

A multi-dimensional data-driven sparse identification technique: the sparse Proper Generalized Decomposition

Rubén Ibáñez¹, Emmanuelle Abisset-Chavanne¹, Amine
Ammar², David González³, Elías Cueto³, Antonio Huerta⁴,
Jean Louis Duval⁵, and Francisco Chinesta¹

¹ESI Chair, ENSAM ParisTech. 151, bvd. de l'Hôpital, F-75013 Paris, France

²LAMPA, ENSAM ParisTech. 2, bvd. de Ronceray. F-49035 Angers, France

³Aragon Institute of Engineering Research. Universidad de Zaragoza. Maria de
Luna, s.n. E-50018 Zaragoza, Spain

⁴Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya. Jordi
Girona 1-3, E-08034 Barcelona, Spain

⁵ESI Group. 99 rue des Solets. F-94150 Rungis, France.

September 25, 2018

Abstract

Sparse model identification by means of data is specially cumbersome if the sought dynamics live in a high dimensional space. This usually involves the need for large amount of data, unfeasible in such a high dimensional settings. This well-known phenomenon, coined as the curse of dimensionality, is here overcome by means of the use of separate representations. We present a technique based on the same principles of the Proper Generalized Decomposition that enables the identification of complex laws in the low-data limit. We provide examples on the performance of the technique in up to ten dimensions.

1 Introduction

In recent years there has been a growing interest in incorporating data-driven techniques into the field of mechanics. While almost classical in other domains of science like economics, sociology, etc., big data has arrived with important delay to the field of computational mechanics. It is worth noting that, in our field, the amount of data available is very often no so big, and therefore we speak of data-driven techniques instead of big-data techniques.

Among the first in incorporating data-driven technologies to the field of computational mechanics one can cite the works of Kirchdoerfer et al. [1, 2], or the ones by Brunton et al. [3] [4] [5]. Previous attempts exist, however, to construct data-driven identification algorithms, see for instance [6] [7]. More recently, the issue of compliance with general laws like the ones of thermodynamics has been also achieved, which is a distinct feature of data-driven mechanics [8]. Other applications include the identification of biological systems [9] or financial trading [10], to name but a few.

The problem with high dimensional systems is that data in these systems is often sparse (due precisely to the high dimensional nature of the phase space) while the system has, on the contrary, low dimensional features—at least very frequently—. Based on this, a distinction should be made between methods that require an a priori structure of the sampling points and others which do not require such a regularity.

Regarding the methods that need a rigid structure in the sampling points, the Non Intrusive Sparse Subspace Learning (SSL) method is a novel technique which has proven to be very effective [11]. The basic ingredient behind such a technique is that the parametric space is explored in a hierarchical manner, where sampling points are collocated at the Gauss-Lobato-Chebyshev integration points. Also, using a hierarchical base allows to improve the accuracy adding more hierarchical levels without perturbing the previous ones. To achieve such hierarchical property, just the difference at a given point between the real function minus the estimated value, using the precedent hierarchical levels, is propagated. For more details about the method, the reader is referred to [11]. However, in the high-dimensional case, this technique shows severe limitations, as will be detailed hereafter.

On the other hand, non-structured data-driven techniques are commonly based on Delaunay triangularization techniques, providing an irregular mesh whose nodes coincides with the sampling points. Afterwards, depending on the degree of approximation inside each one of the Delaunay triangles, it

gives rise to different interpolation techniques, i.e. linear, nearest, cubic, natural, among other techniques are commonly used. Apart from techniques that depend on a given triangularization, it is worth to mention Kriging interpolants as an appealing technique to provide response surfaces from non-structured data points. The key ingredient behind such technique is that each sampling point is considered as a realization of a random process. Therefore, defining a spatial correlation function allows to infer the position of unknown points just like providing confidence intervals based on the distance to the measured points. Nevertheless, the calibration of the correlation matrix has an important impact in the performance of the method itself.

Kriging also possesses a very interesting property: it is able to efficiently filter noise and outliers. Therefore, it is expected that it also could help us in problems with noise in the data.

However, in high dimensional settings, all of the just mentioned techniques fail to identify the nature of the system due precisely to the curse of dimensionality. A recent alternative for such a system could be Topological Data Analysis (TDA), which is based on the use of algebraic topology and the concept of *persistent homology* [12]. A sparse version of this technique also exists [13].

Hence, if a competitive data-driven identification technique is desired, such a technique should meet the following requirements:

- *Non-structured data set*: this characteristic provides versatility to the method. Indeed, when evaluating the response surface requires a lot of computational effort, recycling previous evaluations of the response surface, which do not coincide with a given structure of the data, may be very useful. In addition, the SSL technique establishes sampling points at locations in the phase space with no physical meaning in an industrial setting.
- *Robustness with respect to high dimensionality*: triangularization-based techniques suffer when dealing with multidimensional data just because a high dimensional mesh has to be generated. Nevertheless, the separation of variables could be an appealing technique to circumvent the problem of generating such a high dimensional mesh.
- *Curse of dimensionality*: all previous techniques suffer when dealing with high dimensional data. For instance, the SSL needs 2^D sampling points just to reach the first level of approximation. Thus, when dealing

with high dimensional data ($D > 10$ uncorrelated dimensions) plenty of sampling points are required to properly capture a given response surface.

In what follows we present a method based on the concept of separate representations to overcome the curse of dimensionality. Such separate representation has previously been employed by the authors to construct a priori reduced-order modeling techniques, coined as Proper Generalized Decompositions [14] [15] [16] [17] [18] [19] [20]. This will give rise to a sparse Proper Generalized Decomposition (s-PGD in what follows) approach to the problem. We then analyze the just developed technique through a series of numerical experiments in Section 4, showing the performance of the method. Examples in up to ten dimensions are shown. The paper is completed with some discussions.

2 A sparse PGD (s-PGD) methodology

2.1 Basics of the technique

In this section we develop a novel methodology for sparse identification in high dimensional settings. For the ease of the exposition and, above all, representation, but without loss of generality, let us begin by assuming that the unknown objective function $f(x, y)$ lives in \mathbb{R}^2 and that is to be recovered from sparse data. As in previous references, see for instance [21], we have chosen to begin with a Galerkin projection, in the form

$$\int_{\Omega} w^*(x, y) (u(x, y) - f(x, y)) dx dy = 0, \quad (1)$$

where $\Omega \subset \mathbb{R}^2$ stands for the—here, still two-dimensional—domain in which the identification is performed and $w^*(x, y) \in \mathcal{C}^0(\Omega)$ is an arbitrary test function. Finally, $u(x, y)$ will be the obtained approximation to $f(x, y)$, still to be constructed. In previous works of the authors [8] as well as in other approaches to the problem (e.g., [21]), this projection is subject to additional constraints of thermodynamic nature. In this work no particular assumption is made in this regard, although additional constraints could be imposed to the minimization problem.

Following the same rationale behind the Proper Generalized Decomposition (PGD), the next step is to express the approximated function $u^M(x, y) \approx u(x, y)$ as a set of separate one-dimensional functions,

$$u^M(x, y) = \sum_{k=1}^M X^k(x)Y^k(y). \quad (2)$$

The determination of the precise form of functional pairs $X^k(x)Y^k(y)$, $k = 1, \dots, M$, is done by first projecting them on a finite element basis and by employing a greedy algorithm such that, once the approximation up to order $M - 1$ is known, the new M -th order term

$$u^M(x, y) = u^{M-1}(x, y) + X^M(x)Y^M(y) = \sum_{k=1}^{M-1} X^k(x)Y^k(y) + X^M(x)Y^M(y),$$

is found by any non-linear solver (Picard, Newton, ...).

It is well-known that this approach produces optimal results for elliptic operators (here, note that we have in fact an identity operator acting on u) in two dimensions, see [14] and references therein. There is no proof, however, that this separate representation will produce optimal results (in other words, will obtain *parsimonious models*) in dimensions higher than two. In two dimensions and with $w^* = u^*$ it provides the singular value decomposition of $f(x, y)$ [15]. Our experience, nevertheless, is that it produces almost optimal results in the vast majority of the problems tested so far.

It is worth noting that the product of the test function $w^*(x, y)$ times the objective function $f(x, y)$ is only evaluated at few locations (the ones corresponding to the experimental measurements) and that, in a general high dimensional setting, we will be in the low-data limit necessarily. Several options can be adopted in this scenario. For instance, the objective function can be first interpolated in the high dimensional space (still 2D in this introductory example) and then integrated together with the test function. Indeed, this will be the so-called *PGD in approximation* [15], commonly used when either $f(x, y)$ is known everywhere and a separated representation is sought or if $f(x, y)$ is known in a separated format but a few pairs M are needed for any reason. Under this rationale the converged solution $u(x, y)$ tries to capture the already interpolated solution in the high dimensional space but in a more compact format. As a consequence, the error due to interpolation of experimental measurements on the high dimensional space will persist in the final separate identified function.

In order to overcome such difficulties, we envisage a projection followed by interpolation method. However since information is just known at P sampling points (x_i, y_i) , $i = 1, \dots, P$, it seems reasonable to express the test function not in a finite element context, but to express it as a set of Dirac delta functions collocated at the sampling points,

$$\begin{aligned} w^*(x, y) &= u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) \\ &= (X^*(x)Y^M(y) + X^M(x)Y^*(y)) \sum_{i=1}^P \delta(x_i, y_i), \quad (3) \end{aligned}$$

giving rise to

$$\begin{aligned} &\int_{\Omega} w^*(x, y) (u(x, y) - f(x, y)) dx dy \\ &= \int_{\Omega} u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) (u(x, y) - f(x, y)) dx dy = 0, \end{aligned}$$

The choice of the test function $w^*(x, y)$ in the form dictated by Eq. (3) is motivated by the desire of employing a collocation approach while maintaining the symmetry of standard Bubnov-Galerkin projection operation.

2.2 Matrix form

Let us detail now the finite element projection of the one-dimensional functions $X^k(x)$, $Y^k(y)$, $k = 1, \dots, M$, (often referred to as *modes*) appearing in Eq. (2). Several options can be adopted, ranging from standard piecewise linear shape functions, global non-linear shape functions, maximum entropy interpolants, splines, kriging, etc. Regarding the kind of interpolant to use, an analysis will be performed in the sequel. Nevertheless, no matter which precise interpolant is employed, it can be expressed in matrix form as

$$X^k(x) = \sum_{j=1}^N N_j^k(x) \alpha_j^k = [N_1^k(x) \dots N_N^k(x)] \begin{bmatrix} \alpha_1^k \\ \vdots \\ \alpha_N^k \end{bmatrix} = (\mathbf{N}_x^k)^T \mathbf{a}^k, \quad (4)$$

$$Y^k(y) = \sum_{j=1}^N N_j^k(y) \beta_j^k = [N_1^k(y) \dots N_N^k(y)] \begin{bmatrix} \beta_1^k \\ \vdots \\ \beta_N^k \end{bmatrix} = (\mathbf{N}_y^k)^T \mathbf{b}^k, \quad (5)$$

where α_j^k and β_j^k , $j = 1, \dots, N$, represent the degrees of freedom of the chosen approximation. We employ \mathbf{N}^k as the most usual nomenclature for the shape function vector. It is important to remark that the approximation basis could even change from mode to mode (i.e., for each i). For the sake of simplicity we take the same number of terms for both $X^k(x)$ and $Y^k(y)$, namely, N .

By combining Eqs. (1)-(5) a non linear system of equations is derived, due to products of terms in both spatial directions. An alternate direction scheme is here preferred to linearize the problem, which is also a typical choice in the PGD literature. Note that, when computing modes $X^M(x)$, the variation in the other spatial direction vanishes, $Y^*(y) = 0$, and vice versa.

In order to fully detail the matrix form of the resulting problem, we first employ the notation “ \otimes ” as the standard tensorial product (i.e., $\mathbf{b} \otimes \mathbf{c} = b_i c_j$), and define the following matrices

$$\begin{aligned} \mathbf{A}_x^{k\ell} &= \mathbf{N}_x^k \otimes \mathbf{N}_x^\ell, \\ \mathbf{A}_y^{k\ell} &= \mathbf{N}_y^k \otimes \mathbf{N}_y^\ell, \\ \mathbf{C}_{xy}^{k\ell} &= \mathbf{N}_x^k \otimes \mathbf{N}_y^\ell. \end{aligned}$$

For the sake of simplicity but without loss of generality, evaluations of the former operators at point (x_i, y_i) are denoted as

$$\begin{aligned} \mathbf{A}_{x_i}^{k\ell} &= \mathbf{N}_x^k(x_i) \otimes \mathbf{N}_x^\ell(x_i), \\ \mathbf{A}_{y_i}^{k\ell} &= \mathbf{N}_y^k(y_i) \otimes \mathbf{N}_y^\ell(y_i), \\ \mathbf{C}_{x_i y_i}^{k\ell} &= \mathbf{N}_x^k(x_i) \otimes \mathbf{N}_y^\ell(y_i). \end{aligned}$$

Eqs. (6)-(7) below show the discretized version of the terms appearing in the weak form, Eq. (1), when computing modes in the x direction. Again, M stands for the number of modes in the solution $u(x, y)$ while P denotes the

number of sampling points.

$$\int_{\Omega} u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) u(x, y) dx dy = \sum_{k=1}^M \sum_{i=1}^P ((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{Mk} \mathbf{b}^k) ((\mathbf{a}^*)^T \mathbf{A}_{x_i}^{Mk} \mathbf{a}^k), \quad (6)$$

$$\int_{\Omega} u^*(x, y) \sum_{i=1}^P \delta(x_i, y_i) f(x, y) dx dy = \sum_{i=1}^P f(x_i, y_i) ((\mathbf{a}^*)^T \mathbf{C}_{x_i y_i}^{MM} \mathbf{b}^M). \quad (7)$$

Hence, by defining

$$\begin{aligned} \mathbf{M}_x &= \sum_{i=1}^P ((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{MM} \mathbf{b}^M) \mathbf{A}_{x_i}^{MM}, \\ \mathbf{m}_x &= \sum_{k=1}^{M-1} \sum_{i=1}^P ((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{Mk} \mathbf{b}^k) \mathbf{A}_{x_i}^{Mk} \mathbf{a}^k, \\ \mathbf{f}_x &= \sum_{i=1}^P f(x_i, y_i) \mathbf{C}_{x_i y_i}^{MM} \mathbf{b}^M, \end{aligned}$$

allows to write a system of algebraic equations

$$\mathbf{M}_x \mathbf{a}^M = \mathbf{f}_x - \mathbf{m}_x. \quad (8)$$

Exactly the same procedure is followed to obtain an algebraic system of equations for \mathbf{b}^M . This allows to perform an alternating directions scheme to extract a new couple of $X^M(x)$ and $Y^M(y)$ modes.

This formulation has several aspects that deserve to be highlighted:

1. No assumption about $f(x, y)$ has been made other than assuming known its value at sampling points. Indeed, both problems of either interpolating or making a triangulation in a high dimensional space are circumvented due to the separation of variables.
2. The operator \mathbf{M}_x is composed of P rank-one updates. Meaning that the rank of such operator is at most P . Furthermore, if a subset of measured points share the same coordinate x_i , the entire subset will increase the rank of the operator in one unity.

3. The position of the sampling points will constraint the rank of the PGD operators. That is the reason why, even if the possibility of having a random sampling of points is available, it is always convenient to perform a smart sampling technique such that the rank in each direction tends to be maximized. Indeed, the higher the rank of the PGD operator is, the more cardinality of \mathbf{a} and \mathbf{b} can be demanded without degenerating into an underdetermined system of equations.

There are plenty of strategies to smartly select the position of the sampling points. They are based on either knowing an a priori error indicator or having a reasonable estimation of the sought response surface. Certainly, an adaptive strategy based on the gradient of the precomputed modes could be envisaged. However, the position of the new sampling points will depend on the response surface calculated using the previous sampling points, making parallelization difficult. That is the reason why latin hypercube is chosen in the present work. Particularly, latin hypercube tries to collocate P sampling points in such a way that the projection of those points into x and y axis are as far as possible.

2.3 Choice of the 1D basis

In the previous section, nothing has been specified about the basis in which each one of the one-dimensional modes was expressed. In this subsection, we will use an interpolant based on Kriging techniques. Simple Kriging has been used throughout history in order to get relatively smooth solutions, avoiding spurious oscillations characteristic of high order polynomial interpolation. This phenomena is called Runge's phenomenon. It appears due to the fact that the sampling point locations are not chosen properly, i.e., they will not be collocated, in general, at the Gauss-Lobato-Chebyshev quadrature points. Kriging interpolants consider each point as a realization of a Gaussian process, so that high oscillations are considered as unlikely events.

Hence, by defining a spatial correlation function based on the relative distance between two points, $\mathcal{D}(x_i - x_j) = \mathcal{D}_{ij}$, an interpolant is created over the separated 1D domain,

$$X^k(x) = \sum_{i=1}^N \frac{\alpha_i^k}{N} + \sum_{j=1}^N \lambda(x - x_j) \left(\alpha_j^k - \sum_{l=1}^N \frac{\alpha_l^k}{N} \right),$$

where $\lambda(x - x_j)$ is a weighting function which strongly depends on the definition of the correlation function, and the α_i coefficients are the nodal values associated to the x_i Kriging control points. Note that these control points are not the sampling points. We have chosen this strategy so as to allow us to accomplish an adaptivity strategy that will be described next. In the present work, these control points are uniformly distributed along the 1D domain. Although several definitions of the correlation function exist, a Gaussian distribution is chosen as

$$\mathcal{D}_{ij} = \mathcal{D}(x_i - x_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - x_j)^2}{2\sigma^2}},$$

where σ is the variance of the Gaussian distribution. Several a priori choices can be adopted to select the value of the variance based on the distance between two consecutive control points, e.g., $\sigma = h\sqrt{(x_{i+1} - x_i)^2}$. The magnitude of h should be adapted depending on the desired global character of the support. To ensure the positivity of the variance, h should be in the interval $]0, +\infty[$.

Let us define now a set of C control points

$$\mathbf{x}^{cp} = [x_1^{cp}, x_2^{cp}, \dots, x_C^{cp}],$$

and the P sampling points

$$\mathbf{x}^{sp} = [x_1^{sp}, x_2^{sp}, \dots, x_P^{sp}].$$

Let us define in turn a correlation matrix between all control points and a correlation matrix between the control points and the sampling points as

$$\mathcal{C}_{ij}^{cp-cp} = \mathcal{D}(x_i^{cp} - x_j^{cp}),$$

$$\mathcal{C}_{ij}^{cp-sp} = \mathcal{D}(x_i^{cp} - x_j^{sp}).$$

Under these settings, we define a weighting function for each control point and for each sampling point as

$$\Lambda = (\mathbf{C}^{cp-cp})^{-1} \mathbf{C}^{cp-sp},$$

where $\lambda(x_i^{cp} - x_j^{sp}) = \Lambda_{ij}$.

If we reorganize the terms in the same way that we did in the previous section to have a compact and close format of the shape function \mathbf{N}_x^k , we arrive to

$$X^k(x_j^{sp}) = \sum_{i=1}^N N_i^k(x_j^{sp}) \alpha_i^k = [N_1^k(x_j^{sp}) \dots N_N^k(x_j^{sp})] \begin{bmatrix} \alpha_1^k \\ \vdots \\ \alpha_N^k \end{bmatrix} = (\mathbf{N}_{x_j^{sp}}^k)^T \mathbf{a}^k,$$

where each shape function is given by:

$$N_i^k(x_j^{sp}) = \frac{1 - \sum_{j=1}^N \Lambda_{ij}}{N} + \Lambda_{ij}.$$

Figs. 1-2 depict the appearance of the simple Kriging interpolants using 7 control points uniformly distributed along the domain, for $h = 1$ and $h = \frac{1}{3}$, respectively. It can be highlighted that both the Kronecker delta (i.e., strict interpolation) and partition of unity properties are satisfied for any value of h . Moreover, it is worth noting that the higher the variance the correlation function has, the more global the shape functions are. Furthermore, it is known that 99 per cent of the probability of a Gaussian distribution is comprised within a interval of $[m - 3\sigma, m + 3\sigma]$, being m the mean value of the distribution. This issue explains perfectly well why the support of each Gaussian distribution takes 2 elements for the case where $h = \frac{1}{3}$. Indeed, the shape of the interpolants is quite similar to standard finite element shape functions, but with a Gaussian profile. The remaining 1 per cent of probability is comprised in the small ridges happening in the middle of the elements.

In light of these results, a family of interpolants based on Kriging can be easily created just selecting the value of the variance within the correlation function. Therefore, globality of the support can be easily adjusted always under the framework of the partition of unity.

2.4 Modal adaptivity strategy

In a standard PGD framework, the final solution is approximated as a sum of M modes or functional products, see Eq. (2). Each one of the separated modes must be projected onto a chosen basis to render the problem finite dimensional. A standard choice is to select the same basis for each one of the modes:

$$\mathbf{N}^1 = \mathbf{N}^2 = \dots = \mathbf{N}^M.$$

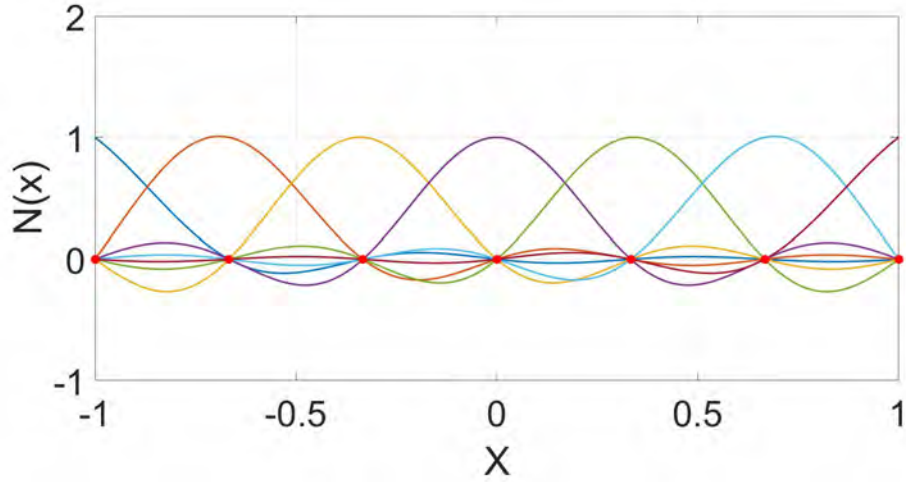


Figure 1: Kriging shape functions using $\sigma = \sqrt{(x_{i+1} - x_i)^2}$ for 7 control points uniformly distributed along the 1D domain.

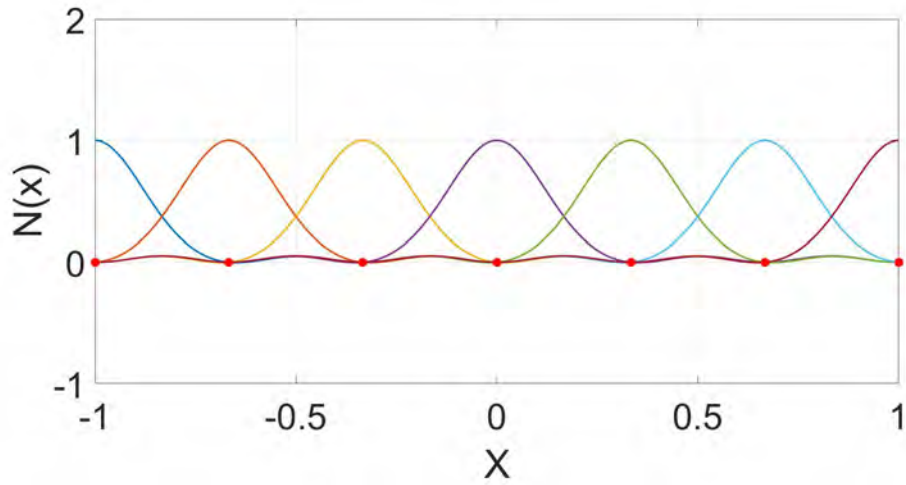


Figure 2: Kriging shape functions using $\sigma = \frac{1}{3}\sqrt{(x_{i+1} - x_i)^2}$ for 7 control points uniformly distributed along the 1D domain.

Despite of the fact that this choice seems reasonable, when dealing with non-structured sparse data, it may not be such. In the previous section we proved that the rank of the separated system strongly depends on the distribution of the data sampling. Therefore, the cardinality of the interpolation basis must not exceed the maximum rank provided by the data sampling. Indeed, this constraint, which provides an upper bound to build the interpolation basis, only guarantees that the minimization is satisfied at the sampling points, without saying anything out of the measured points. Hence, if sampling points are not abundant, in the limit of low-data regime, high oscillations may appear out of these measured points. These oscillations are not desirable since the resulting prediction properties of the proposed method could be potentially decimated.

In order to tackle this problem, we take advantage of the residual-based nature of the PGD. Indeed, the greedy PGD algorithm tries to enrich a solution composed by M modes,

$$u^M(x, y) = \sum_{k=1}^M X^k(x)Y^k(y),$$

just by looking at the residual that accounts for the contribution of the previous modes, as shown in Eq. (8).

Therefore, an appealing strategy to minimize spurious oscillations out of the sampling points is to start the PGD algorithm looking for modes with relatively smooth basis (for instance, Kriging interpolants with a few control points). Therefore, an indicator in order to make an on-line modal adaptive strategy is required. In the present work, we use the norm of the PGD residual,

$$\mathcal{R}_{\mathcal{P}}^M = \frac{1}{\sqrt{P}} \sqrt{\sum_{i \in \mathcal{P}} (f(x_i, y_i) - u^M(x_i, y_i))^2},$$

where \mathcal{P} is the set of P measured points and $f(x, y)$ is the function to be captured.

In essence, when the residual norm stagnates, a new control mesh is defined, composed by one more control point and always uniformly distributed, following

$$\Delta \mathcal{R}_{\mathcal{P}}^M = \mathcal{R}_{\mathcal{P}}^M - \mathcal{R}_{\mathcal{P}}^{M-1} < \epsilon_r.$$

By doing this, oscillations are reduced, since higher-order basis will try to capture only what remains in the residual. Here, ϵ_r is a tolerance defining the

resilience of the sPGD to increase the cardinality of the interpolation basis. The lower ϵ_r is, the more resilient the method is to increase the cardinality.

To better understand the method, we will quantify the error for two set of points: the first set is associated to the sampling points, \mathcal{P} ,

$$\mathcal{E}_{\mathcal{P}} = \frac{1}{\#\mathcal{P}} \sum_{s \in \mathcal{P}} \sqrt{\frac{(f(x_s, y_s) - u^M(x_s, y_s))^2}{f(x_s, y_s)^2}},$$

where $f(x_s, y_s)$ is assumed not to vanish and where \mathcal{L} also includes points other than the sampling points. This is done in order to validate the algorithm, by evaluating the reference solution—which is a priori unknown in a general setting—at points different to the sampling ones,

$$\mathcal{E}_{\mathcal{L}} = \frac{1}{\#\mathcal{L}} \sum_{s \in \mathcal{L}} \sqrt{\frac{(f(x_s, y_s) - u^M(x_s, y_s))^2}{f(x_s, y_s)^2}}.$$

Since the s-PGD algorithm minimizes the error only at the sampling points \mathcal{P} it is reasonable to expect that $\mathcal{E}_{\mathcal{P}} \leq \mathcal{E}_{\mathcal{L}}$.

2.5 A preliminary example

To test the convergence of the just presented algorithm, we consider

$$f_1(x, y) = (\cos(3\pi x) + \sin(3\pi y))y^2 + 4,$$

that presents a quite oscillating behavior along the x direction, whereas the y direction is quadratic. We are interested in capturing such a function in the domain $\Omega_y = \Omega_x = [-1, 1]$.

Figs. (3)-(4) show the errors $\mathcal{E}_{\mathcal{P}}$ and $\mathcal{E}_{\mathcal{L}}$ in identifying the function $f_1(x, y)$. In this case, we consider two distinct possibilities: no modal adaptivity at all, and a modal adaptivity based on the residual, respectively. Several aspects can be highlighted. The first one is that $\mathcal{E}_{\mathcal{P}}$ (asterisks) decreases much faster when there is no modal adaptivity. This is expected, since we are minimizing with a richer basis since the very beginning, instead of starting with smooth functions like in the residual based approach. However, even if the minimization in the sampling points is well achieved, when no modal adaptivity is considered, the error out of the sampling points may increase as the solution is enriched with new modes. Nevertheless, the residual-based

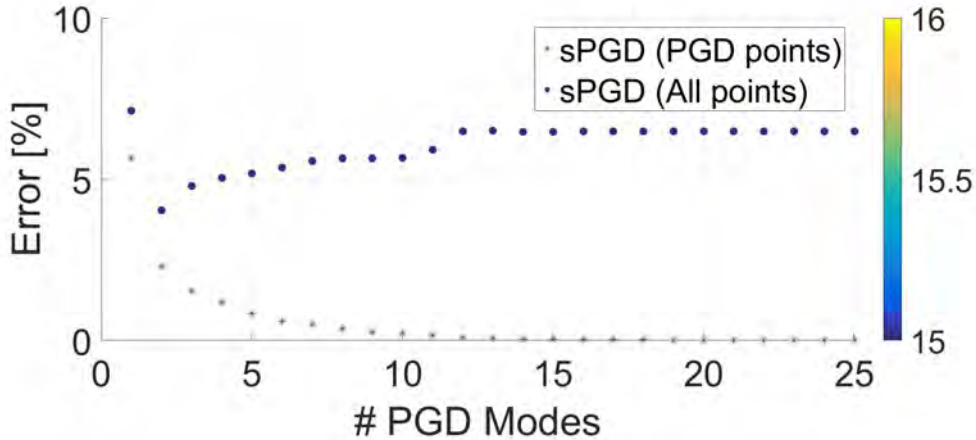


Figure 3: $\mathcal{E}_{\mathcal{L}}$ (points) and $\mathcal{E}_{\mathcal{P}}$ (asterisk) versus the number of modes for $f_1(x, y)$, $\#\mathcal{P} = 100$, $\#\mathcal{L} = 1000$. No modal adaptivity.

modal adaptivity alleviates this problem. As it can be noticed, starting with relatively smooth functions drives the solution out of the sampling points to be smooth as well, avoiding the problem of high oscillations appearing out of the sampling points.

3 A local approach to s-PGD

It is well-known that, as in POD, reduced basis or, in general, any other linear model reduction technique, PGD gives poor results—in the form of a non-parsimonious prediction—when the solution of the problem lives in a highly non-linear manifold. Previous approaches to this difficulty included the employ of non-linear dimensionality reduction techniques such as Locally Linear Embeddings [22], kernel Principal Component Analysis [23] [24] or isomap techniques [25]. Another, distinct, possibility, is to employ a local version of PGD [18], in which the domain is sliced so that at every sub-region PGD provides optimal or nearly optimal results. We explore this last option here for the purpose of sparse regression, although a bit modified, as will be detailed hereafter.

The approach followed herein is based on the employ of the partition of unity property [26] [27]. In essence, it is well-known that any approximat-

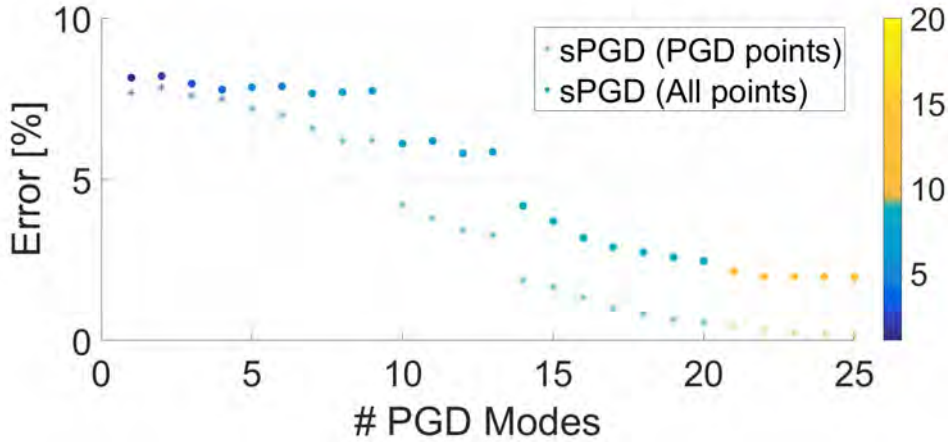


Figure 4: $\mathcal{E}_{\mathcal{L}}$ (points) and $\mathcal{E}_{\mathcal{P}}$ (asterisk) versus the number of modes for $f_1(x, y)$, $\#\mathcal{P} = 100$, $\#\mathcal{L} = 1000$. Modal adaptivity based on the residual, $\epsilon_r=1e-2$.

ing function (like finite element shape functions, for instance) that forms a partition of unity can be enriched with an arbitrary function such that the resulting approximation inherits the smoothness of the partition of unity and the approximation properties of the enriching function.

With this philosophy in mind, we proposed to enrich a finite element mesh with an s-PGD approximation. The resulting approximation will be local, due to the compact support of finite element approximation, while inheriting the good approximation properties, already demonstrated, of s-PGD. In essence, what we propose is to construct an approximation of the type

$$u(x, y) \approx \sum_{i \in \mathcal{I}} N_i(x, y) u_i + \sum_{p \in \mathcal{I}_{\text{enr}}} \sum_{e \in \mathcal{I}_p} N_e(x, y) \underbrace{\sum_{k=1}^M X_p^k(x) Y_p^k(y)}_{f_p^{\text{enr}}(x, y)},$$

where \mathcal{I} represents the node set in the finite element mesh, \mathcal{I}_{enr} the set of enriched nodes, u_i are the nodal degrees of freedom of the mesh, \mathcal{I}_p is the number of finite elements covered by node p shape function's support and $X_p^k(x)$ and $Y_p^k(x)$ functions are the k -th one-dimensional PGD modes enriching node p , that in fact constitute an enriching function $f_p^{\text{enr}}(x, y)$.

Of course, as already introduced in Eqs. (4) and (5), every PGD mode is in turn approximated by Galerkin projection on a judiciously chosen basis. In other words,

$$u(x, y) \approx \sum_{i \in \mathcal{I}} N_i(x, y) u_i + \sum_{p \in \mathcal{I}_{\text{enr}}} \sum_{e \in \mathcal{I}_p} N_e(x, y) \sum_{k=1}^M (\mathbf{N}_x^k)^T \mathbf{a}_p^k (\mathbf{N}_y^k)^T \mathbf{b}_p^k, \quad (9)$$

with \mathbf{a}_p^k and \mathbf{b}_p^k the nodal values describing each one-dimensional PGD mode.

In this framework, the definition of a suitable test function can be done in several ways. As a matter of fact, the test function can be expressed as the sum of a finite element and a PGD contribution,

$$u^* = u_{FEM}^* + u_{PGD}^*,$$

so that an approach similar to that of Eq. (3) can be accomplished.

An example of the performance of this approach is included in Section 4.4.

4 Numerical results

The aim of this section is to compare the ability of sparse model identification for different interpolation techniques. On one hand, the performance of standard techniques based on Delaunay triangulation such as linear, nearest neighbor or cubic interpolation is compared. Even though these techniques are simple, they allow to have a non-structured sampling point set since they rely on a Delaunay triangulation. On the other hand, the results are compared to the Sparse Subspace Learning (SSL) [11]. The convergence and robustness of this method is proven to be very effective since the points are collocated at the Gauss-Lobato-Chebyshev points. However, two main drawbacks appear considering this method. The first one is that there is a high concentration of points in the boundary of the domain, so that this quadrature is meant for functions that vary mainly along the boundary. Indeed, if the variation of the function appears in the middle of the domain, many sampling points will be required to converge to the exact function. The second one is that the sampling points have to be located at specific points in the domain. The s-PGD method using simple Kriging interpolants will be compared as well.

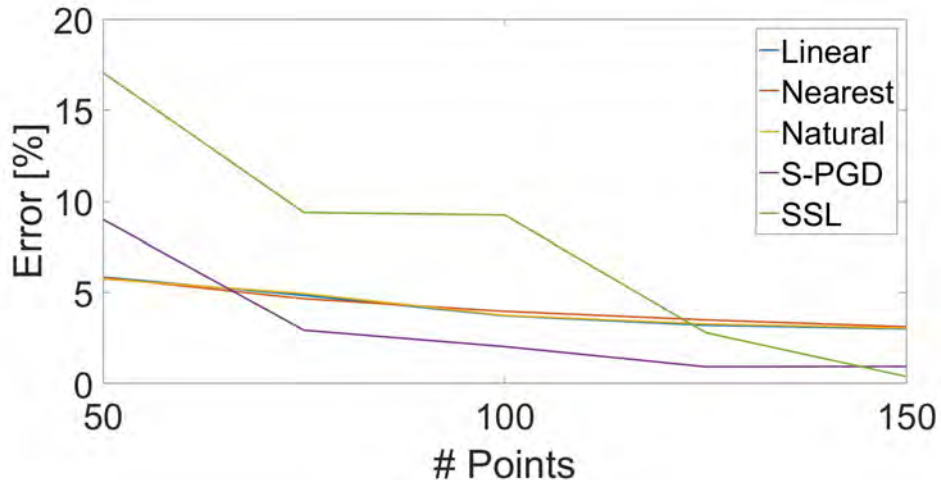


Figure 5: $\mathcal{E}_{\mathcal{L}}$ of $f_1(x, y)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$.

The numerical results are structured as follows: first two synthetic 2D functions are analyzed; secondly, two 2D response surfaces coming from a thermal problem and a Plastic Yield function are reconstructed; finally, a 10D synthetic function is reconstructed by means of the s-PGD algorithm.

4.1 2D synthetic functions

The first considered function is $f_1(x, y)$, as introduced in the previous section. Fig. 5 shows the reconstruction error ($\mathcal{E}_{\mathcal{L}}$) of $f_1(x, y)$ for different sampling points. As it can be noticed, the s-PGD algorithm performs well for a wide range of sampling points. Nevertheless, the SSL method is the one presenting the lower error level when there are more than 150 sampling points.

A second synthetic function is defined as

$$f_2(x, y) = \cos(3xy) + \log(x + y + 2.05) + 5.$$

This function is intended to be reconstructed in the domain $\Omega_x = \Omega_y = [-1, 1]$. It was chosen in such a way that it is relatively smooth in the center of the domain, whereas the main variation is located along the boundary of the domain. Indeed, this function is meant to show the potential of the SSL technique.

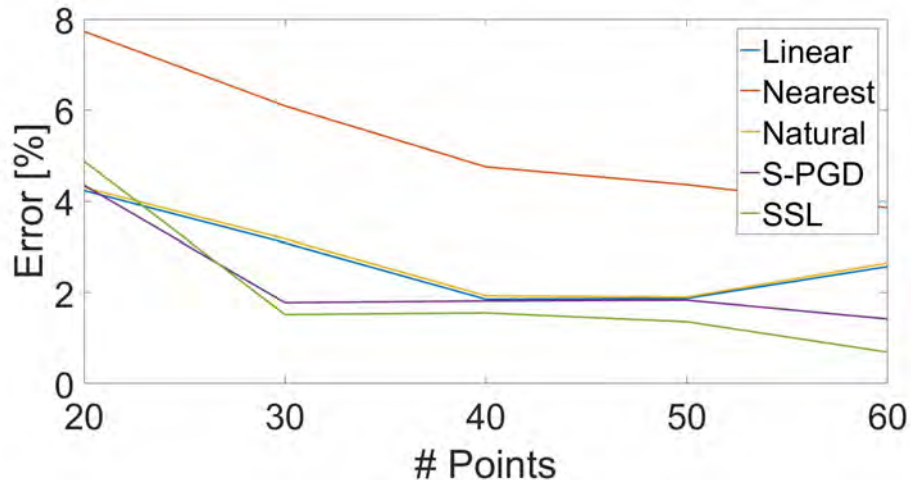


Figure 6: $\mathcal{E}_{\mathcal{L}}$ of $f_2(x, y)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$.

Fig. 6 shows the reconstruction error of the $f_2(x, y)$ function for different interpolation techniques. As it can be noticed, both SSL and s-PGD methods are the ones that present the best convergence properties. If the number of points is increased even more, the SSL method is the one that presents the lowest interpolation error. They are followed by linear and natural neighbor interpolations. Finally, the nearest neighbor method is the one presenting the worst error for this particular case.

4.2 2D response surfaces coming from physical problems

Once the convergence of the methods have been unveiled for synthetic functions, it is very interesting to analyze the power of the former methods by trying to identify functions that are coming from either simulations or models popular in the computational mechanics community. Indeed, two functions will be analyzed: the first one is an anisotropic Plastic Yield function, whereas the second one is a solution coming from a quasi-static thermal problem with varying source term and conductivity.

Fig. 7 shows the Yld2004-18p anisotropic plastic yield function, defined

by Barlat et al. in [28]. Under plane stress hypothesis, this plastic yield function is a convex and closed surface defined in a three-dimensional space. Therefore, the position vector of an arbitrary point in the surface can be easily parameterized in cylindrical coordinates as $R(\theta, \sigma_{xy})$. The $R(\theta, \sigma_{xy})$ function for the Yld2004-18p is shown in Fig. 8, where anisotropies can be easily seen. Otherwise, the radius function will be constant for a given σ_{xy} .

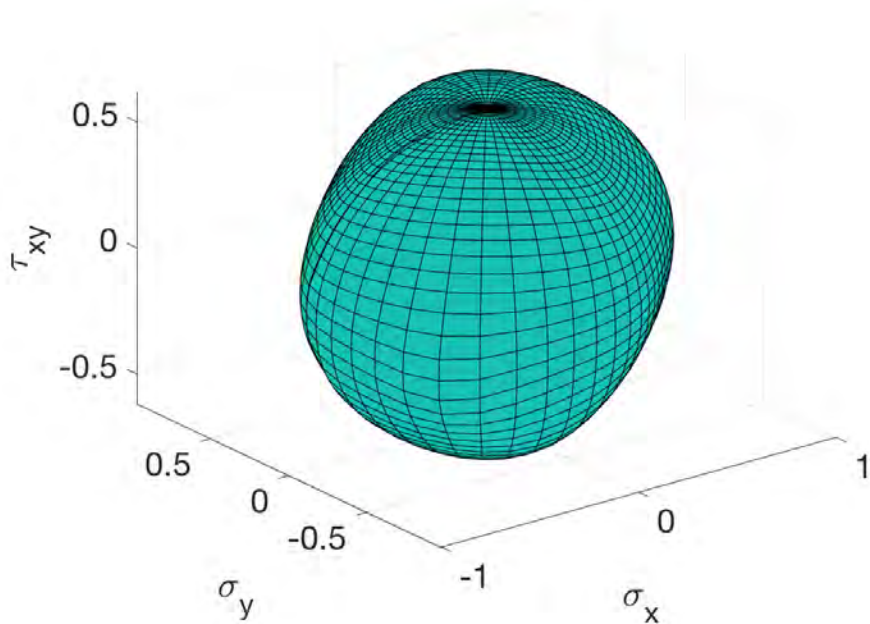


Figure 7: Barlat's Yld2004-18p function under plane stress hypothesis.

Fig. 9 shows the error in the identification of the Barlat's plastic yield function Yld2004-18p. As it can be noticed, the s-PGD technique outperforms the rest of techniques. Indeed, the s-PGD is exploiting the fact that the response surface is highly separable.

As mentioned above, the second problem is the sparse identification of the solution of a quasi-static thermal problem modeled by

$$\nabla \cdot (\eta(x, t) \nabla(u(x, t))) = f(t), \quad \text{in } \Omega_x \times \Omega_t = [-1, 1] \times [-1, 1], \quad (10)$$

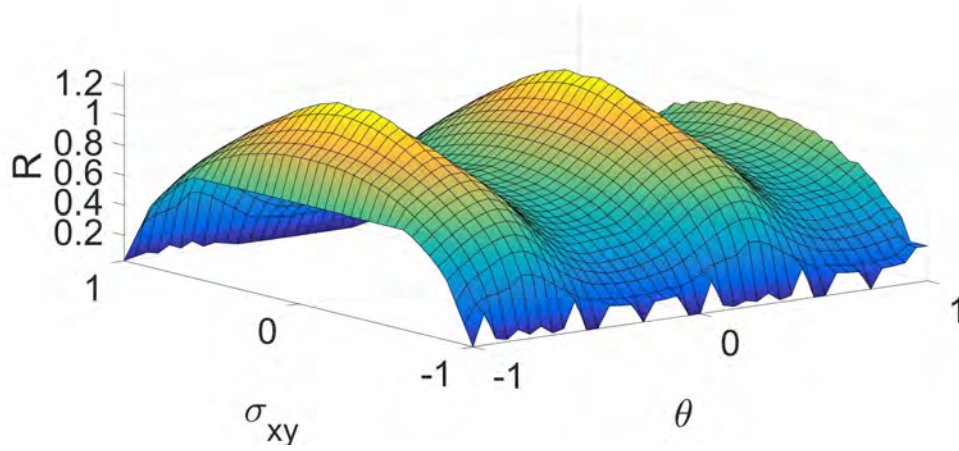


Figure 8: $R(\theta, \sigma_{xy})$ function for Barlat's Yld2004-18p yield function.

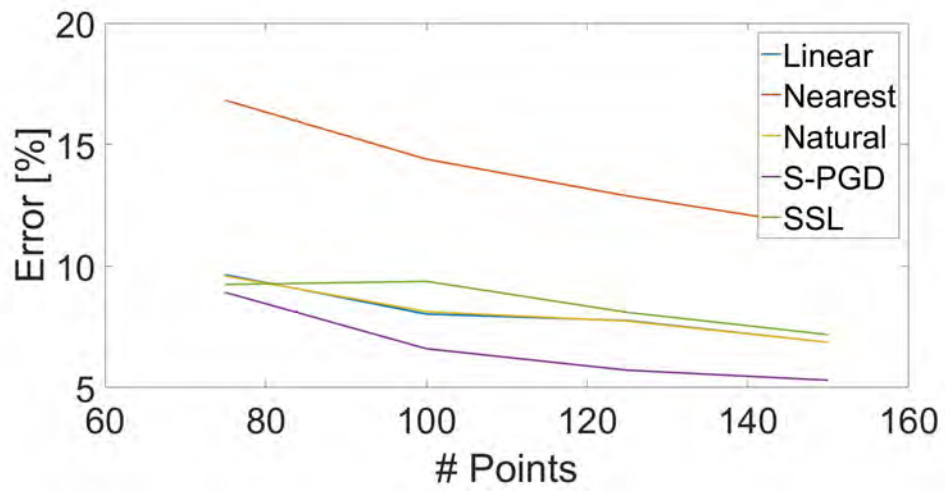


Figure 9: $\mathcal{E}_{\mathcal{L}}$ of $R(\theta, \sigma_{xy})$ varying $\#\mathcal{P}$ for different sparse identification techniques. $\#\mathcal{L} = 1000$. $\epsilon_r = 5 \cdot 10^{-4}$.

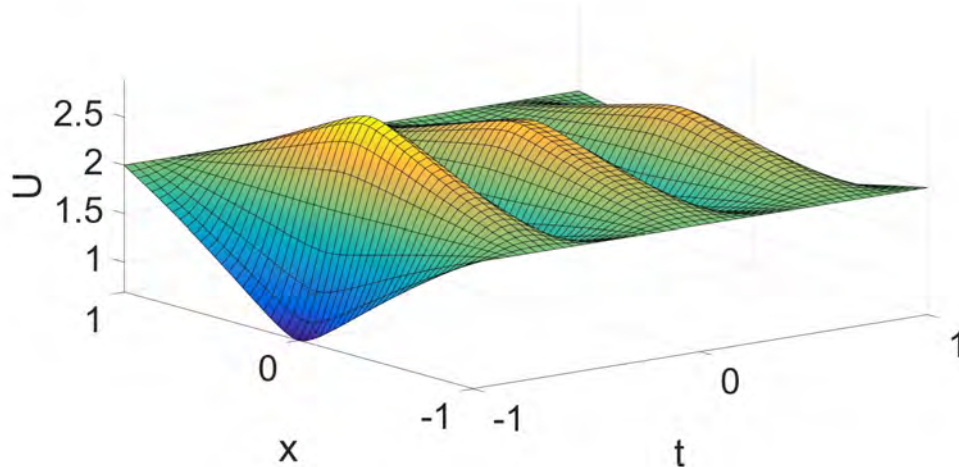


Figure 10: Quasi-static solution to the thermal problem $u(x, t)$.

where conductivity varies in space-time as

$$\eta(x, t) = (1 + 10 \text{abs}(x) + 10x^2) \log(t + 2.5) \quad u(1, t) = 2 \quad (11)$$

$$f(x, t) = 10 \cos(3\pi t) \quad u(-1, t) = 2, \quad (12)$$

and the source term varies in time. Homogeneous Dirichlet boundary conditions are imposed at both spatial boundaries and no initial conditions are required due to quasi-stationarity assumptions.

Fig. 10 shows the evolution of the temperature field as a function of space time for the set of Eqs. (10)-(12). It can be noticed how the variation of the temperature throughout time is caused mainly due to the source term. However, conductivity modifies locally the curvature of the temperature along the spatial axis. Symmetry with respect the $x = 0$ axis is preserved due to the fact that the conductivity presents a symmetry along the same axis.

Fig. 11 shows the performance of each one of the techniques when trying to reconstruct the temperature field from certain sampling points. As can be noticed, the s-PGD in conjunction with Kriging interpolants is the one that presents the fastest convergence rate to the actual function, which is considered unknown. It is followed by linear and natural interpolations. The SSL method presents a slow convergence rate in this case, due to the fact that the main variation of the function $u(x, t)$ is happening in the center of

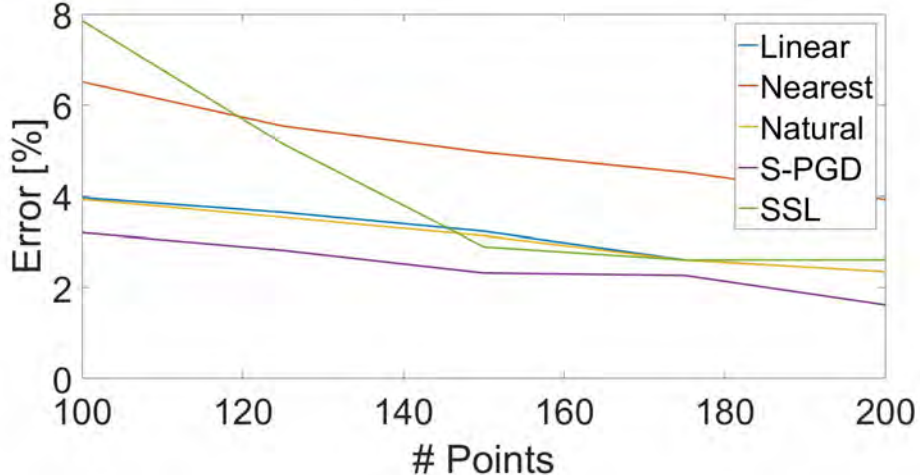


Figure 11: $\mathcal{E}_{\mathcal{L}}$ of $u(x, t)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$. $\epsilon_r = 2.5 \cdot 10^{-3}$.

the domain and not in the boundary.

4.3 A 10D multivariate case

In this subsection, we would like to show the scalability that s-PGD presents when dealing with relatively high-dimensional spaces. Since our solution is expressed in a separated format, an N dimensional problem (ND) is solved as a sequence of N 1D problems, which are solved using a fixed-point algorithm in order to circumvent the non-linearity of the separation of variables.

The objective function that we have used to analyze the properties of the s-PGD is defined as

$$f_3(x_1, x_2, \dots, x_N) = 2 + \frac{1}{8} \sum_{i=1}^N x_i + \prod_{i=1}^N x_i + \prod_{i=1}^N x_i^2,$$

with $N = 10$ in this case.

Fig. 12 shows the error convergence in both sampling points ($\mathcal{E}_{\mathcal{P}}$, asterisks) and points out of the sampling ($\mathcal{E}_{\mathcal{L}}$, filled points). The \mathcal{L} data set was composed by 3000 points, the \mathcal{P} data subset for the s-PGD algorithm was composed by 500 points. The number of points required to properly capture

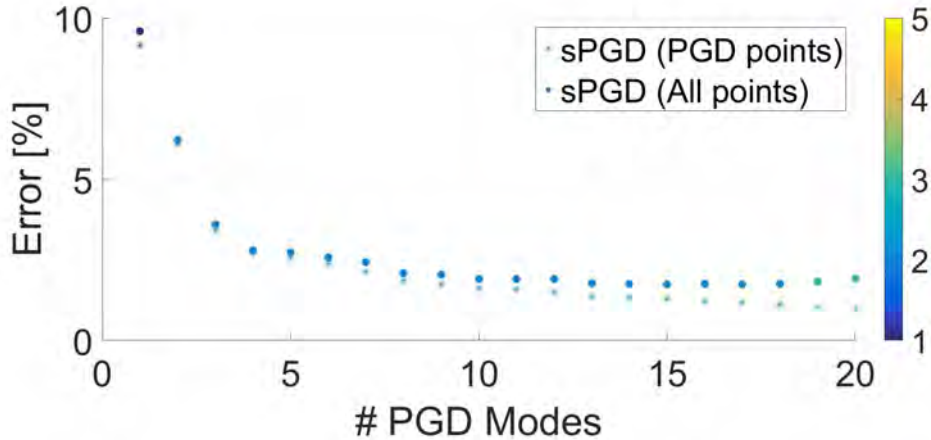


Figure 12: $\mathcal{E}_{\mathcal{L}}$ (points) and $\mathcal{E}_{\mathcal{P}}$ (asterisk) versus the number of modes for $f_3(x_1, x_2, \dots, x_N)$, $\#\mathcal{P} = 500$, $\#\mathcal{L} = 3000$. Modal adaptivity based on the residual, $\epsilon_r = 1e - 3$.

the hyper-surface has increased with respect to the 2D examples due to the high dimensionality of the problem. Special attention has to be paid when increasing the cardinality of the interpolant basis without many sampling points, because the problem of high oscillations outside the control points may be accentuated.

4.4 An example of the performance of the local s-PGD

The last example corresponds to the sparse regression of an intricate surface, that has been created by mixing three different Gaussian surfaces so as to generate a surface with no easy separate representation (a non parsimonious model, if we employ a different vocabulary). The appearance of this synthetic surface is shown in Fig. 13.

The sought surface is defined in the domain $\Omega = [0, 1]^2$, which has been split into the finite element mesh shown in Fig. 14. Every element in the mesh has been colored according to the number of enriching PGD functions, ranging from a single one to four. The convergence plot of this example as a function of the number of PGD modes added to the approximating space is included in Fig. 15.

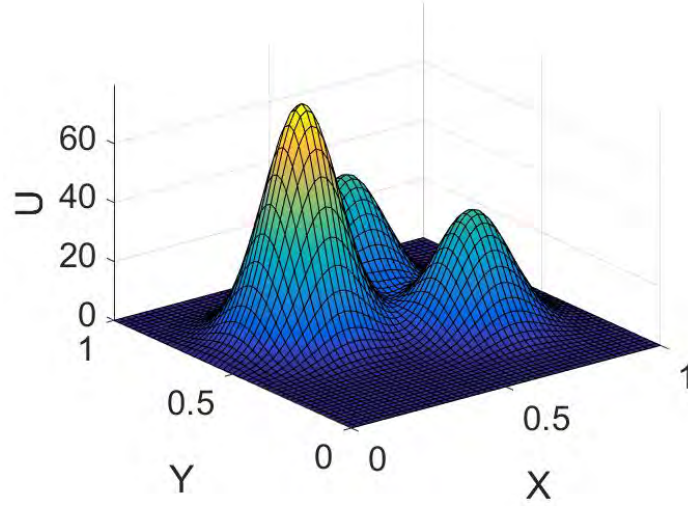


Figure 13: A synthetic surface generated by superposition of three different Gaussians, that is to be approximated by local s-PGD techniques.

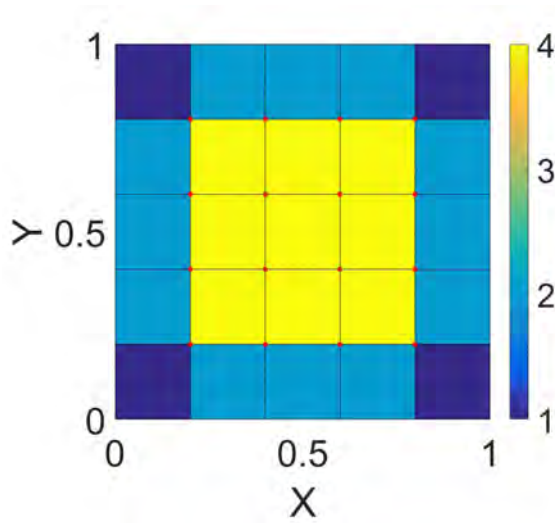


Figure 14: Finite element mesh for the example in Section 4.4. All the internal nodes have been enriched.

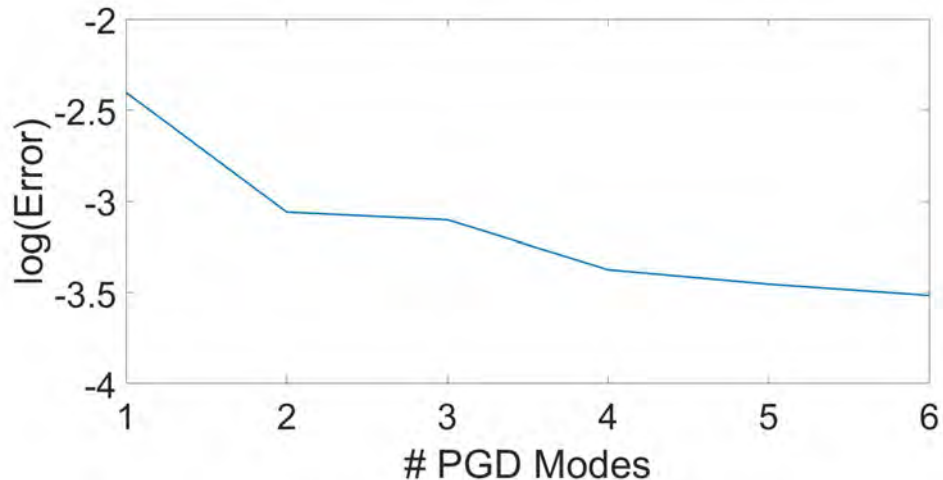


Figure 15: Convergence plot for the example in Section 4.4.

5 Conclusions

In this paper we have developed a data-based sparse reduced-order regression technique under the Proper Generalized Decomposition framework. This algorithm combines the robustness typical of the separation of variables together with properties of collocation methods in order to provide with parsimonious models for the data at hand. The performance of simple Kriging interpolation has proven to be effective when the sought model presents some regularity. Furthermore, a modal adaptivity technique has been proposed in order to avoid high oscillations out of the sampling points, characteristic of high order interpolation methods when data is sparse.

For problems in which the result lives in a highly non-linear manifold, a local version of the technique, that makes use of the partition of unity property, has also been developed. This local version outperforms the standard one for very intricate responses.

The s-PGD method has been compared advantageously versus other existing methods for different example functions. Finally, the convergence of s-PGD method for a high dimensional function has been demonstrated as well.

Although the sparsity of the obtained solution could not seem evident

for the reader, we must highlight the fact that the very nature of the PGD strategy *a priori* selects those terms in the basis that play a relevant role in the approximation. So to speak, PGD algorithms automatically discard those terms that in other circumstances will be weighted by zero. Sparsity, in this sense, is equivalent in this context to the number of sums in the PGD separated approximation. If only a few terms are enough to reconstruct the data—as is almost always the case—, then sparsity is guaranteed in practice.

Sampling strategies other than the latin hypercube method could be examined as well. This, and the coupling with error indicators to establish good stopping criteria, constitute our effort of research at this moment. In fact, the use of reliable error estimators could even allow for the obtention of adaptive samplings in which the cardinality of the basis could be different along different directions.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 675919. Also by the Spanish Ministry of Economy and Competitiveness through Grants number DPI2017-85139-C2-1-R and DPI2015-72365-EXP and by the Regional Government of Aragon and the European Social Fund, research group T24 17R.

References

- [1] T. Kirchdoerfer and M. Ortiz. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 304:81 – 101, 2016.
- [2] T. Kirchdoerfer and M. Ortiz. Data driven computing with noisy material data sets. *Computer Methods in Applied Mechanics and Engineering*, 326:622 – 641, 2017.
- [3] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of non-linear dynamical systems. *PROCEEDINGS OF THE NATIONAL*

ACADEMY OF SCIENCES OF THE UNITED STATES OF AMERICA, 113(15):3932–3937, APR 12 2016.

- [4] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Sparse identification of nonlinear dynamics with control (sindyc). *arXiv preprint arXiv:1605.06682*, 2016.
- [5] Markus Quade, Markus Abel, J. Nathan Kutz, and Steven L. Brunton. Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):063116, 2018.
- [6] D. Gonzalez, F. Masson, F. Poulhaon, E. Cueto, and F. Chinesta. Proper generalized decomposition based dynamic data driven inverse identification. *Mathematics and Computers in Simulation*, 82:1677–1695, 2012.
- [7] Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196 – 215, 2016.
- [8] David González, Francisco Chinesta, and Elías Cueto. Thermodynamically consistent data-driven computational mechanics. *Continuum Mechanics and Thermodynamics*, May 2018.
- [9] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, June 2016.
- [10] Jordan Mann and J. Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, 16(11):1643–1655, 2016.
- [11] Domenico Borzacchiello, José V. Aguado, and Francisco Chinesta. Non-intrusive sparse subspace learning for parametrized problems. *Archives of Computational Methods in Engineering*, Jul 2017.
- [12] Charles Epstein, Gunnar Carlsson, and Herbert Edelsbrunner. Topological data analysis. *Inverse Problems*, 27(12):120201, 2011.

- [13] W. Guo, K. Manohar, S. L. Brunton, and A. G. Banerjee. Sparse-tda: Sparse realization of topological data analysis for multi-way classification. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1403–1408, July 2018.
- [14] F. Chinesta, A. Ammar, and E. Cueto. Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering*, 17(4):327–350, 2010.
- [15] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations*. Springer International Publishing Switzerland, 2014.
- [16] F. Chinesta and P. Ladeveze, editors. *Separated Representations and PGD-Based Model Reduction*. Springer International Publishing, 2014.
- [17] D. Gonzalez, A. Ammar, F. Chinesta, and E. Cueto. Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81(5), 2010.
- [18] Alberto Badiás, David González, Iciar Alfaro, Francisco Chinesta, and Elias Cueto. Local proper generalized decomposition. *International Journal for Numerical Methods in Engineering*, 112(12):1715–1732, 2017. nme.5578.
- [19] E. Cueto, D. González, and I. Alfaro. *Proper Generalized Decompositions: An Introduction to Computer Implementation with Matlab*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, 2016.
- [20] David González, Alberto Badiás, Iciar Alfaro, Francisco Chinesta, and Elías Cueto. Model order reduction for real-time data assimilation through Extended Kalman Filters. *Computer Methods in Applied Mechanics and Engineering*, 326(Supplement C):679 – 693, 2017.
- [21] Jean-Christophe Loiseau and Steven L. Brunton. Constrained sparse galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.
- [22] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

- [23] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Non-linear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.
- [24] B. Scholkopf, A. Smola, and K.R. Muller. Kernel principal component analysis. In *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*, pages 327–352. MIT Press, 1999.
- [25] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [26] I. Babuška and J. M. Melenk. The partition of unity finite element method: Basic theory and applications. *Comp. Meth. in Appl. Mech. and Eng.*, 4:289–314, 1996.
- [27] I. Babuška and J. M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.
- [28] J.W. Yoon, F. Barlat, R.E. Dick, and M.E. Karabin. Prediction of six or eight ears in a drawn cup based on a new anisotropic yield function. *International Journal of Plasticity*, 22(1):174 – 193, 2006.