Comput. Methods Eng. (2017), https://doi.org/10.1007/s11831-017-9241-4.