

A local multiple Proper Generalized Decomposition based on the Partition of Unity

Rubén Ibáñez^{1,3}, Emmanuelle Abisset-Chavanne¹, Francisco Chinesta²,
Antonio Huerta³, and Elías Cueto⁴

¹ICI - High Performance Computing Institute, Ecole Centrale de Nantes, 1 rue de la Noe, F-44300 Nantes, France
²ESI Group Chair @ PIMM - Procédés et Ingénierie en Mécanique et Matériaux, ENSAM ParisTech, 151 boulevard de l'Hôpital, Paris, France
³Laboratori de Calcul Numeric, Universitat Politècnica de Catalunya, BarcelonaTech, Calle Jordi Girona 1, 08034 Barcelona, Spain
⁴Aragon Institute of Engineering Research, Universidad de Zaragoza, Edificio Betancourt. Maria de Luna, s.n., 50018 Zaragoza, Spain

SUMMARY

It is well known that model order reduction techniques that project the solution of the problem at hand onto a low-dimensional subspace present difficulties when this solution lies on a non-linear manifold. To overcome these difficulties—notably, an undesirable augment in the number of required *modes* in the solution—several solutions have been suggested. Among them we can cite the use of non-linear dimensionality reduction techniques or, alternatively, the employ of local linear reduced order approaches. These last approaches usually present the difficulty of ensuring continuity between these local models. Here, a new method is presented that ensures this continuity by resorting to the paradigm of the partition of unity, while employing Proper Generalized Decompositions at each local patch.

Received . . .

KEY WORDS: Model order reduction; Proper Generalized Decomposition, Local Reduced Order Modeling, Partition of Unity

1. INTRODUCTION

The finite element method is the ubiquitous technique for the approximation of partial differential equations (PDE). Its generality has led it to succeed in many areas of engineering interest. However, when real time or many-query applications are envisaged, it is well known to be a technique somewhat slow. In recent years, model order reduction (MOR) techniques have shown that a minimum number of carefully-chosen degrees of freedom may be enough for an accurate solution of these same PDE. Instead of choosing general-purpose piecewise polynomials as basis functions, MOR techniques construct *ad hoc* basis functions following different techniques. For instance, Proper Orthogonal Decomposition (POD) [1–5] constructs an efficient basis from a set of precomputed snapshots of the full-order PDE solution,

$$u(x, t) = \sum_{i=1}^N \alpha_i(t) \phi_i(x), \quad (1)$$

where, very much like in the finite element context, α_i are a set of time-dependent coefficients that evolve in time, and $\phi_i(x)$ are time-independent basis functions obtained by some algebraic treatment of the system snapshots. For instance, many MOR techniques employ the most energetic

eigenfunctions of the snapshot autocorrelation matrix to construct these $\phi_i(x)$. These play a similar role to the finite element shape functions, albeit they are global instead of local. Other techniques, such as Reduced Basis methods, for instance [6–8], employ some snapshots of the full-order solution as basis for the approximate solution of the system. These snapshots are calculated in a greedy fashion, at time (or parameter) instants at which the error in the approximation is maximal.

It is important to note that Eq. (1) constitutes in fact a separated expression of the solution—note the space-time separation, which also holds in finite element approximations—. Assuming this separate or affine decomposition of the solution is also on the origin of another family of MOR techniques coined as Proper Generalized Decomposition (PGD) [9–12]. These methods compute the basis function *on the fly*, that is to say, by means of a greedy algorithm that enriches successively the basis until the desired precision is achieved. This methodology has proven to be very effective from a wide variety of high dimensional problems ranging from the resolution of Fokker-Planck equation [13], to patient-specific liver responses [14, 15], structural dynamics [16], computational rheology [17] or, more generally, to any parametric problem that could be written in separate form [13].

There are situations, however, when the solution is highly non-separable. In other words, the solution of the problem lives on a non-linear manifold. In this situation, what MOR techniques do is to project the solution on the tangent space to the manifold at a given (time or parameter) point [18]. This leads to poor approximation properties far from the tangency point, unless special techniques are chosen, as in [19], for instance, where asymptotic expansions were used.

A way to alleviate this problem is to follow the same philosophy than non-linear-dimensionality reduction methods. For instance, Locally-Linear-Embedding (LLE) [20] tries to unveil the latent variables by means of imposing a local linear variation on the function, which will change from neighborhood to neighborhood. Another technique which deals with non-linear problems is the so-called kernel Principal Component Analysis (k-PCA) [21]. In this particular case, the definition of an efficient kernel function allows to project the snapshots to high dimensional spaces (potentially, infinite dimensional) in which the solution manifold is flat. In this particular situation, standard interpolation techniques work well.

A. Badías et al. [22] studied the case of a moving source in a transient heat transfer problem. The problem of the non-separability of the solution was circumvented by means of making a partition of the time domain, dedicating a different PGD for each partition. However, the imposition of the interface conditions will become a tedious task when dealing with partitions involving variables other than time.

The main objective of this paper is to develop a generalized PGD formulation in which continuity between subdomains is guaranteed. This will be achieved by resorting to the Partition of Unity (PU) paradigm [23, 24]. By employing the partition of unity, continuity of the solution is guaranteed if the chosen PU is continuous. This will allow us to glue different PGD approximations defined at particular regions of the space, time or parameter spaces, thus ensuring a global solution with a minimum of degrees of freedom.

The structure of the paper is organized as follows: the second section depicts the general aspects of the proposed methodology, the third section shows the capability of the method to approximate highly non-linear functions, the fourth section shows applications for different PDEs from fully diffusive to transient equations. The fifth section will be devoted to the conclusions. All details related to the variational form will be given in the appendix.

2. BASICS OF THE METHOD

The method here proposed is based, as mentioned earlier, in the application of the Partition of Unity (PU) paradigm [23, 24]. In essence, the PU method states that, given a collection of non-overlapping patches defined over the domain Ω , Ω_i , $i = 1, \dots, n_{\text{patch}}$, a partition of unity φ_i defined on these patches (i.e., $\sum_i \varphi_i = 1$ everywhere in the domain), and function spaces V_i , associated to

each patch, the approximation obtained by

$$V = \sum_{i=1}^{n_{\text{patch}}} \varphi_i V_i$$

inherits the approximation properties of the spaces V_i and the continuity properties of φ_i .

For the sake of simplicity but without losing generality, the local PGD method here presented is illustrated by starting with a weak form related to a general two-dimensional PDE

$$\mathcal{L}(u^*(x, y), u(x, y); \mu) = \mathcal{F}(u^*(x, y), f(x, y)) \text{ in } \Omega.$$

Normally, solving the weak form (2) requires an approximation space for the essential variable $u(x, y)$ and its associated variation $u^*(x, y)$. The classical finite element approximation,

$$u(x, y) = \sum_{i \in \mathcal{I}} N_i(x, y) u_i$$

where \mathcal{I} is the entire set of shape functions defined over the integration domain, Ω , satisfies the partition of unity and linear consistency properties,

$$\begin{aligned} \sum_{i \in \mathcal{I}} N_i(x, y) &= 1, & \forall x, y \in \Omega, \\ \sum_{i \in \mathcal{I}} N_i(x, y) x_i &= x, & \forall x, y \in \Omega, \\ \sum_{i \in \mathcal{I}} N_i(x, y) y_i &= y, & \forall x, y \in \Omega. \end{aligned}$$

The main ingredient of the method here proposed is the combination of the finite element shape functions, $N_i(x, y)$, as an example of very convenient partition of unity, enriched with low rank PGD approximations. Therefore, each degree of freedom u_i associated to the FEM shape function $N_i(x, y)$ will be enriched with a PGD approximation, aiming to capture the details of the solution which are not captured by the standard low-order FE meshes. Therefore, the approximation of the solution will read

$$u(x, y) = \sum_{i \in \mathcal{I}} N_i(x, y) \sum_{k=1}^M X_k^i(x) Y_k^i(y),$$

where $X_k^i(x)$ and $Y_k^i(y)$ functions are the k -th one-dimensional modes related to the i -th PGD.

Several approaches can be adopted to define the trial function. In this particular case, a Bubnov-Galerkin projection is selected. Hence, the same approximating space is chosen for $u(x, y)$ and $u^*(x, y)$,

$$u^*(x, y) = \sum_{i \in \mathcal{I}} N_i(x, y) (X_M^{i*}(x) Y_M^i(y) + X_M^i(x) Y_M^{i*}(y)). \quad (2)$$

It is worth noting the greedy nature of the PGD algorithm, since the variation only takes into account the last, M -th, PGD mode. Previous modes are considered as known, and therefore do not appear in Eq. (2). Furthermore, a non-linear problem, which has been created due to the separation of variables—note that we solve for a pair of functions, X_M, Y_M —, is solved using an alternate direction scheme. Conceptually speaking, it is not necessary to have the same number of modes (M) for each macro shape function in \mathcal{I} . A possible strategy would be to stop enriching with new modes in areas where the residual of the equation is low enough. In this first work, all PGDs are enriched with the same number of modes (M) as the main focus is placed on the methodology itself and proving that combining different PGDs within a continuous framework could be useful to obtain separated solutions in complex scenarios where standard PGD performance is decimated. That being said, the definition of mode adopted herein is the number of separated functions (1 per element in the set \mathcal{I}) that has to be given to the macro partition to reproduce the solution, or in other words, the

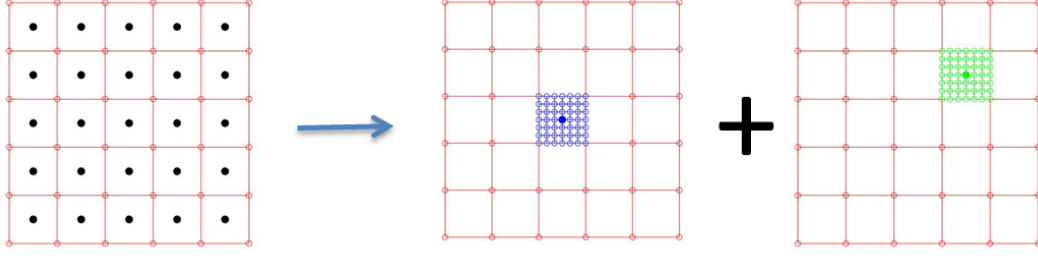


Figure 1. Left, piecewise constant FE approximation. Center, domain of influence of i -th PGD. Right, domain of influence of k -th PGD. As can be noticed, no overlapping exists between different PGD approximations.

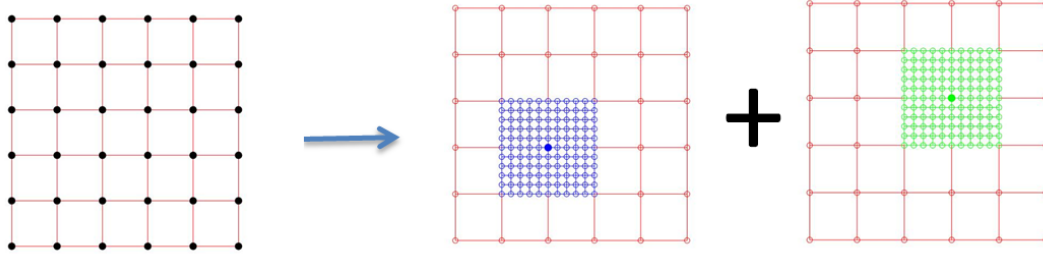


Figure 2. Left, piecewise linear partition of unity. Center, domain of influence of i -th PGD. Right, domain of influence of k -th PGD.

number of modes equal to M . It is also important to reckon that the size of the mode will increase according to the number of elements in \mathcal{I} , since the mode involves as many separated functions per direction as number of degrees of freedom in \mathcal{I} .

The properties of the method will change depending on the way finite element shape functions are defined. Fig. (1) exemplifies the partition when piecewise constant shape functions are used. As it can be seen, no overlapping between different PGDs exists. Therefore, extra effort has to be done to set proper interface conditions along the red lines appearing in the same figure. This situation would be equivalent to the method described in [22]. This is in sharp contrast with the purely additive approach to couple FE and PGD proposed in [25] or in [26], for instance. In that case, the method resembled the s -version of the FEM [27] or the multiscale FEM proposed by Rank and coworkers [28]. In that case, a single PGD approximation was enriched by FE in order to capture high-gradient details of the solution, that usually provoke an increase in the number of PGD modes. Here, on the contrary, we develop an approximation by means of multiple PGD approximations, glued together by means of the Partition of Unity paradigm. We have coined this method the *PU-PGD*.

Fig. (2) depicts a piecewise linear partition of the domain. As it can be noticed, a quadrilateral element has contributions coming from four different PGDs. The coupling conditions between different PGDs are automatically taken into account in the elemental contributions. Indeed, this kind of partition imposes a smooth transition between PGDs, which vary continuously along the domain in accordance with the partition of the domain.

It is worth mentioning that the full potential of PGD approximation resides in the capability of writing the integral form in a separated manner. By doing that, all integrals in a high dimensional space will be split into a product of integrals related to each one of the subspaces. For the sake of simplicity, we will assume that the shape functions $N_i(x, y)$ are exactly separable in one single mode

$$N_i(x, y) = N_i^x(x)N_i^y(y).$$

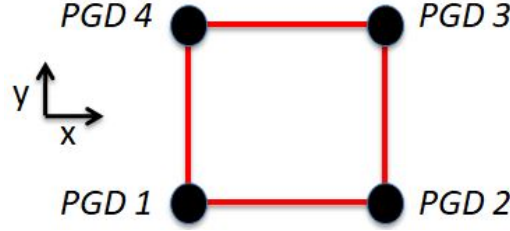


Figure 3. In the case of bi-linear finite elements, there is one PGD enrichment per node.

It is important to notice that this will be the case, for instance, of 2D quadrilateral elements with straight and parallel sides.

Even though this assumption may be seen as a limitation, it is important to notice that there are some variables such as time or parameters, that naturally admit cartesian decompositions. Therefore, when dealing with a complex spatial geometry that evolves in time, a smart partition would be to keep the space variable as is. Nevertheless, this very first work is meant to provide an insight of the method, thus, all results presented in the sequel will be related to piecewise linear FE shape functions based on squared elements.

3. PU-PGD FOR APPROXIMATION PROBLEMS

3.1. Basics of the method

In this section, we analyze the capability of the proposed methodology in approximation problems of the form

$$\int_{\Omega} u^*(x, y) (u(x, y) - f(x, y)) d\Omega = 0, \quad (3)$$

where a function $f(x, y)$ is to be approximated in an PU -PGD framework. As it can be noticed, there are no partial derivatives involved in this type of problems. Instead just a compact expression of the function $f(x, y)$ is sought. It is important to highlight that continuity is not a requirement as no differential equation is involved this particular case. Indeed, considering a broken approximation space with a much more local procedure will provide better results if the function to be captured presents high gradients or even discontinuities. However, this very first example is included as an initial point to illustrate the continuous approach which is mandatory to solve PDEs

Following the spirit of standard finite element approximation, the integral over the entire domain Ω is split into a sum of integrals for each one of the finite elements Ω_e appearing in the domain. Recall that piecewise bi-linear shape functions present contributions from four different PGDs per element (one for each corner), as shown in Fig. (3).

It is important to notice that the i -th PGD, thanks to the partition of unity, affects the support of each finite element node, and thus four elements in a regular lattice.

Omitting the dependence with respect to space variables, x and y , and thanks to the separability of the FE shape function (if defined in a regular lattice as a tensorial product of one-dimensional functions), the first term in the integral form of Eq. (3) particularized for a element Ω_e reads

$$\int_{\Omega_e} u^* u dx dy = \sum_{i=1}^4 \sum_{j=1}^4 \int_{\Omega_e} N_i^x N_i^y (X_M^i Y_M^i)^* \left(N_j^x N_j^y \sum_{k=1}^M X_k^j Y_k^j \right) dx dy.$$

The next step is to split the 2D integral form into a tensorial product of 1D integrals as

$$\begin{aligned} & \int_{\Omega_e} u^* u dx dy \\ &= \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} N_i^x N_j^x X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy + \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^i N_i^x N_j^x X_k^j dx \int_y Y_M^{i*} N_i^y N_j^y Y_k^j dy. \end{aligned} \quad (4)$$

As can be noticed, the variation along the x direction shares all the operators with the variation along the y direction. Indeed, this property arises naturally from the alternate directions scheme that is used to solve the non-linear system of the PGD.

Indeed, the efficiency of the PU -PGD algorithm is maximized when all operators related to Eq. (4) are precomputed in an off-line phase as

$$\begin{aligned} \alpha_x^{ijk} &= \int_x X_M^i N_i^x N_j^x X_k^j dx, \\ \alpha_y^{ijk} &= \int_y Y_M^i N_i^y N_j^y Y_k^j dy. \end{aligned}$$

With the source term $f(x, y)$ expressed in a separated format (e.g., by invoking the SVD) as

$$f(x, y) = \sum_{z=1}^Z f_z^x(x) f_z^y(y),$$

the right hand side term appearing in Eq. (3) reads

$$\int_{\Omega_e} u^* f dx dy = \sum_{i=1}^4 \sum_{z=1}^Z \int_x X_M^{i*} N_i^x f_z^x dx \int_y Y_M^i N_i^y f_z^y dy + \sum_{i=1}^4 \sum_{z=1}^Z \int_x X_M^i N_i^x f_z^x dx \int_y Y_M^{i*} N_i^y f_z^y dy,$$

for x and y systems respectively.

Once more, all operators related to the integration of the source term can be precomputed off-line as

$$\begin{aligned} \varepsilon_x^{iz} &= \int_x X_M^i N_i^x f_z^x dx, \\ \varepsilon_y^{iz} &= \int_y Y_M^i N_i^y f_z^y dy. \end{aligned}$$

3.2. A preliminar example

Let us illustrate the methodology by analyzing a highly non separable function,

$$f_1(x, y) = \frac{10}{\sigma\sqrt{2\pi}} e^{-\frac{(x-(vy+x_0))^2}{2\sigma^2}},$$

where $\sigma = 0.05$, $v = 0.5$ and $x_0 = 0.2$.

Fig. (4) depicts the resulting $u(x, y) = f_1(x, y)$ scalar field from different perspectives. As it can be seen, the source term is going through the diagonal of the domain, generating a highly non-separable function. Indeed, a standard PGD algorithm encounters many problems to capture such kind of solution.

Two different partitions of the domain has been tested, as shown in Fig. (5). Along the right and left sides of the domain no enrichment is added to the FE approximation as the seeked solution is already zero along this boundary. It is important to reckon that right and left sides of the domain could have been enriched, since no boundary conditions need to be imposed in this approximation

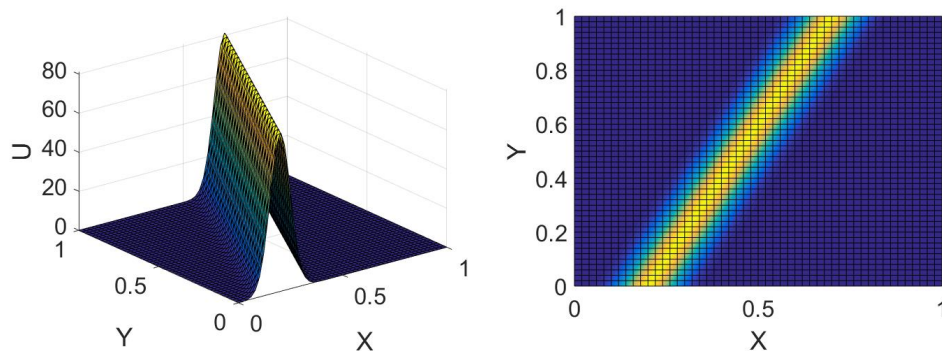


Figure 4. Left, 3D view of $f_1(x, y)$. Right, top view of $f_1(x, y)$.

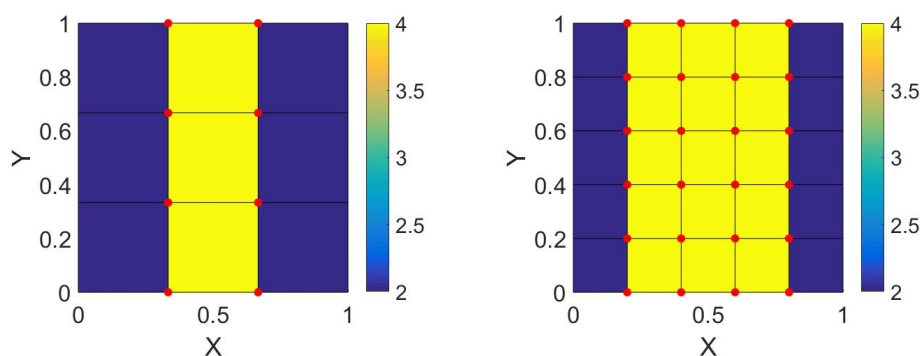


Figure 5. Left, domain enriched with 8 PGDs. Right, domain enriched with 24 PGDs. Red points, centroid of each PGD. The legend shows the number of PGD enrichments affecting each element.

problem. However, the fact that compatible boundary conditions, (i.e. zero along right and left sides) can be imposed was used as a validation point for the next numerical examples involving PDEs. The elements are colored in accordance with the number of active PGDs acting in each element.

Fig. (6) shows the reconstructed solution using 4 modes per local PGD, when the domain has 8 active PGDs (left) and 24 PGDs (right). As it can be seen, even though both approximations capture the main features of the solution, the solution using 24 PGDs is slightly better than the one with 8 PGDs.

Fig. (7) shows the convergence error for the approximation of $f_1(x, y)$. The relative error is measured as

$$\mathcal{E}^m = \frac{\|u_{ref} - u^m\|}{\|u_{ref}\|},$$

where u^m is the reconstructed solution using m modes. u_{ref} represents the ground truth, that is calculated in this case with the help of a dense enough finite element mesh. As can be noticed, singular value decomposition (SVD) method suffers from separating this kind of solutions, having a slow decay of the relative error with respect to the number of modes. As it is well-known from the literature, standard PGD in two dimensions is exactly equivalent to SVD when an identity operator is employed (the so-called PGD in approximation [29, 30]). On the other hand, using the *PU*-PGD algorithm the relative error decays much faster than the one related to standard SVD or PGD approximations.

It is important to notice that the more a domain is partitioned the smaller the relative error is. This behavior is expected since the smaller the elements are, the easier is to capture a local behavior. However, the price to pay is that the cost of computing one mode per PGD will increase just like

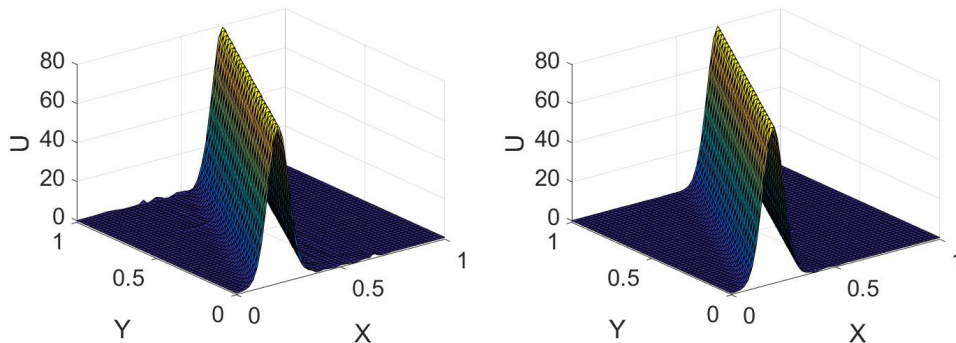


Figure 6. Left, solution with 8 PGDs. Right, solution with 24 PGDs. Every PGD enrichment incorporates four modes.

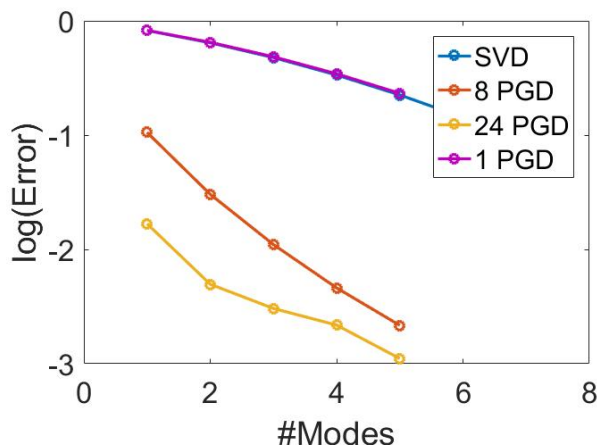


Figure 7. Convergence error for the approximation problem. Comparison between SVD, standard PGD, and two different local strategies.

the storage cost as well. Certainly, the systems that are solved in the fixed point iteration are not local due to the fact that macro shape functions are overlapped between them. If the macro partition involves $\#\mathcal{I}$ shape functions and each local PGD is discretized using n_x and n_y degrees of freedom, the number of unknowns in each PGD fixed point are $\#\mathcal{I} \cdot n_x$ and $\#\mathcal{I} \cdot n_y$ for x and y directions, respectively. Therefore, it is important to keep in mind that refining the macro mesh would reduce the number of modes (M) to represent the solution, but it would also increase the computational cost for obtaining one mode. However, the main focus of this work was to prove that the proposed methodology was able to obtain separated solutions in scenarios where standard PGD performance was not good enough.

4. PU-PGD FOR THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

The aim of this section is to show the potential of the proposed methodology when applied to the solution of different PDEs. We consider examples of increasing complexity. Therefore, the first test case is a pure diffusion equation. The second case is a convection-reaction-diffusion equation. Finally, the last example relates to a transient thermal problem with a moving source term. Even

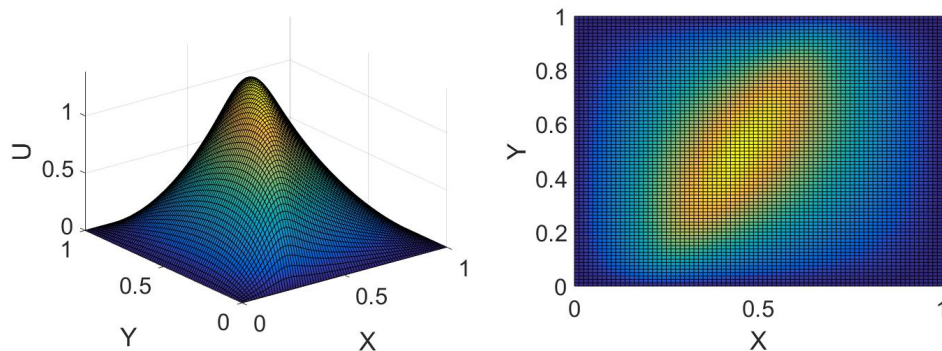


Figure 8. Left, 3D view of $u(x, y)$. Right, planar view of $u(x, y)$ for the diffusion equation.

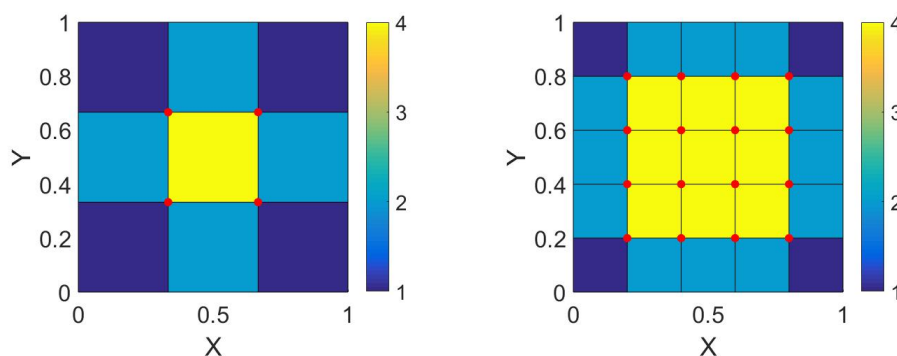


Figure 9. Left, problem solved with four PGDs. Right, problem solved with 16 PGDs. Red points represent finite element nodes enriched with PGD. The legend represents the number of PGDs per element.

though the construction of the operators for each PDE follows the same trend than the ones presented in the *PU*-PGD in approximation, further details are given in the appendix.

4.1. Diffusion PDE

We first consider the following weak form corresponding to a diffusion problem:

$$\int_{\Omega} \eta \nabla u^* \cdot \nabla u dx dy = \int_{\Omega} u^* f_1 dx dy.$$

Vanishing Dirichlet boundary conditions are imposed on the entire boundary. The source term is equal to the function $f_1(x, y)$ defined in the previous section and the diffusion parameter, η , is set to 1.

Fig. (8) depicts the reference $u(x, y)$ scalar field for the diffusion equation from different perspectives. It has been obtained with a 80×80 finite element discretization. As it can be seen, the solution diffuses the source term through the diagonal of the domain.

Two different PGD enrichments has been tested as shown in Fig. (9). The PGDs acting on top, bottom, left and right sides are set to zero to be consistent with the problem statement where zero Dirichlet boundary conditions are imposed along the entire boundary domain. The red points indicate the enriched nodal supports. The elements are colored in accordance with the number of enriching PGDs. Elements on the corners have only one active PGD, whereas elements on the sides have two active PGDs and interior elements have four active PGDs.

The reconstructed solution using 4 modes per PGD, when the domain has 4 active PGDs (left) and 16 PGDs (right) are in perfect agreement with the reference solution appearing in Fig.9.

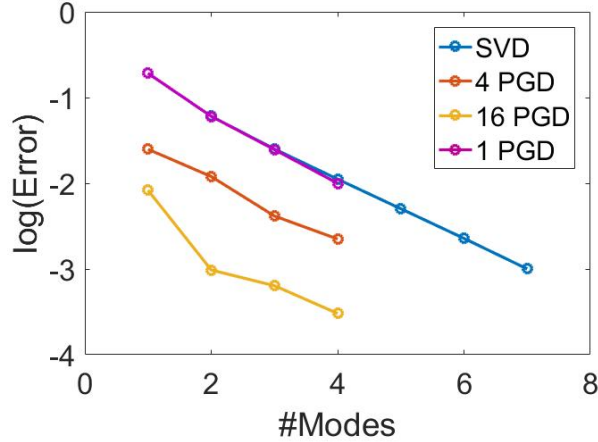


Figure 10. Convergence error for the diffusion problem. Comparison between the number of modes provided by SVD, standard PGD, and two different enrichment strategies considered so far.

Fig. (10) shows the convergence error for the diffusive equation. As it can be noticed, all methods converge monotonically, being the SVD the one that presents the slower convergence with respect the number of modes. Standard PGD shows a similar convergence rate to SVD, as expected. Even if no formal proof exists of this behavior for operators other than the identity, this is the observed rate for all the experiments did so far. Finally, using the multi PGD algorithm presents a faster decay in the relative error.

4.2. Convection-Reaction-Diffusion PDE

In this subsection a convection-reaction-diffusion equation is studied. Indeed, the major novelty is the introduction of the convective term since it will involve a non-symmetric operator. The weak form related to the this equation is

$$\int_{\Omega} u^* \mathbf{v} \cdot \nabla u + \eta \nabla u^* \cdot \nabla u + \sigma u^* u \, dx dy = \int_{\Omega} u^* f_2 \, dx dy, \quad (5)$$

where vanishing Dirichlet boundary conditions are considered on the entire boundary. The integration domain is $\Omega = [0, 1] \times [0, 1]$. The convective velocity field is assumed to be $\mathbf{v} = 500(y - 0.5, 0.5 - x)^T$. The reaction term is set to $\sigma = 10$ and the diffusion coefficient to $\eta = 1$. Regarding the source term, a Gaussian located in $[x_0, y_0] = [0.75, 0.75]$ as

$$f_2(x, y) = \frac{800}{\sqrt{2\pi}} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{0.005}},$$

is considered.

Fig. (11) depicts the reference $u(x, y)$ scalar field for Eq. (5)—computed by employing an 80×80 finite element mesh, thus with no need for stabilization—from different perspectives. Note how the Gaussian source term is convected circularly according to the prescribed velocity field. Moreover, the diffusion term makes the Gaussian to be smoothed along its convection path.

The same partitions than in the diffusive case have been tested as shown in Fig. (9), since also null Dirichlet boundary conditions are imposed along the entire boundary of the domain. Fig. (12) shows the reconstructed solution using 4 modes per local PGD, when the problem has been approximated by four PGDs (left) and sixteen PGDs (right). Notice how the approximation related to four PGDs does not capture the solution well. Indeed, a higher number of modes will be needed to converge to the reference solution. On the other hand, the solution involving 16 PGDs is in perfect accordance with the reference solution, capturing all the features of the scalar field without any perceptible oscillation.

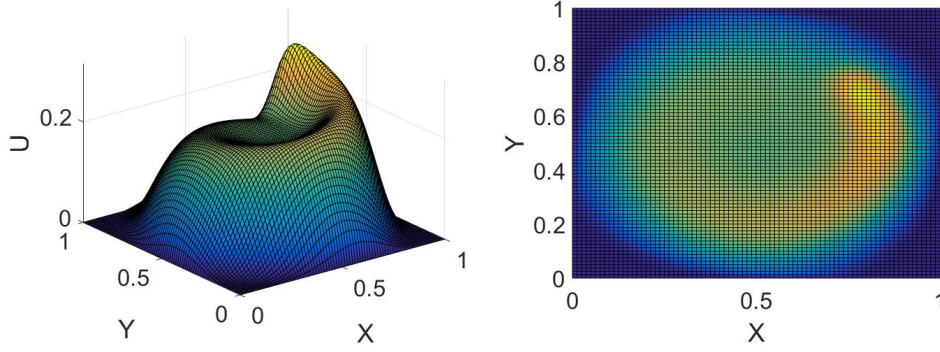


Figure 11. Left, 3D view of the reference solution $u(x,y)$. Right, planar view of $u(x,y)$ for convection-reaction-diffusion equation.

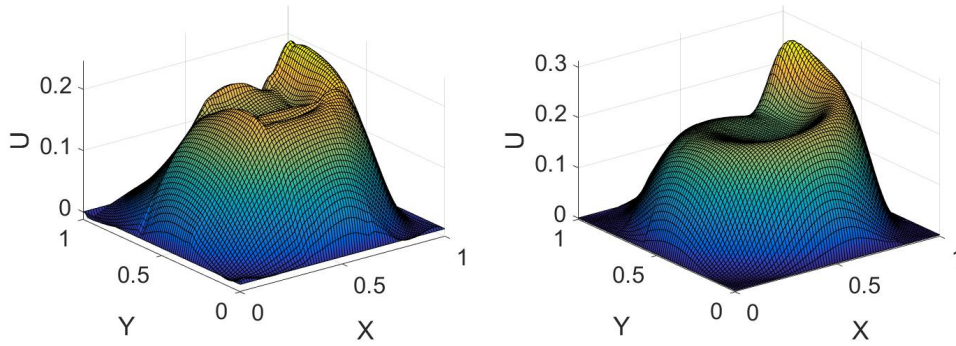


Figure 12. Solution of the convection-diffusion-reaction equation. Left, 4 PGDs. Right, 16 PGDs. Both solutions have four modes per PGD.

Fig. (13) shows the convergence of the relative error in logarithmic scale for the SVD and the two different partitions proposed for the CDR equation. It can be clearly seen that the convergence related to the 4 PGD partition is the slowest one. Indeed, this problematic is derived from the fact that the first PGD modes have to capture a highly non-linear behavior, meaning that this partition will require a high amount of modes to converge to the real solution. On the other hand, the 16 PGD partition convergence is quite fast compared with the other two methods.

4.3. Transient Heat Transfer Equation

This last example is devoted to the analysis of a transient 1D heat transfer problem, a test proposed by S. Idelsohn [31] to verify the compactness of different model order reduction techniques:

$$\int_{\Omega} u^* \frac{\partial u}{\partial t} + \eta \frac{\partial u^*}{\partial x} \frac{\partial u}{\partial x} dx dt = \int_{\Omega} u^* f_3 dx dt,$$

where the diffusion coefficient is set to $\eta = 0.01$.

The domain of study is $\Omega = \Omega_x \times \Omega_t = [0, 1] \times [0, 1]$. The set of boundary conditions imposed for the field $u(x, t)$ are $u(x, 0) = 0$, $u(0, t) = 0$ and $u(1, t) = 0$. The source term $f_3(x, y)$ is

$$f_3(x, t) = \frac{10}{\sigma\sqrt{2\pi}} e^{-\frac{(x-(vt+x_0))^2}{2\sigma^2}},$$

where $\sigma = 0.05$, $v = 0.5$ and $x_0 = 0.1$.

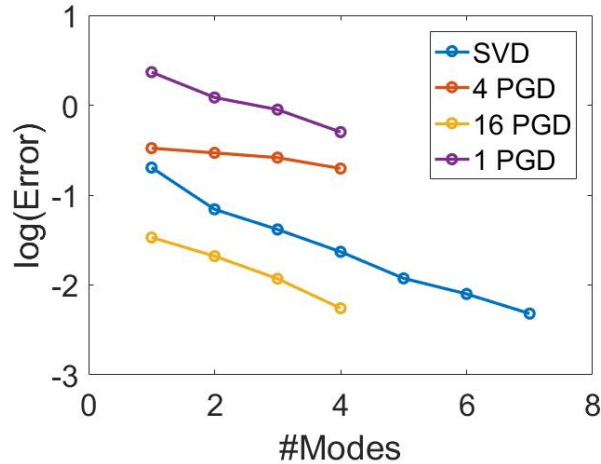


Figure 13. Convergence plot for the convection-diffusion-reaction problem. Comparison between SVD, standard PGD, and two different enrichment strategies.

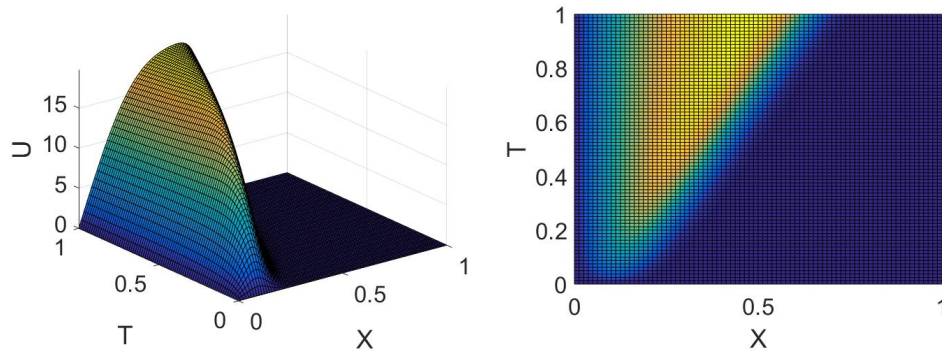


Figure 14. Left, 3D view of $u(x, t)$. Right, top view of $u(x, t)$ for the transient heat transfer equation.

Fig. (14) depicts the reference $u(x, y)$ scalar field for the transient heat equation from different perspectives. Again, a 80×80 finite element mesh of bi-linear elements has been employed. This kind of PDE creates a boundary layer along the diagonal of the space-time domain which is very hard to capture when using standard separate approximations like POD and classical PGD.

Two different enrichment strategies have been tested as shown in Fig. (15). It is important to highlight that PGDs acting on $x = 0$, $x = 1$ and $t = 0$ are set to zero to satisfy the null Dirichlet and initial boundary conditions in this portion of the boundary. However, the PGD acting on the interior nodes of the line $t = 1$ are not set to zero. This fact is a direct consequence of the pure convective behavior that present the time variable. Indeed, the line $t = 1$ acts like an outflow boundary, thus, no condition should be imposed there to ensure the well-posedness of the problem.

Fig. (16) shows the reconstructed solution using four modes per local PGD, when the domain has six active PGDs (left) and twenty PGDs (right). Notice how both approximations capture the main features of the reference solution. However, the approximation with 6 active PGD presents small oscillations due to the lack of PGD modes to converge to the reference solution.

Fig. (17) shows the convergence plot for the SVD and the two different enrichments proposed for the transient heat equation. It can be stated that both partitions present a faster convergence than the SVD convergence. However, the convergence slope of the PGD partitions tend to be flatter than the SVD one in the last part of the convergence plot.

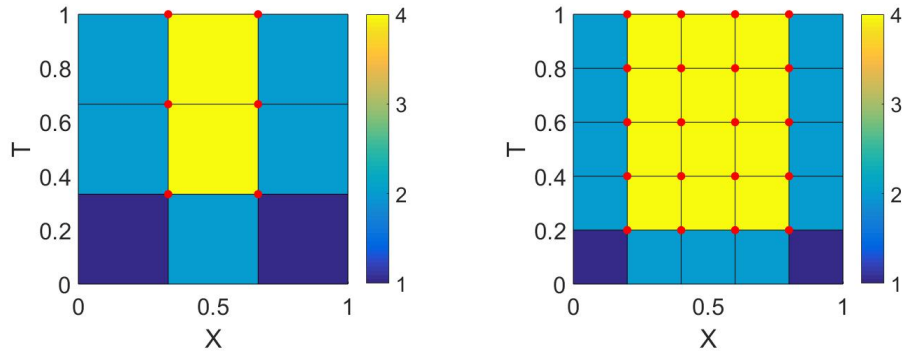


Figure 15. Left, problem solved with 6 PGDs. Right, problem solved with 20 PGDs. Red points represent the finite element nodes that incorporate an enrichment. The legend indicates the number of PGDs per element.

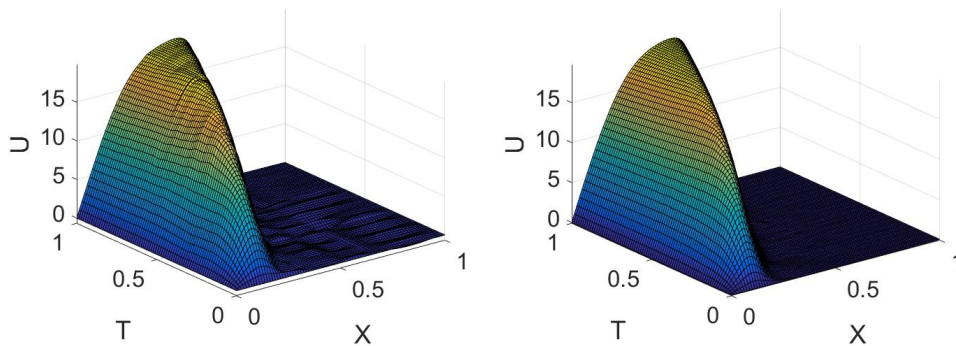


Figure 16. Solution of the transient heat equation. Left, 6 PGDs. Right, 20 PGDs. Both solutions have four modes per PGD.

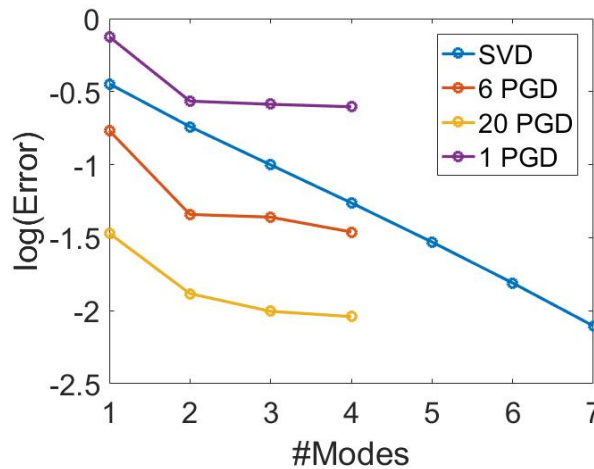


Figure 17. Convergence error for transient heat problem in logarithmic scale for different number of modes. Comparison between SVD, standard PGD, and two different partitions of the domain.

To establish a fair comparison between the methods presented so far, it is interesting to perform a comparison on a per-degree-of-freedom basis. Fig. (18) shows the convergence plot related to

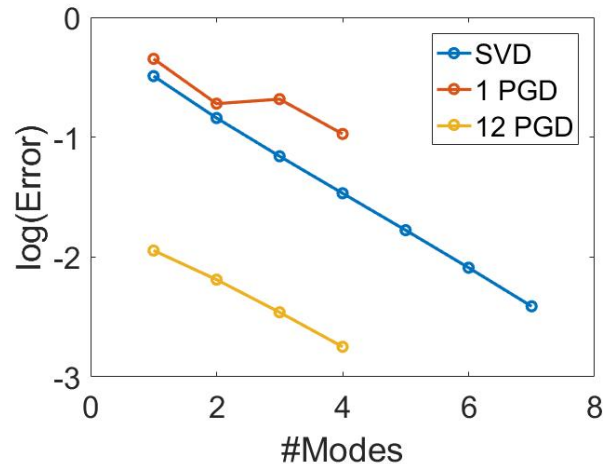


Figure 18. Convergence plot for Idelsohn problem using different partitions of the domain. All approximations have the same number of degrees of freedom.

the transient thermal problem in which the standard PGD approximation has the same degrees of freedom than the *PU*-PGD approximation. It is worth noting the fast convergence of the *PU*-PGD, that comes from the fact that it takes profit of the locality of the solution, avoiding problem of highly non-linear global functions.

5. DISCUSSION. FUTURE WORK

The method presented overcomes some of the most important limitations of standard PGD approximations. These, which are indeed common to every model order reduction technique, are related to the non-linearity of the solution map. As in previous works in the literature, an appealing strategy is to develop local approximations. This approach comes from the well-known fact that a manifold is flat in the neighborhood of every point.

The local, multiple approximations here developed have shown to reconstruct the solution up to a certain tolerance without using a moderate number of modes. However, as previously stated, the size of the mode will increase according to the macro partition of the domain. Therefore, the choice of the macro partition constitutes one of the central keypoints of this approach. A macro partition such that it minimizes the number of macro elements while it reduces the number of modes to represent the solution would be the optimal one. Certainly, a possible future route is to mimic the element refinement in FEM, starting with a coarse macro partition and perform only macro refinement in areas where the residual of the equation is high enough.

One drawback, however, of the proposed technique, is related to the curse of dimensionality in high-dimensional problems. The *PU*-PGD approach introduces a mesh to construct the partition of unity. This may result in prohibitive computational costs if the number of dimensions of the solution grows.

However, as in previous local approaches of PGD [22], there is no need to partition every dimension into elements. Instead, only those parameter interval of interest can be partitioned that provoke the most non-linear response of the solution. Knowing in advance the non-linear shape of the solution manifold is by no means straightforward. This constitutes our current effort of research.

A. ELEMENTAL OPERATORS

In this appendix we give exhaustive details about the construction of the operators required to make the *PU*-PGD approximation for the different PDEs studied in the paper. Detailed information was given for the case of the *PU*-PGD in approximation. Hence, further details will be shown related to the diffusive and convective terms.

A.1. Diffusive Operator

In this subsection a diffusion of a scalar field $u(x, y)$ along the x direction is developed. The derivation for the diffusion along the y direction follows similar guidelines. For the sake of simplicity but without losing generality, we will also assume that a new mode in the x direction is sought. Therefore, all variations related to the y direction are set to zero.

$$\begin{aligned}
& \int_{\Omega_e} \frac{\partial u^*}{\partial x} \frac{\partial u}{\partial x} dx dy = \\
& \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_{\Omega_e} \left(\frac{\partial N_i}{\partial x} X_M^{i*} Y_M^i + N_i \frac{\partial X_M^{i*}}{\partial x} Y_M^i \right) \left(\frac{\partial N_j}{\partial x} X_k^j Y_k^j + N_j \frac{\partial X_k^j}{\partial x} Y_k^j \right) dx dy = \\
& = \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} \frac{\partial N_i^x}{\partial x} \frac{\partial N_j^x}{\partial x} X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy + \\
& + \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x \frac{\partial X_M^{i*}}{\partial x} N_i^x \frac{\partial N_j^x}{\partial x} X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy + \\
& + \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x \frac{\partial X_M^{i*}}{\partial x} N_i^x N_j^x \frac{\partial X_k^j}{\partial x} dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy + \\
& + \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} \frac{\partial N_i^x}{\partial x} N_j^x \frac{\partial X_k^j}{\partial x} dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy \tag{6}
\end{aligned}$$

It can be highlighted that four different contributions are appearing due to the application of the chain rule. Furthermore, all terms appearing in the last part of Eq. (6) present already a separated format.

A.2. Convective Operator

In this subsection a pure convection of a scalar field $u(x, y)$ along the x direction is derived. It is important to notice that this case is a particular case of a general convection given by the velocity field \mathbf{v} , where a separated representation of the velocity field would be required. The derivation for the convection along the y direction follows similar guidelines. For the sake of simplicity but without losing generality, we will also assume that a new mode in the x direction is desired. Therefore, all

variations related to the y direction are set to zero.

$$\begin{aligned}
& \int_{\Omega_e} u^* \frac{\partial u}{\partial x} dx dy = \\
& \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_{\Omega_e} N_i X_M^{i*} Y_M^i \left(\frac{\partial N_j}{\partial x} X_k^j Y_k^j + N_j \frac{\partial X_k^j}{\partial x} Y_k^j \right) dx dy = \\
& = \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} N_i^x \frac{\partial N_j^x}{\partial x} X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy + \\
& + \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} N_i^x N_j^x \frac{\partial X_k^j}{\partial x} dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy
\end{aligned} \tag{7}$$

Once again all integrals appearing in the last part of Eq. (7) are written already in a separated way, improving the efficiency of the algorithm.

REFERENCES

1. Karhunen K. Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae, ser. Al. Math. Phys.* 1946; **37**.
2. Loève MM. *Probability theory*. The University Series in Higher Mathematics, 3rd ed. Van Nostrand, Princeton, NJ, 1963.
3. Park HM, Cho DH. The use of the Karhunen-Loève decomposition for the modeling of distributed parameter systems. *Chemical Engineering Science* 1996; **51**(1):81–98.
4. Meyer M, Matthies HG. Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods. *Computational Mechanics* 2003; **31**(1-2):179–191, doi:10.1007/s00466-002-0404-1. URL <http://dx.doi.org/10.1007/s00466-002-0404-1>.
5. Chinesta F, Ladeveze P (eds.). *Separated Representations and PGD-Based Model Reduction*. Springer International Publishing, 2014.
6. Patera A, Rozza G. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. *Technical Report*, MIT Pappalardo Monographs in Mechanical Engineering 2007.
7. Quarteroni A, Rozza G, Manzoni A. Certified reduced basis approximation for parametrized pde and applications. *J. Math in Industry* 2011; **3**.
8. Rozza G. Fundamentals of reduced basis method for problems governed by parametrized pdes and applications. *CISM Lectures notes “Separated Representation and PGD based model reduction: fundamentals and applications”*, Ladeveze P, Chinesta F (eds.). Springer Verlag, 2014.
9. Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *J. Non-Newtonian Fluid Mech.* 2006; **139**:153–176.
10. Chinesta F, Ammar A, Cueto E. Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering* 2010; **17**(4):327–350.
11. Ladeveze P, Passieux JC, Neron D. The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(21-22):1287 – 1296, doi:10.1016/j.cma.2009.06.023. URL <http://www.sciencedirect.com/science/article/pii/S0045782509002643>.
12. Chinesta F, Cueto E. *PGD-Based Modeling of Materials, Structures and Processes*. Springer International Publishing Switzerland, 2014.
13. Pruliere E, Chinesta F, Ammar A. On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition. *Mathematics and Computers in Simulation* DEC 2010; **81**(4):791–810, doi: {10.1016/j.matcom.2010.07.015}.
14. Mena A, Bel D, Alfaro I, Gonzalez D, Cueto E, Chinesta F. Towards a pancreatic surgery simulator based on model order reduction. *Advanced Modeling and Simulation in Engineering Sciences* 2015; **2**(1):31, doi: 10.1186/s40323-015-0049-1. URL <http://www.amses-journal.com/content/2/1/31>.
15. González D, Cueto E, Chinesta F. Computational patient avatars for surgery planning. *Annals of Biomedical Engineering* 2015; **44**(1):35–45, doi:10.1007/s10439-015-1362-z. URL <http://dx.doi.org/10.1007/s10439-015-1362-z>.
16. Gonzalez D, Cueto E, Chinesta F. Real-time direct integration of reduced solid dynamics equations. *International Journal for Numerical Methods in Engineering* 2014; **99**(9):633–653.
17. Chinesta F, Ammar A, Leygue A, Keunings R. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non Newtonian Fluid Mechanics* 2011; :395–404.
18. Amsallem D, Farhat C. An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA Journal* 2008; **46**:1803–1813.

19. Niroomandi S, Alfaro I, Cueto E, Chinesta F. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering* 2010; **81**(9):1180–1206, doi:10.1002/nme.2733. URL <http://dx.doi.org/10.1002/nme.2733>.
20. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science* 2000; **290**(5500):2323–2326, doi:10.1126/science.290.5500.2323. URL <http://www.sciencemag.org/content/290/5500/2323.abstract>.
21. Scholkopf B, Smola A, Muller K. Kernel principal component analysis. *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*, MIT Press, 1999; 327–352.
22. Badías A, González D, Alfaro I, Chinesta F, Cueto E. Local proper generalized decomposition. *International Journal for Numerical Methods in Engineering* 2017; **112**(12):1715–1732, doi:10.1002/nme.5578. URL <http://dx.doi.org/10.1002/nme.5578>.
23. Babuška I, Melenk JM. The partition of unity finite element method: Basic theory and applications. *Comp. Meth. in Appl. Mech. and Eng.* 1996; **4**:289–314.
24. Babuška I, Melenk JM. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
25. Ammar A, Chinesta F, Cueto E. Coupling finite elements and proper generalized decompositions. *International Journal for Multiscale Computational Engineering* 2011; **9**(1):17–33.
26. Quesada C, González D, Alfaro I, Cueto E, Chinesta F. Computational vademecums for real-time simulation of surgical cutting in haptic environments. *International Journal for Numerical Methods in Engineering* ; **108**(10):1230–1247, doi:10.1002/nme.5252. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5252>.
27. Fish J. The s-version of the finite-element method. *Computers and Structures* 1992; **43**(3):539–547.
28. Rank E, Krause R. A multiscale finite-element method. *Computers and Structures* 1997; **64**(1-4):139 – 144, doi: DOI:10.1016/S0045-7949(96)00149-6. URL <http://www.sciencedirect.com/science/article/B6V28-3SMT86H-13/2/d27473bdcad1af00344fe68925eedede>, computational Structures Technology.
29. Chinesta F, Keunings R, Leygue A. *The Proper Generalized Decomposition for Advanced Numerical Simulations*. Springer International Publishing Switzerland, 2014.
30. Le Bris C, Lelièvre T, Maday Y. Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations. *Constructive Approximation* Oct 2009; **30**(3):621, doi:10.1007/s00365-009-9071-1. URL <https://doi.org/10.1007/s00365-009-9071-1>.
31. Neron D, Ladeveze P. Idelsohns benchmark. *Technical Report* 2013.