Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/18681

To cite this version :

David GONZÁLEZ, José Vicente AGUADO, Emmanuelle ABISSET-CHAVANNE, Francisco CHINESTA - kPCA-Based Parametric Solutions Within the PGD Framework - Archives of Computational Methods in Engineering - Vol. 25, p.69-86 - 2018

Any correspondence concerning this service should be sent to the repository Administrator : archiveouverte@ensam.eu

Arts et Métiers Institute of Technology

kPCA-Based Parametric Solutions Within the PGD Framework

D. González[1] · J. V. Aguado[2] · E. Cueto[1] · E. Abisset-Chavanne[3] · F. Chinesta[2]

Abstract Parametric solutions make possible fast and reliable real-time simulations which, in turn allow real time optimization, simulation-based control and uncertainty propagation. This opens unprecedented possibilities for robust and efficient design and real-time decision making. The construction of such parametric solutions was addressed in our former works in the context of models whose parameters were easily identified and known in advance. In this work we address more complex scenarios in which the parameters do not appear explicitly in the model—complex microstructures, for instance. In these circumstances the parametric model solution requires combining a technique to find the relevant model parameters and a solution procedure able to cope with high-dimensional models, avoiding the well-known curse of dimensionality. In this work, kPCA (kernel Principal Component Analysis) is used for extracting the hidden model parameters, whereas the PGD (Proper Generalized Decomposition) is used for calculating the resulting parametric solution.

✉ D. González
gonzal@unizar.es

J. V. Aguado
Jose.Aguado-Lopez@ec-nantes.fr

E. Cueto
ecueto@unizar.es

E. Abisset-Chavanne
Emmanuelle.Abisset-Chavanne@ec-nantes.fr

F. Chinesta
francisco.chinesta@ec-nantes.fr

[1] I3A, Universidad de Zaragoza, Maria de Luna s/n, 50018 Zaragoza, Spain

[2] ESI Chair "Advanced Computational Manufacturing Processes", ICI - High Performance Computing Institute @ Ecole centrale de Nantes, Institut Universtaire de France, 1 rue de la Noe, BP 92101, 44321 Nantes Cedex 3, France

[3] GEM, UMR CNRS - Centrale Nantes, 1 rue de la Noe, BP 92101, 44321 Nantes Cedex 3, France

1 Introduction to Parametric Modeling

Many problems related to important societal and industrial challenges require decision-making procedures to be accomplished fast and reliably. These are in general data-driven and arise from complex models expressed in the form of partial differential equations. They involve usually enormous amounts of information. In addition, very often, solutions are needed in real-time. Moreover, there is an industrial claim towards "democratization" of simulation, so that these simulations can be employed by non-specialists running deployed platforms such as smartphones or tablets. To our knowledge, the solution of such complex computational models has been very often addressed by employing high-performance computing running in supercomputers. It is expected that, in the near future, real-time simulation, optimization and control in applied sciences and engineering will be achieved by extensive usage of supercomputing frameworks. Consequently, it is expected that important advances in hardware and software for high-performance computing will be achieved. On the contrary, there is also an alternative approach to this end with an eye towards the development of as simple as possible models (within a prescribed degree of simulation realism, of course!). It can now be foreseen that a new generation of simulation techniques, beyond high-performance computing, will be developed so as to improve efficiency or simply to allow obtaining results in such challenging scenarios.

As mentioned before, many problems in applied science and engineering remain intractable, in spite of the impressive progresses attained in modeling, numerical analysis, discretization techniques and computer science during the last decade. This is because their numerical complexity, or the restrictions imposed by different requirements make them unaffordable for today's technologies. Many problems in the fields of parametric modeling, inverse identification, process or shape optimization, usually require, when standard techniques are employed, the direct computation of a very large number of solutions of the model for particular values of the parameters. When the number of parameters increases such a procedure becomes intractable.

The human being has developed throughout history distinct facilities for giving fast responses to these questions. Thus, abaci were already used 2700 years B.C. in Mesopotamia, for instance, as a means to cope with parametric problems. However, the initial arithmetic was rapidly complemented with more complex representations; some of them were the charts and the nomograms. The just mentioned abaci allowed for fast calculations and data manipulations. Nomograms can be easily constructed when the mathematical relationship that they express is purely algebraic, eventually nonlinear. In these cases it was easy to represent some outputs as a function of some inputs. Computations necessary for such data representations were performed "offline" and then used "online" in many branches of engineering sciences for design and optimization.

However, these procedures fail when addressing more complex scenarios. Thus, sometimes engineers dealt with non-properly understood physics, and in that case the construction of nomograms based on a too rude modeling could be dangerous. Under these circumstances one could proceed by making several experiments from which defining a sort of experiment-based nomogram. In other cases, mathematical objects to be manipulated consisted of a system of complex coupled nonlinear partial differential equations, whose solution for each possible combination of the involved parameter values is simply out of reach for modern computational resources. In these cases, it becomes necessary to design a set of experiments or expensive computational solutions for a sampling of possible states of the system. For these, a simplified model, linking the inputs to the outputs of interest is elaborated. These simplified models have different names: surrogate models, meta-models, response-surface methodologies, etc.

More recently, model order reduction (MOR) opened new possibilities. MOR based upon techniques such as Proper Orthogonal Decomposition, Proper Generalized Decomposition or Reduced Basis is nowadays widely considered from both fundamental and applicative viewpoints.

Proper Orthogonal Decomposition (POD, also known as Principal Component Analysis, PCA) is a general technique for extracting the most significant characteristics of a system's behavior and representing them in a set of "POD basis vectors" [36, 44]. These basis vectors then provide an efficient, low-dimensional representation of the system behavior, which proves useful in a variety of ways. The most common use is to project governing equations onto the reduced-order subspace spanned by the POD basis vectors. This yields an explicit POD reduced model that can be solved in place of the original system. The POD basis can also provide a low-dimensional description in which to perform parametric interpolation, infill missing or "gappy" data, and perform model adaptation. There exists an extensive literature on the topic and POD has been applied broad application across fields [8, 9, 13, 16, 28, 58, 67, 68, 70, 73]. Some review of POD and its applications to model order reduction can be found in [25, 57, 75].

Another family of model reduction techniques lies in the use of Reduced Basis constructed by combining a greedy algorithm and "a posteriori" error indicators. As for the POD, the Reduced Basis method requires some amount offline work. Once computed, however, the reduced basis approach can be used online with the notable advantage of a rigorous control of the solution accuracy, thanks to the availability of error bounds. The reduced basis can be enriched if the attained error is judged too high, by invoking a greedy adaption strategy [47, 48, 65]. Useful review works on the subject are [34, 49, 59, 61, 64, 66].

Techniques based on the use of separated representations are at the heart of the so-called Proper Generalized Decomposition methods. Such separated representations are rooted in the very classical method of separation of variables due to Fourier. More recently, they have been applied to quantum chemistry for approximating multidimensional quantum wave-functions, e.g. Hartree-Fock and post-Hartree-Fock methods [17]. In the eighties, Pierre Ladeveze proposed the use of space-time separated representations of transient solutions arising in strongly nonlinear models, defining a non-incremental integration procedure [37, 38]. Separated representations were then employed for solving highly multidimensional models. These suffer the so-called *curse of dimensionality* [3, 4, 42] and in the context of stochastic modeling [56]. They soon were extended for separating space coordinates, thus making possible the solution of models defined in degenerated domains such as plate and shells [14, 15] as well as for addressing parametric models, where model parameters were considered as model extra-coordinates. This "extra-coordinate" assumption made possible an offline calculation of the parametric solution, that plays the role of a meta-model or a computational *vademecum*, to be used online for real-time simulation, optimization, inverse

analysis and simulation-based control [22]. Some recent reviews concerning the PGD can be found in [20, 21, 24], along with the recently published primer [23].

1.1 Proper Generalized Decomposition

Most of the existing model reduction techniques proceed by projecting the problem solution onto a reduced basis (this constitutes the wide class of *projection-based model order reduction* methods [10]). Therefore, the construction of the reduced basis usually constitutes the first step in the solution procedure, giving rise to a second important distinction when classifying MOR techniques: *a posteriori* versus *a priori* MOR [69]. One must be careful on the suitability of a particular reduced basis when employed for representing the solution of a particular problem, particularly if it was obtained through *snapshots* of slightly different problems. This difficulty (at least partially) disappears if the reduced basis is constructed at the same time that the problem is solved (in other words: a priori with no need for snapshots of different problems). Thus, each problem has its associated basis in which its solution is expressed. One could consider few vectors in the basis, leading to a reduced representation, or all the terms needed for approximating the solution up to a certain accuracy level. The Proper Generalized Decomposition (PGD), which is described in general terms in the next section, proceeds in this manner.

When calculating the transient solution of a generic problem, say $u(x, t)$, we usually consider a given basis of space functions $N_i(x)$, $i = 1, \ldots, N$, the so-called shape functions within the finite element framework. They approximate the problem solution as

$$u(x,t) \approx \sum_{i=1}^{N} a_i(t) N_i(x).$$

This implies a space-time separated representation where the time-dependent coefficients $a_i(t)$ are unknown at each time instant (when proceeding incrementally) and the space functions $N_i(x)$ are given "a priori", e.g., piece-wise polynomials. POD and Reduced Basis methodologies consider a set of global, reduced basis $\phi_i(x)$ for approximating the solution instead of the generic, but local, finite element functions $N_i(x)$. The former are expected to be more adequate to approximate the problem at hand. Thus, it results

$$u(x,t) \approx \sum_{i=1}^{R} b_i(t) \phi_i(x), \tag{1}$$

where it is expected that $R \ll N$. Again, Eq. (1) represents a space-time separated representation where the time-dependent coefficient must be calculated at each time instant during the incremental solution procedure.

Inspired from these results, one could consider the general space-time separated representation

$$u(x,t) \approx \sum_{i=1}^{N} X_i(x) \cdot T_i(t), \tag{2}$$

where now neither the time-dependent functions $T_i(t)$ nor the space functions $X_i(x)$ are *a priori* known. Both will be computed on the fly when solving the problem.

As soon as one postulates that the solution of a transient problem can be expressed in the separated form (2), whose approximation functions $X_i(x)$ and $T_i(t)$ will be determined during the problem solution, one could make a step forward and assume that the solution of a multidimensional problem $u(x_1, \ldots, x_d)$ could be found in the separated form

$$u(x_1, x_2, \ldots, x_d) \approx \sum_{i=1}^{N} X_i^1(x_1) \cdot X_i^2(x_1) \cdot \ldots \cdot X_i^d(x_d),$$

and even more, expressing the 3D solution $u(x, y, z)$ as a finite sum decomposition involving low-dimensional functions

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

or

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

Equivalently, the solution of a parametric problem $u(\boldsymbol{x}, t, p_1, \ldots, p_\wp)$ could be approximated as

$$u(\boldsymbol{x}, t, p_1, \ldots, p_\wp) \approx \sum_{i=1}^{N} X_i(\boldsymbol{x}) \cdot T_i(t) \cdot \prod_{k=1}^{\wp} P_i^k(p_k).$$

The performance of all these separated representations is excelent in many cases, leading to important time savings. However, the key point when considering such a separated representation lies in the algorithm to be used for calculating the involved functions: $T_i(t)$, $X_i(x)$, $P_i(p)$. Both questions will be addressed in this section.

This kind of parametric modeling has been deeply studied in a panoply of applications, where material and/or process parameters [1, 2, 5, 14, 35, 39, 60, 74], initial conditions [29, 31], boundary conditions [26, 27, 30, 55], different scales [6, 19, 33] and parameters defining the geometry [7] were considered extra-coordinates within the PGD framework. All these parametric solutions were successfully employed for performing real time simulations (e.g. surgical simulation involving haptic devices involving contact, cutting, etc.) [52], material homogenization [39],

real-time process optimization [26, 27], inverse analysis and simulation-based control [29]. They where also employed in dynamic data driven application systems.

1.2 Dimensionality Reduction

In the framework just described, model parameters are explicitly defined. Initial conditions, boundary conditions, material or process parameters or some geometrical parameter defining the domain in which the model is defined can easily be considered as parameters under this framework. In all the treated cases these parameters were explicitly given and the only difficulty was to transfer all them into the extended weak form of the problem before applying the PGD rationale to construct the parametric separated representation [22, 23].

The difficulty appears as soon as the model contains some hidden parameters, that are not explicitly known. In that case, these parameters must be previously identified and extracted and then introduced into the model before computing its parametric solution.

This situation is found in many engineering applications. In this paper we address two of them. The first concerns the parametrization of microstructures consisting of inclusions into a matrix phase (a situation encountered in the analysis of composite materials, for instance). The second concerns patient-specific biomechanics modeling for surgery simulation and planning, in which the solution must encompass both parametric loading and organ shape (patient-specific anatomy). In essence, this situation arises whenever *shape* itself is a parameter of the model. How to parametrize the *shape* of the domain with a minimal number of degrees of freedom is thus a question of utmost interest.

In this framework loads are easily parametrized because their intensity and the region in which they apply can be easily defined [55]. However, parameters defining the organ shape (anatomy) are not explicitly available. No CAD description or similar is available for organs and consequently an extractor of the parameters defining the organ shape is compulsory. These questions were addressed in some of our former works [32, 45] in which some preliminary answers were proposed based on the use of Locally Linear Embedding (LLE) manifold learning techniques.

POD, that is equivalent to PCA—Principal Components Analysis—, can be viewed as an information extractor from a raw data set that attempts to find a linear subspace of lower dimensionality than the original space. If the data has more complicated structures which cannot be well represented in a linear subspace, standard PCA will not be very helpful, leading to too many vectors in the base. Fortunately, kernel PCA allows us to generalize standard PCA to nonlinear dimensionality reduction [71, 72, 76]. Locally Linear Embedding (LLE) [63] results from a particular choice of the kernel within the kPCA framework [77].

In [45] LLE was considered for performing suitable interpolations on the data manifold available from offline information. Thus, homogenized properties in heterogeneous microstructures were inferred in real-time and with a minimum amount of calculation. In [32], again within the LLE framework, parametric solutions related to organ deformation for parametrized loads were interpolated on the manifold defined by organ shapes to create patient-specific surgery tools.

In the present work we move a step forward. More than extracting a manifold for interpolating on it model solutions (or parametric solutions), we propose to compute parametric solutions by properly integrating on the manifold defined by all possible domain geometries.

2 From PCA to kPCA

2.1 Principal Component Analysis—PCA—

Let us consider D observed variables defining the vector $\boldsymbol{y} \in \mathbb{R}^D$. These are commonly referred to in the MOR literature as the *snapshots* of the system: nodal values of the essential field of the mode throughout time in usual finite element modeling, or parameter values at these nodal locations, for instance. We assume that these variables are therefore not uncorrelated and, notably, that there exists a linear transformation $\boldsymbol{W}$ defining the vector $\boldsymbol{v} \in \mathbb{R}^d$, where $d < D$ represents the unknown so-called *latent variables*, according to

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{v}. \tag{3}$$

The transformation $\boldsymbol{W}$, $D \times d$, is assumed to verify the orthogonality condition $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_d$, where $\boldsymbol{I}_d$ represents the $d \times d$-identity matrix ($\boldsymbol{W}\boldsymbol{W}^T$ is not necessarily $\boldsymbol{I}_D$). The existence of such a transformation is precisely at the origin of PCA methods.

We assume the existence of M different snapshots $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M$, that can be stored in the columns of the $D \times M$ matrix $\boldsymbol{Y}$. The associated $d \times M$ reduced matrix $\boldsymbol{\Upsilon}$ contains the associated vectors $\boldsymbol{v}_i$, $i = 1, \ldots, M$.

We assume that both observed and latent variables are centered, that is $\sum_{i=1}^{M} \boldsymbol{y}_i = \boldsymbol{0}$ and $\sum_{i=1}^{M} \boldsymbol{v}_i = \boldsymbol{0}$. If it is not the case, prior to proceed, observed variables must be centered by removing the expectation of $\mathrm{E}\{\boldsymbol{y}\}$ to each observation $\boldsymbol{y}_i$, $i = 1, \ldots, M$. Since the exact expectation is unknown, one commonly accepted procedure is to substitute it by the sample mean.

PCA is able to calculate both d—the necessary number of members in the basis of the reduced-order subspace—and the transformation matrix $\boldsymbol{W}$. PCA proceeds by guaranteeing maximal preserved variance and decorrelation in the latent variable set $\boldsymbol{\upsilon}$. From a statistical point of view, therefore, it can be assumed that the latent variables in $\boldsymbol{\upsilon}$ are uncorrelated (no linear dependencies among them) or mutually orthogonal, thus constituting a basis. In practice, this means that the covariance matrix of $\boldsymbol{\upsilon}$, defined as

$$\boldsymbol{C}_{\upsilon\upsilon} = \mathrm{E}\{\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T\}, \tag{4}$$

for centered $\boldsymbol{y}$ data, is diagonal.

However, the observed variables are expected to be correlated. The goal of PCA is then to extract the d uncorrelated latent variables in $\boldsymbol{\upsilon}$, according to

$$\begin{aligned}\boldsymbol{C}_{yy} &= \mathrm{E}\{\boldsymbol{Y}\boldsymbol{Y}^T\} = \mathrm{E}\{\boldsymbol{W}\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T\boldsymbol{W}^T\} \\ &= \boldsymbol{W}\mathrm{E}\{\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T\}\boldsymbol{W}^T = \boldsymbol{W}\boldsymbol{C}_{\upsilon\upsilon}\boldsymbol{W}^T,\end{aligned}$$

that by pre-multiplying and post-multiplying by $\boldsymbol{W}^T$ and $\boldsymbol{W}$ respectively, and taking into account that $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$, leads to:

$$\boldsymbol{C}_{\upsilon\upsilon} = \boldsymbol{W}^T\boldsymbol{C}_{yy}\boldsymbol{W}. \tag{5}$$

The covariance matrix $\boldsymbol{C}_{yy}$ can then be factorized by applying the singular value decomposition,

$$\boldsymbol{C}_{yy} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T, \tag{6}$$

with $\boldsymbol{V}$ containing the orthonormal eigenvectors and $\boldsymbol{\Lambda}$ the diagonal matrix containing the eigenvalues (non-negative real numbers), assumed in descending order.

Substituting the factorized expression of the covariance matrix (6) into Eq. (5) it results

$$\boldsymbol{C}_{\upsilon\upsilon} = \boldsymbol{W}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{W}.$$

This equality holds only when the d columns of $\boldsymbol{W}$ are taken collinear with d columns of $\boldsymbol{V}$. If the PCA model is fully respected, then only the first d eigenvalues in $\boldsymbol{\Lambda}$ are strictly larger than zero; the other ones are zero.

The eigenvectors associated with these d nonzero eigenvalues must be kept:

$$\boldsymbol{W} = \boldsymbol{V}\boldsymbol{I}_{D\times d},$$

yielding

$$\boldsymbol{C}_{\upsilon\upsilon} = \boldsymbol{I}_{d\times D}\boldsymbol{\Lambda}\boldsymbol{I}_{D\times d}.$$

This shows that the eigenvalues in $\boldsymbol{\Lambda}$ correspond to the variances of the latent variables (the diagonal entries of $\boldsymbol{C}_{\upsilon\upsilon}$).

In real situations, some noise may corrupt the observed variables. As a consequence, all eigenvalues of $\boldsymbol{C}_{\upsilon\upsilon}$ are larger than zero, and the choice of d columns in $\boldsymbol{V}$ becomes more difficult. Assuming that the latent variables have larger variances than the noise, it suffices to choose the eigenvectors associated with the largest eigenvalues. This is the common practice in finite element model order reduction procedures. A number of columns of $\boldsymbol{V}$ are kept so as to preserve a chosen amount of the energy of the system.

From a geometrical point of view, the columns of $\boldsymbol{V}$ indicate the directions in $\mathbb{R}^D$ that span the subspace of the latent variables $\boldsymbol{\upsilon}$. The name PCA then arises naturally from the fact of keeping the *components*—columns—associated with the largest variance.

PCA constitutes a polyvalent method, developed, discovered and re-discovered many times in different branches of applied science and engineering [36, 44, 46]. It determines data dimensionality, builds an embedding accordingly, and extracts the latent variables. However, PCA is still based upon one critical assumption: the linear dependency expressed by Eq. (3) between observed and latent variables (in other words, between the reduced-order and full-order models). It has been observed, however, that very often this is not the case. Frequently, latent variables posses a manifold structure, and therefore it simply does not exist a basis able to construct a projection such as that in Eq. (3). This is the case, for instance, in non-linear, large strain solid dynamics, where a *slow manifold* can be found in which the displacement of the solid evolves [53].

Nonlinear methods are often more powerful than linear ones, because the connection between the latent variables and the observed ones may be much richer than a simple matrix multiplication.

Next section extends linear PCA to nonlinear dimensionality reduction, and describes the so-called kernel Principal Component Analysis—kPCA.

2.2 Kernel Principal Component Analysis (kPCA)

PCA works with the sample covariance matrix, $\boldsymbol{Y}\boldsymbol{Y}^T$. On the contrary, kPCA works with the matrix of pairwise scalar products that defines the Gram matrix $\boldsymbol{S} = \boldsymbol{Y}^T\boldsymbol{Y}$ as it is also the case of Multidimensional Scaling (MDS) methods, also known as method of snapshots [43].

Multidimensional scaling methods construct a configuration of points in a target metric space from information about point distances. Among the most basic non-linear dimensionality reduction method classification (that of distance- or neighbourhood-preserving methods), MDS falls within the first. In its classical version, MDS preserves pairwise scalar products instead of pairwise distances (both are closely related). Moreover, classical metric MDS

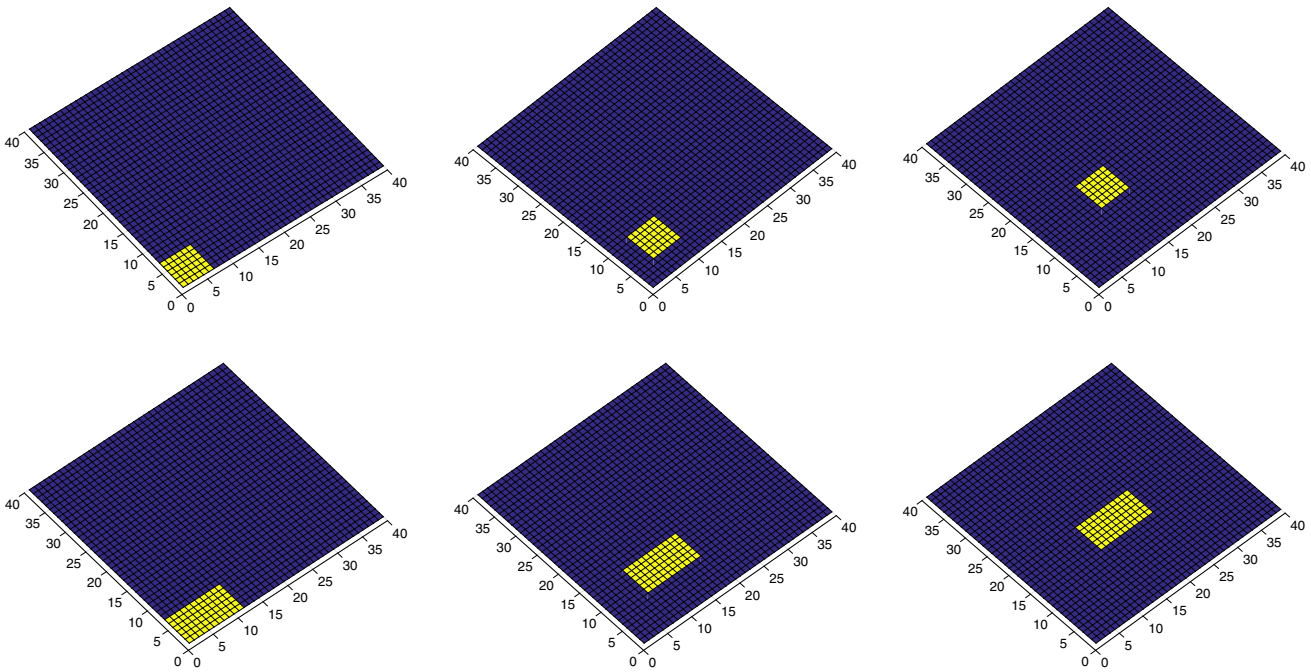

Fig. 1 Different positions of inclusions ω_s (*top*) and ω_r (*bottom*)

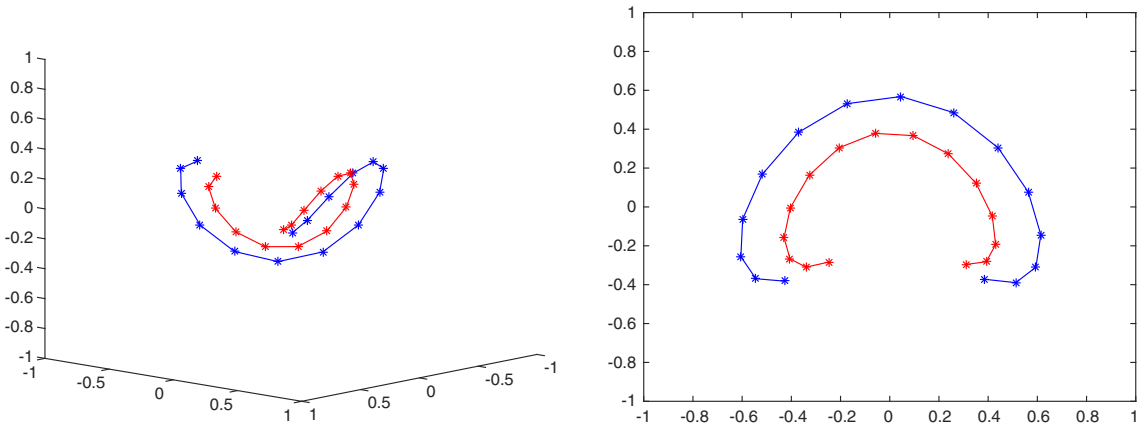

Fig. 2 Reduced data embedded in a 3D (*left*) and 2D (*right*) spaces

cannot achieve dimensionality reduction in a nonlinear way. MDS proceeds from

$$\boldsymbol{S} = \boldsymbol{Y}^T\boldsymbol{Y} = \boldsymbol{\Upsilon}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^T\boldsymbol{\Upsilon},$$

whose eigenvalue decomposition results

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \left(\boldsymbol{U}\boldsymbol{\Lambda}^{1/2}\right)\left(\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}^T\right) = \left(\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}^T\right)^T\left(\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}^T\right),$$

from which it results

$$\boldsymbol{\Upsilon} = \boldsymbol{I}_{d\times M}\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}^T,$$

being easy to prove the equivalence between MDS and PCA [43].

The idea behind kernel-PCA methods is simple, yet appealing: data not linearly separable in D dimensions, could be linearly separated if previously projected to a space in $Q > D$ dimensions. Thus, surprisingly, kPCA begins by projecting the data to an even higher dimensional space. In other words, it proceeds by linearizing the underlying manifold $\mathcal{M}$. To this end, a mapping

$$\phi: \ \mathcal{M} \subset \mathbb{R}^D \rightarrow \mathbb{R}^Q, \ \boldsymbol{y} \rightarrow \boldsymbol{z} = \phi(\boldsymbol{y}),$$

is employed, where Q may be any dimension. One the biggest advantages of this is that there is no need to explicitly determine the analytical expression of the mapping ϕ (it may be even infinite dimensional!).

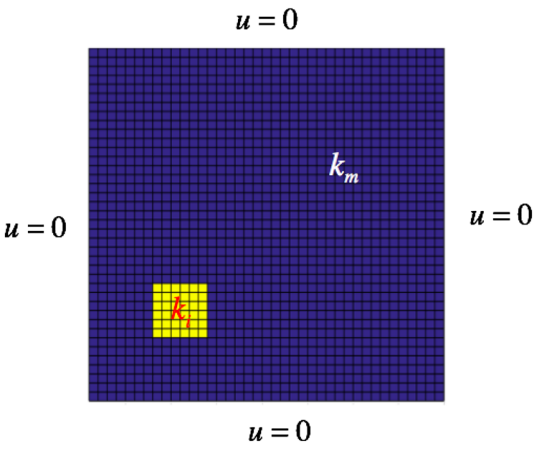

Fig. 3 Thermal problem

The symmetric matrix $\boldsymbol{\Phi} = \boldsymbol{Z}^T\boldsymbol{Z}$ has to be decomposed in eigenvalues and eigenvectors. However, the mapped data z_i involved in $\boldsymbol{\Phi}$ must be previously centered. It is difficult to center it because the mapping is unknown. Fortunately, centering can be achieved in an implicit way by performing the double centering.

The mean of the j-th column of $\boldsymbol{\Phi}$ reads $\mu_i(z_i \cdot z_j)$, and the mean of its i-th row reads $\mu_j(z_i \cdot z_j)$. The mean of all entries of $\boldsymbol{\Phi}$ reads $\mu_{i,j}(z_i \cdot z_j)$. The double centering results from

$$z_i \cdot z_j - \mu_i(z_i \cdot z_j) - \mu_j(z_i \cdot z_j) + \mu_{i,j}(z_i \cdot z_j).$$

Now, the eigenvalue-eigenvector decomposition can be performed on the double centered matrix, according to

$$\boldsymbol{\Phi} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T,$$

from which it results

$$\boldsymbol{\Upsilon} = \boldsymbol{I}_{d\times M}\boldsymbol{\Lambda}^{1/2}\boldsymbol{U}^T.$$

It is worth noting that the mapping ϕ is used solely in scalar products. This may result in a prohibitive computational cost if the mapping is performed onto a space of a high number of dimensions, Q. However, it is possible to simply avoid this difficulty and even ϕ may stay unknown if a kernel function κ is found that directly gives the value of the scalar product $\kappa(\boldsymbol{y}_i, \boldsymbol{y}_j) = z_i \cdot z_j$. This property follows from Mercer's theorem that establishes that if $\kappa(\boldsymbol{u}, \boldsymbol{v})$ is continuous, symmetric and positive definite, then it defines an inner-product in the mapped space.

There exist many different kernels fulfilling Mercer's condition, also known as the "kernel trick". Among them:

- Polynomial kernels: $\kappa(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u} \cdot \boldsymbol{v} + 1)^p$, with p an arbitrary integer;
- Gaussian kernels: $\kappa(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2\sigma^2}\right)$ for a real σ;

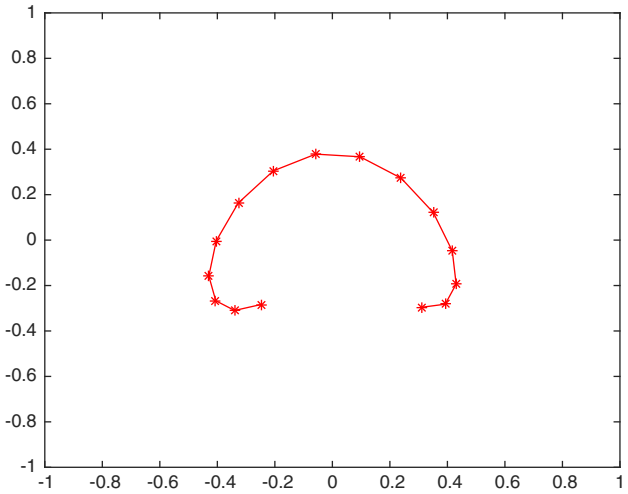

Fig. 4 One-dimensional manifold related to the square inclusion moving along the domain diagonal

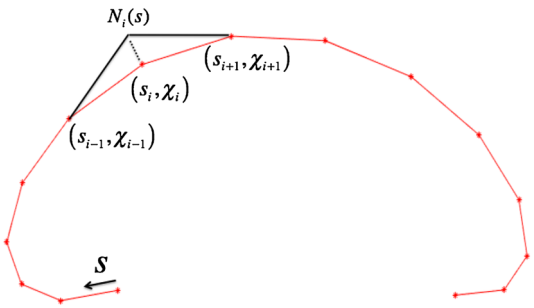

Fig. 5 One-dimensional manifold parametrization

- Sigmoid kernels: $\kappa(\boldsymbol{u}, \boldsymbol{v}) = \tanh(\boldsymbol{u} \cdot \boldsymbol{v} + b)$ for a real b.

The choice of a specific kernel is quite arbitrary and mainly motivated by the hope that the induced mapping ϕ linearizes the manifold to be embedded. If this goal is reached, then PCA applied to the mapped data set should efficiently reveal the nonlinear principal components of the data set.

Remark Other methods proceed by reducing the dimensionality by preserving the topology of data rather than their pairwise distances. Topology preservation seems an appealing route for dimensionality reduction, however, they are in principle more difficult to implement. There exist two variants, the ones that proceed on a predefined topology and the more recent in which the topology is also extracted from the data [43]. Locally Linear Embedding (LLE) is a member of the vast family of techniques. In opposition to most of techniques preserving topology by keeping neighboring points close to each other, LLE is based on conformal mappings. A conformal mapping that represents a transformation that preserves local angles. The preservation of local angles and local distances can be

interpreted as two different ways to preserve local scalar products. This dimensionality reduction technique succeeded for defining robust interpolations and has been employed in some of the author's previous works [32, 45, 63].

2.3 kPCA Dimensionality Reduction from A Numerical Example

To show how kPCA works, consider an idealized composite microstructure. It is defined in a squared domain Ω composed of $D = 40 \times 40$ cells that contains an inclusion. We allow this inclusion to move along one of the diagonals of Ω. First we consider a squared inclusion $\omega_s \subset \Omega$, covering 5×5 cells of Ω. Then, we consider a similar scenario but now consisting in a 10×5-cell rectangular inclusion $\omega_r \subset \Omega$. Both cases are depicted in Fig. 1.

For each microstructure defined by a particular position of the inclusions ω_s^k and ω_r^k, $k = 1, \ldots, M = 15$, within Ω, we define the phase field $p(i, j)$ associated to each cell $\mathcal{C}(i, j) \subset \Omega$:

$$p(i,j) = \begin{cases} 1 & \text{if } \mathcal{C}(\mathrm{i},\mathrm{j}) \subset \omega, \\ 0 & \text{if } \mathcal{C}(\mathrm{i},\mathrm{j}) \subset \Omega - \omega. \end{cases}$$

The phase field of each microstructure can be expressed as a vector with binary entries, $\boldsymbol{p} \in \mathbb{R}^D$. Each constitutes a column of matrix $\boldsymbol{P}$. A Gaussian kernel with $\sigma = 10$ is employed for dimensionality reduction. One of the drawbacks of non-linear dimensional reduction is the need for user-defined parameters. Typically, these include the number of dimensions of the embedding space. The corresponding images $\boldsymbol{\pi}$ related to data in $\boldsymbol{P}$ in a three- and two-dimensional embedding spaces are depicted in Fig. 2. It can be noticed that, as expected, data belong to a manifold of dimension one (more complex situations will be addressed later), and moreover that kPCA succeeded to separate both kind of microstructures, the ones composed of a square inclusion from the ones related to the rectangular inclusion.

3 Combining kPCA Dimensionality Reduction and PGD-Based Parametric Solutions

We consider in this example a problem with a parametric dependence. The difficulty comes from the fact that this parametric dependence is not explicit. We do not know even what the parameters are. This type of problems arises naturally in the characterization and ulterior numerical simulation of composite materials, for instance.

We consider the thermal problem illustrated in Fig. 3 that involves the temperature field $u(\boldsymbol{x})$, with $\boldsymbol{x} \in \Omega = [0, 1]^2$, for any position of the inclusion $\omega \subset \Omega$ along the domain diagonal and for any conductivity contrast $\alpha = \frac{k_i}{k_m}$ (inclusion to matrix conductivity ratio), with $\alpha \in \mathcal{I} = [\alpha_{min} = 1, \alpha_{max} = 10]$. The steady-state heat

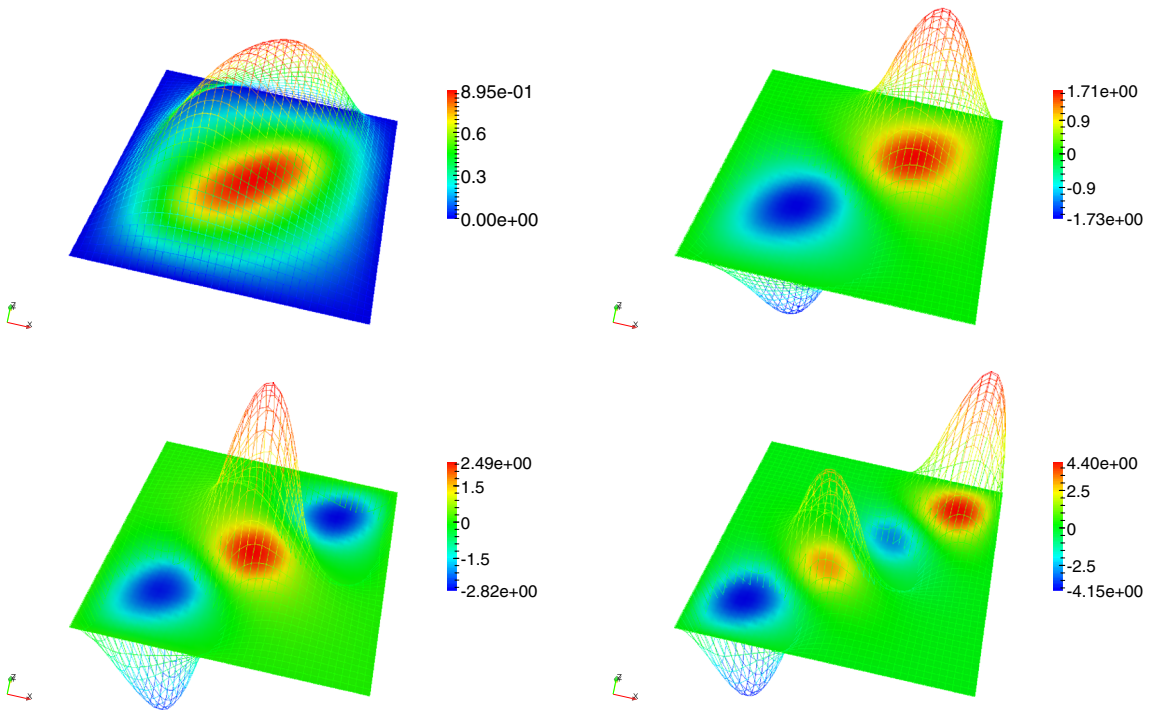

Fig. 6 Space modes: $T_1(\mathbf{x})$ (*top-left*); $T_2(\mathbf{x})$ (*top-right*); $T_3(\mathbf{x})$ (*bottom-left*) and $T_4(\mathbf{x})$ (*bottom-right*)

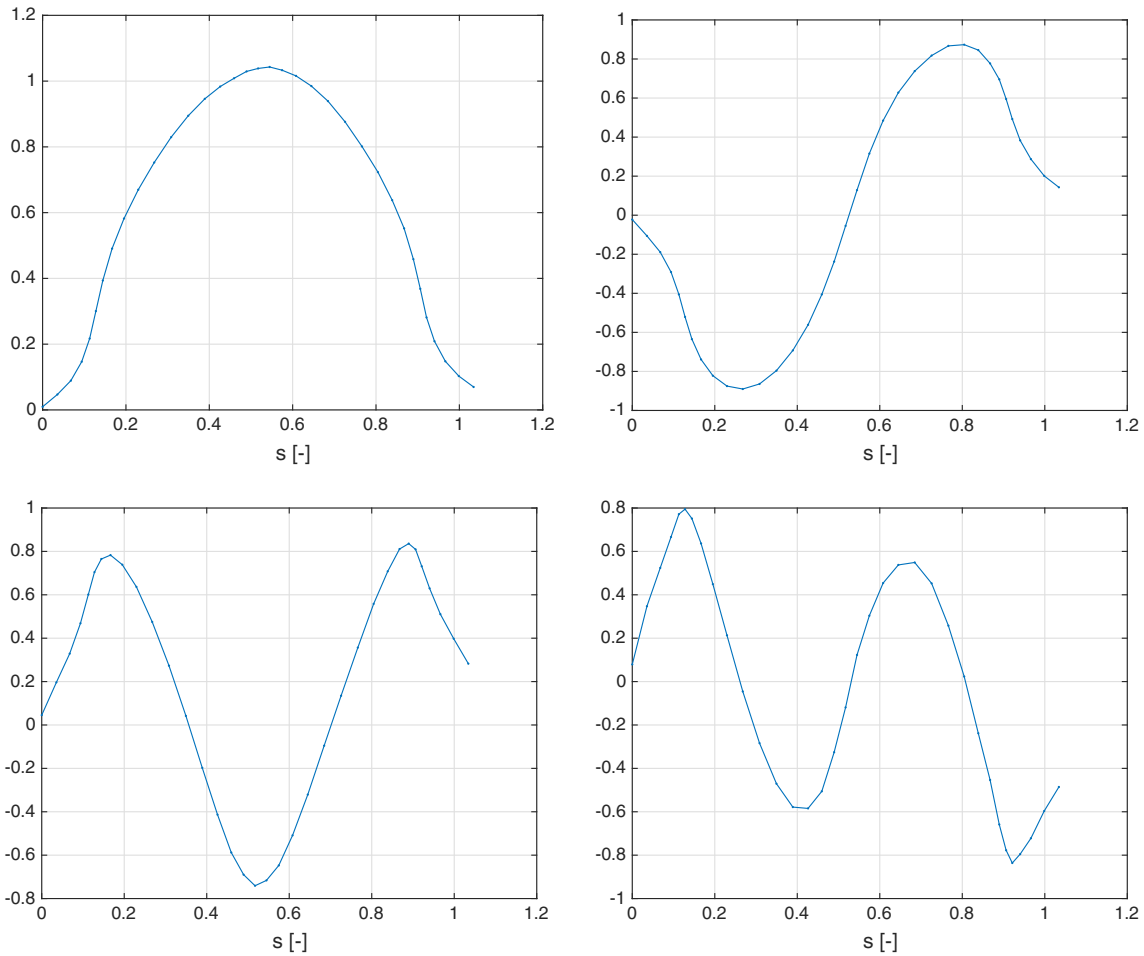

Fig. 7 Modes related to the curvilinear coordinate s: $S_1(s)$ (*top-left*); $S_2(s)$ (*top-right*); $S_3(s)$ (*bottom-left*) and $S_4(s)$ (*bottom-right*)

transfer problem, with both conductivities (of matrix and inclusion phases) assumed homogeneous and isotropic, reads

$$\nabla \cdot (k(\boldsymbol{x})\nabla u(\boldsymbol{x})) = \mathcal{Q}(\boldsymbol{x}), \quad \text{in } \Omega,$$

subjected to the prescribed homogeneous boundary conditions, $u(\boldsymbol{x} \in \partial\Omega) = 0$. The source term $\mathcal{Q}(\boldsymbol{x})$ has a unit value inside the inclusion and vanishes elsewhere.

Since we are interested in calculating the parametric solution for any position of the inclusion (along the domain diagonal) and for any conductivity contrast $\alpha \in \mathcal{I}$, the first step consists in introducing both parameters explicitly into the problem model.

For that purpose, we consider the one-dimensional manifold associated to the inclusion location depicted in Fig. 4, that can be parametrized as depicted in Fig. 5. Each point in the manifold is related to one of the $M = 15$ considered snapshots (positions of ω on the domain diagonal). The manifold is defined by the polygonal joining the different snapshots, even if smoother reconstructions can be defined by using splines, for instance. The manifold is parametrized by the curvilinear coordinate s and each vertex is defined by a coordinate s_i and the associated phase field $\boldsymbol{p}_i$.

When considering a particular position s, the phase field can be approximated using the simplest interpolation schema, the piece-wise linear interpolation defined from

$$\boldsymbol{p}(s) = \sum_{i=1}^{M} N_i(s)\boldsymbol{p}_i, \tag{7}$$

that for $2 \leq i \leq M-1$,

$$N_i(s) = \begin{cases} \dfrac{s - s_{i-1}}{s_i - s_{i-1}} & s_{i-1} \leq s \leq s_i \\ \dfrac{s_{i+1} - s}{s_{i+1} - s_i} & s_i \leq s \leq s_{i+1} \end{cases},$$

where $N_1(s_1 \leq s \leq s_2) = \frac{s_2 - s}{s_2 - s_1}$ and $N_M(s_{M-1} \leq s \leq s_M) = \frac{s - s_{M-1}}{s_M - s_{M-1}}$.

It is important to notice that Eq. (7) constitutes a separated representation with functions $N_i(s)$ depending on the

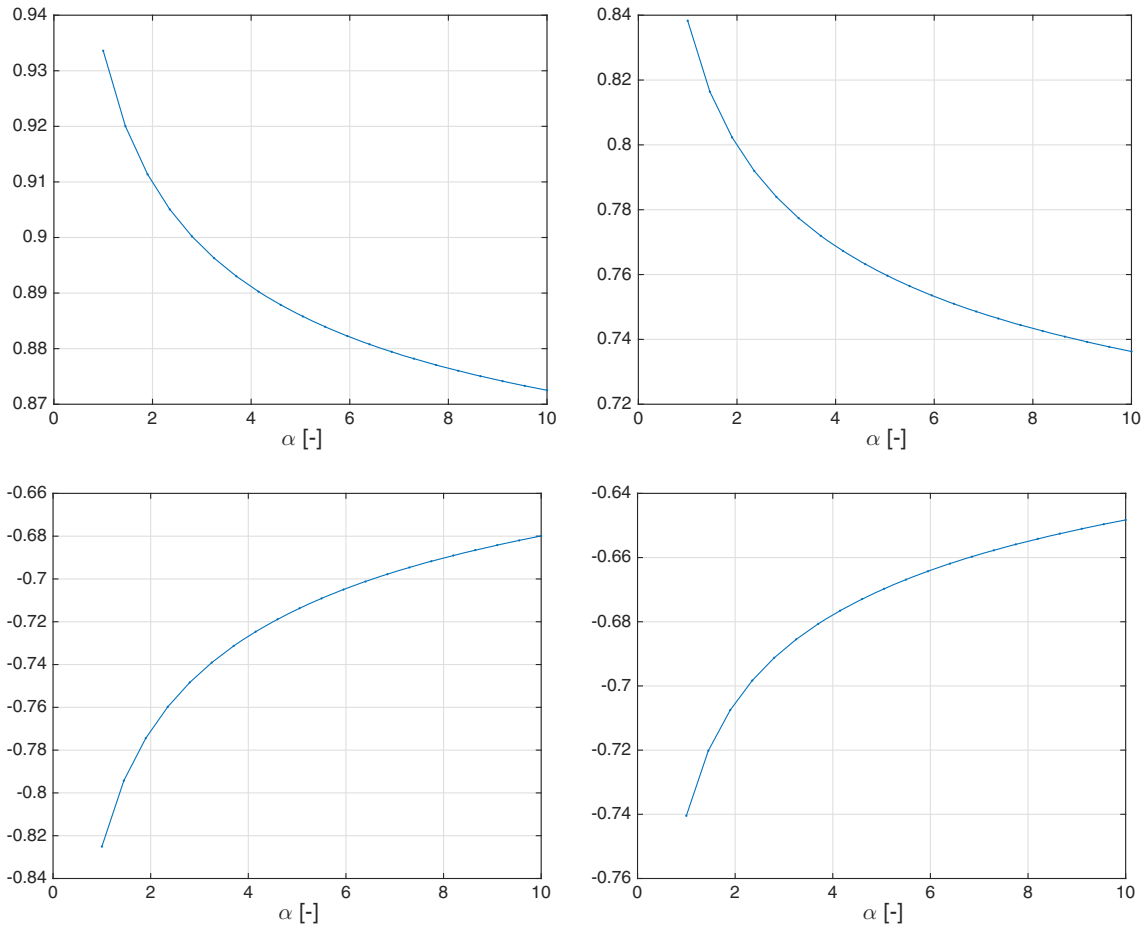

Fig. 8 Modes related to the conductivity contrast α: $A_1(\alpha)$ (*top-left*); $A_2(\alpha)$ (*top-right*); $A_3(\alpha)$ (*bottom-left*) and $A_4(\alpha)$ (*bottom-right*)

coordinate describing the one-dimensional manifold. In turn, phase fields $\boldsymbol{p}_i$ are vectors collecting nodal values of $p(\boldsymbol{x})$ for each considered snapshot. Thus, in a more compact form, it can be written as

$$p(\boldsymbol{x}, s) = \sum_{i=1}^{M} F_i(\boldsymbol{x}) \cdot G_i(s).$$

Within this rationale, the conductivity parametrization results

$$\begin{aligned} k(\boldsymbol{x}) &= k_m + k_m(\alpha - 1)p(\boldsymbol{x}, s) \\ &= k_m + k_m(\alpha - 1)\sum_{i=1}^{M} F_i(\boldsymbol{x}) \cdot G_i(s). \end{aligned}$$

The parametric temperature field can now be written in the separated form

$$u(\boldsymbol{x}, \alpha, s) \approx \sum_{i=1}^{N} T_i(\boldsymbol{x}) \cdot S_i(s) \cdot A_i(\alpha). \tag{8}$$

In order to construct such a separated representation, we consider the triply-weak form

$$\begin{aligned} &\int_{\Omega}\int_{\mathcal{I}}\int_{s_1}^{s_M} u^*(\boldsymbol{x}, \alpha, s) \cdot \nabla((k_m + k_m(\alpha - 1) \\ &\sum_{i=1}^{M} F_i(\boldsymbol{x}) \cdot G_i(s))\nabla u(\boldsymbol{x}, \alpha, s))\, d\boldsymbol{x}\, d\alpha\, ds = 0, \end{aligned}$$

and proceed by calculating iteratively each functional product involved in the separated representation (8). At each iteration a nonlinear problem must be solved, and for that purpose an alternated direction fixed point algorithm is considered. For the implementation details the interested reader can refer to [23] and the numerous references therein.

Figures 6, 7, 8 depict the four more significative modes involved in the separated representation (8), $T_i(\boldsymbol{x})$, $i = 1, \cdots, 4$, $S_i(s)$, $i = 1, \cdots, 4$ and $A_i(\alpha)$, $i = 1, \cdots, 4$ respectively.

Figure 9 depicts the points at which the solution will be particularized, in the left column the manifold is embedded in a 2D dimensional space whereas in the right column these points appears on the one-dimensional manifold now embedded in a 3D space.

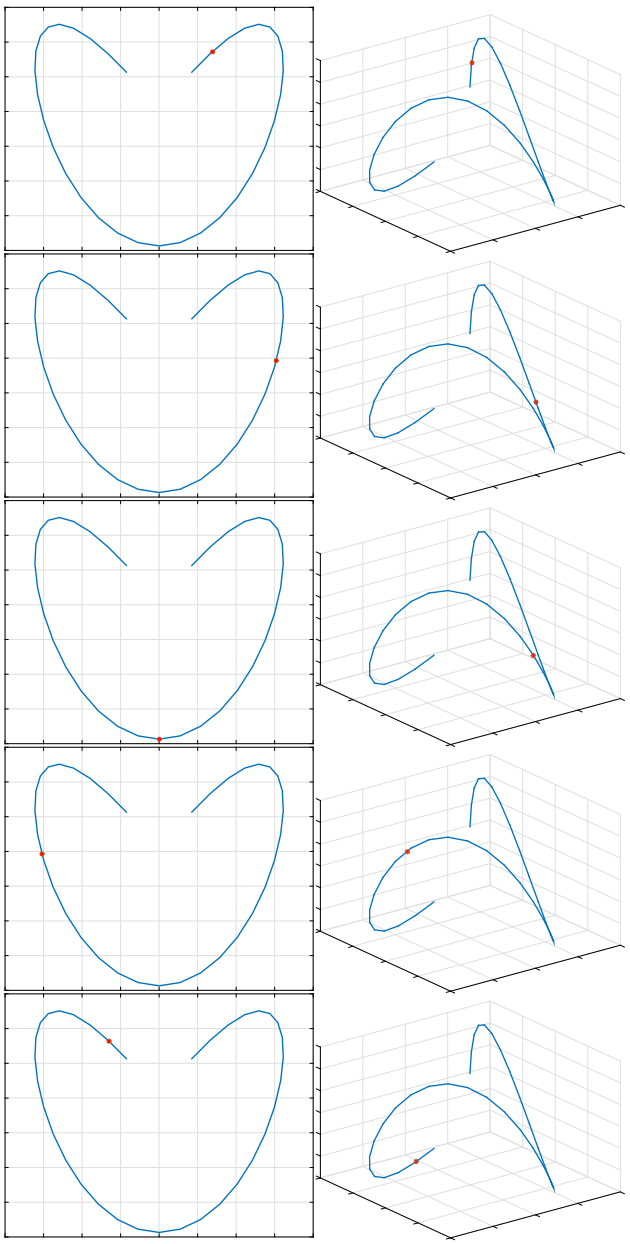

Fig. 9 Points in the 1D manifold embedded in a 2D (*left*) and 3D (*right*) spaces

Finally Figs. 10 and 11 depict respectively the solution at the five positions for $\alpha = 5.5$ (Fig. 10) and the solution at the central point for $\alpha = 1$, $\alpha = 5.5$ and $\alpha = 10$ (Fig. 11).

4 Patient-Specific Computational Liver Vedemecums

Among the problems in which *shape* plays a prominent role, those in the field of biomechanics have utmost importance. What we mean by shape in biomechanics is actually *anatomy*. The problem could thus be formulated in loose terms as: what makes a liver to be a liver (in terms of shape, of course) and therefore be easily recognized by a surgeon? What is the minimal number of parameters that must be employed to properly characterize and identify a human liver? Characterizing the mechanical response of such a parametric liver would make it possible to have the definitive patient-specific model for surgery simulation and/or planning.

One of the most successful approaches to this type of problems within the framework of reduced-order models is that of free-form models [11, 41, 51]. Particularly noteworthy for biomechanics applications is its application to hemodynamics [40, 50, 62]. Essentially, free-form deformations consider the embedding of the model within a cube. By deforming this cube, and assuming an affine deformation of the solid within the cube, a parametrization of the shape of the model is obtained in terms of the nodal discretization of the embedding cube.

4.1 Parametrizing Shapes

Computational vademecums can be used for very different purposes. Real-time simulation for surgery planning and training, for instance, is one of these possible applications [52]. These vademecums represent the response, in the form of a displacement field $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s})$ as a function of the physical point considered, $\boldsymbol{x}$, and the location $\boldsymbol{s}$ of the load provoked by the surgical instrument. In a previous work, the authors employed Locally Linear Embedding techniques to properly interpolate these computational vademecums obtained for different anatomies [32]. Thus, a new patient anatomy was firstly interpolated on the manifold of vademecums to obtain his/her own vademecum.

Here, this approach has been generalized so as to obtain a completely general vademecum in which shape is a parameter itself. Indeed, shape is parameterized by a minimal number of degrees of freedom dictated by the application of kPCA techniques to a set of 75 livers. These organs were obtained by affinely deforming a reference anatomy. More details can be found in [32]. The set of 75 livers is shown in Fig. 12.

Every liver model is then embedded within a mesh composed by $43 \times 31 \times 37$ elements, thus making 49321 nodes. A level set (distance) field is then computed and nodal values stored for each sample in a set of vectors $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{75}\}$, $\boldsymbol{y}_i \in \mathbb{R}^{49321}$, see Fig. 13.

These high-dimensional vectors $\boldsymbol{y}_i$ serve as a precise identification of every anatomy in the sample. However, 49321 values do not constitute an appropriate parametrization of liver geometry, for obvious reasons. These 75 high-dimensional vectors are analyzed by

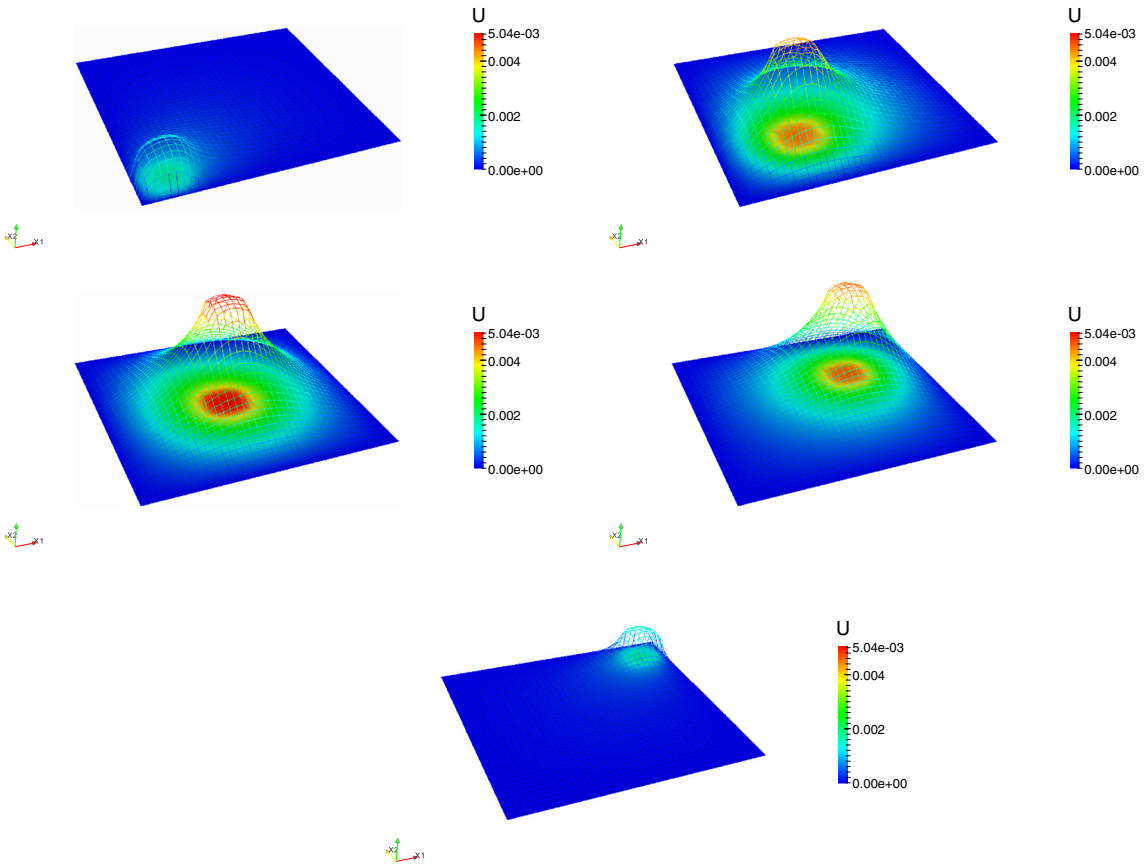

Fig. 10 Reconstructed solution at the five points depicted in Fig. 9 for $\alpha = 5.5$

employing kPCA methods, by employing Gaussian kernels, taking $\sigma = 95 \times 10^3$. Tests done with up to 500 different livers showed that the embedding manifold is actually flat, see Fig. 14. Surprisingly, all the 500 cases lie very accurately in a square domain.

This implies that it is possible to work on a flat space of *shapes*, parameterized by the embedding coordinates $\boldsymbol{\upsilon}_i \in \mathbb{R}^2$, (whatever they mean physically), and that the considered anatomies lie actually within a square. It is therefore possible to mesh the shape space by employing Delaunay triangulations over the set of embedded vectors. Back to the set of 75 livers, the resulting triangulation of the convex hull of shapes is shown in Fig. 15.

Once the shape space has been properly identified and parameterized, it is possible to establish the weak form of the problem. What we call a vademecum [22] is actually a parametric solution for the problem at hand, that is computed off-line once for life, and is then evaluated fast once needed. Therefore, our parametric solution for this problem, or *shape vademecum*, would be $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{\upsilon})$, thus representing the displacement field of a liver, for any load position on its surface $\boldsymbol{s}$ and for any geometry (anatomy) $\boldsymbol{\upsilon} = (\upsilon_1, \upsilon_2) \in \Sigma = \text{conv}(\boldsymbol{\upsilon}_i)$. Here, $\text{conv}(\cdot)$ stands for the convex hull of the set of points.

4.2 Developing the Vademecum

To develop the sought weak for of the parametric problem, we start by the (static, for simplicity) equilibrium equations, namely,

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} \quad \text{in} \quad \Omega, \tag{9}$$

where $\boldsymbol{b}$ represents the volumetric force applied to the body. The domain is subjected to the following boundary conditions

$$\begin{aligned} \boldsymbol{u} &= \bar{\boldsymbol{u}} \quad \text{on} \quad \Gamma_{\text{u}} \\ \boldsymbol{\sigma n} &= \bar{\boldsymbol{t}} \quad \text{on} \quad \Gamma_{\text{t}}. \end{aligned}$$

$\bar{\Gamma} \subseteq \Gamma_t$ represents the portion of the boundary of the organ where the load can be applied (region accesible to the surgeon). After multiplying both sides of Eq. (9) by an admissible variation of the displacement, $\boldsymbol{u}^*$, and integrating over the domain Ω, the standard weak form of the problem is obtained. However, in this case we face a

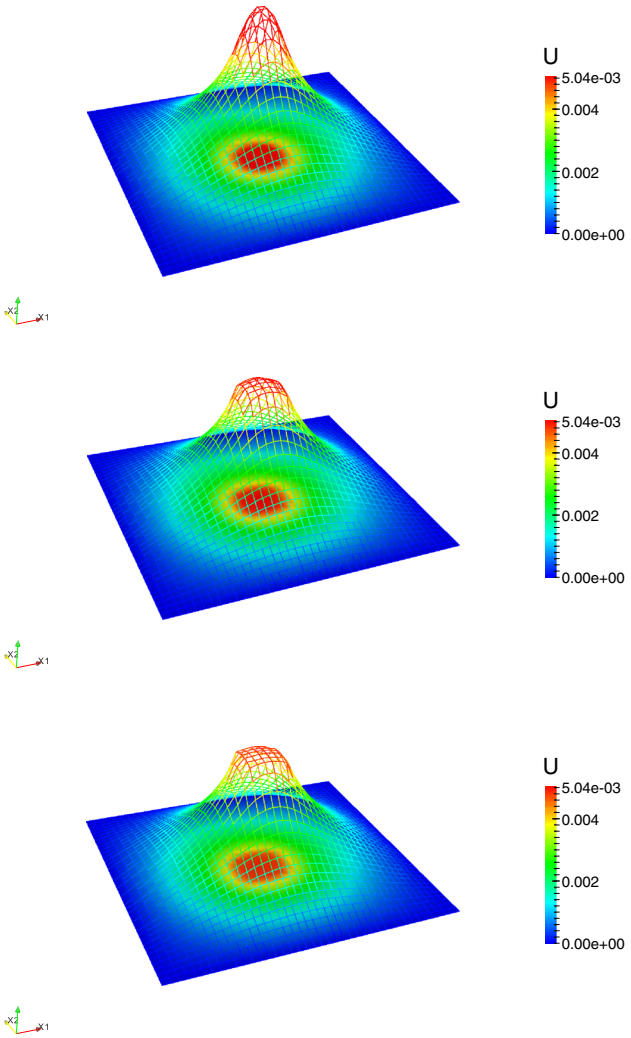

Fig. 11 Reconstructed solution at the central point depicted in Fig. 9 for $\alpha = 1$ (*top*), $\alpha = 5.5$ (*middle*) and $\alpha = 10$ (*bottom*)

parametric problem. In this case, the (triply-)weak form will consist in *finding the displacement* $\boldsymbol{u} \in \mathcal{H}^1(\Omega) \times L^2(\bar{\Gamma}) \times L^2(\Sigma)$ *such that for all* $\boldsymbol{u}^* \in \mathcal{H}^1_0(\Omega) \times L^2(\bar{\Gamma}) \times L^2(\Sigma)$ [54]:

$$\int_\Sigma \int_{\bar{\Gamma}} \int_\Omega (\nabla_s \boldsymbol{u}^*)^T \boldsymbol{\sigma} d\Omega d\bar{\Gamma} = \int_\Sigma \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\boldsymbol{u}^*)^T \boldsymbol{t} d\Gamma d\bar{\Gamma}, \quad (10)$$

where $\nabla_s \boldsymbol{u}$ represents the symmetric part of the gradient of displacements, $\Gamma = \Gamma_u \cup \Gamma_t$ represents the boundary of the solid, divided into essential and natural regions, and where $\Gamma_t = \Gamma_{t1} \cup \Gamma_{t2}$, i.e., regions of homogeneous and non-homogeneous, respectively, natural boundary conditions.

The load $\boldsymbol{t}$ acts on a moving position $\boldsymbol{s}$. It is therefore expressed as $\boldsymbol{t}(\boldsymbol{x}, \boldsymbol{s}) = \boldsymbol{t}\delta(\boldsymbol{x} - \boldsymbol{s})$, where δ represents the Dirac-delta function.This Dirac-delta term should be regularized for computation purposes and approximated by:

$$t_j \approx \sum_{i=1}^{m} f_j^i(\boldsymbol{x}) g_j^i(\boldsymbol{s}),$$

by performing a singular value decomposition of the load, for instance, and truncating the number of terms m according to some error tolerance.

As mentioned before, PGD proceeds in an iterative way, constructing an approximation to the solution composed by a finite sum of separable functions. Let us assume that, at iteration n of this algorithm, convergence has been attained, giving

$$u_j^n(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}) = \sum_{k=1}^{n} F_j^k(\boldsymbol{x}) \cdot G_j^k(\boldsymbol{s}) \cdot H_j^k(\boldsymbol{v}),$$

where the term u_j refers to the j-th component of the displacement vector, $j = 1, 2, 3$ and functions $\boldsymbol{F}^k$, $\boldsymbol{G}^k$ and $\boldsymbol{H}^k$ represent the separated functions used to approximate the unknown field, obtained in previous iterations of the PGD algorithm.

The algorithm now proceeds by looking for an improvement of this approximation in a subsequent iteration. The $(n+1)$-th term will therefore incorporate an unknown functional product:

$$u_j^{n+1}(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}) = u_j^n(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}) + R_j(\boldsymbol{x}) \cdot S_j(\boldsymbol{s}) \cdot T_j(\boldsymbol{v}), \quad (11)$$

where $\boldsymbol{R}(\boldsymbol{x})$, $\boldsymbol{S}(\boldsymbol{s})$ and $\boldsymbol{T}(\boldsymbol{v})$ are the sought functions that improve the approximation.

The admissible variation of the displacement is obtained after straightforward application of the rules of variational calculus,

$$u_j^*(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{v}) = R_j^*(\boldsymbol{x}) \cdot S_j(\boldsymbol{s}) \cdot T_j(\boldsymbol{v}) + R_j(\boldsymbol{x}) \cdot S_j^*(\boldsymbol{s}) \cdot T_j(\boldsymbol{v}) + R_j(\boldsymbol{x}) \cdot S_j(\boldsymbol{s}) \cdot T_j^*(\boldsymbol{v}). \quad (12)$$

At this point several options are at hand so as to determine the new triplet of functions $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$. The most frequently used, due to both its easy of implementation and good convergence properties, in general, is a fixed-point algorithm in which functions $\boldsymbol{R}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ are sought iteratively. For details on this algorithm, we refer the interested reader to any of our previous works in the field [54, 55]. Once Eqs. (11) and (12) have been substituted into the weak form of the problem (10), the matrix form of the problem is obtained. At this point it is worth noting that several approaches have been investigated in the literature to linearize the problem if, as is it very often the case, it presents non-linear constitutive equations. For instance, explicit linearizations are possible [54], as well as the employ of Taylor expansions [55] to avoid the computation of the full-order tangent stiffness problem. The application

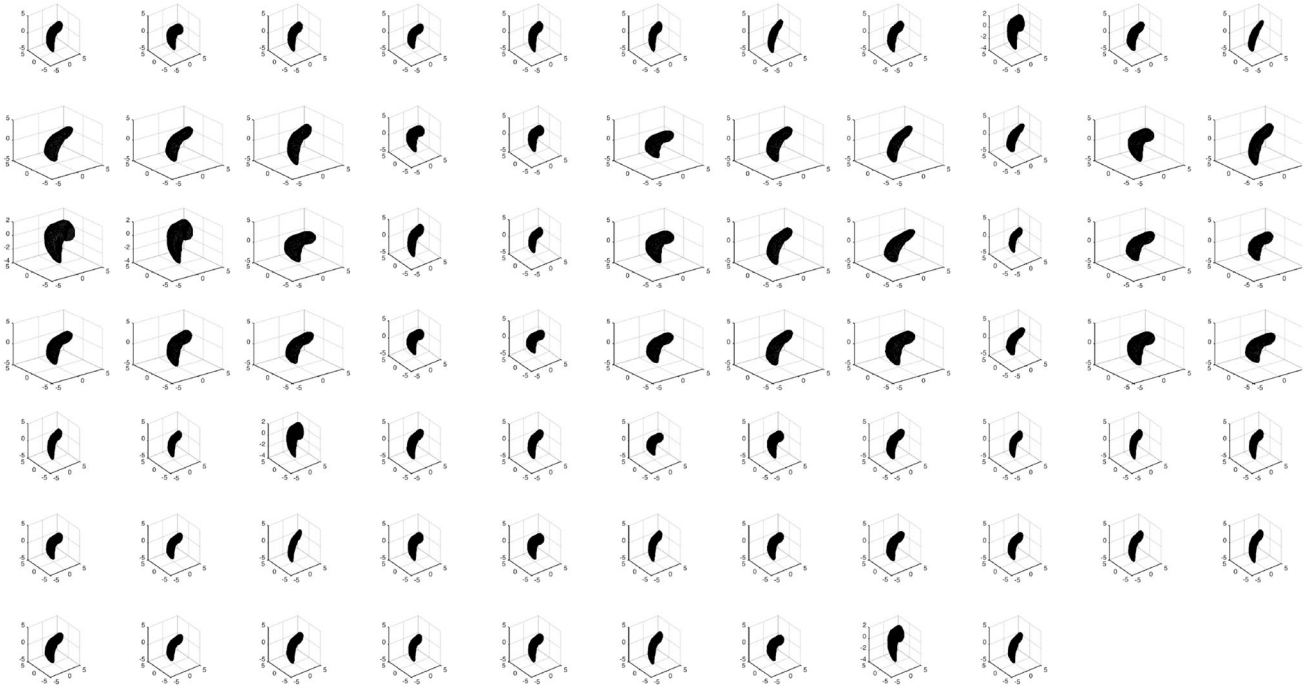

Fig. 12 A group view of the 75 different liver geometries considered for this example

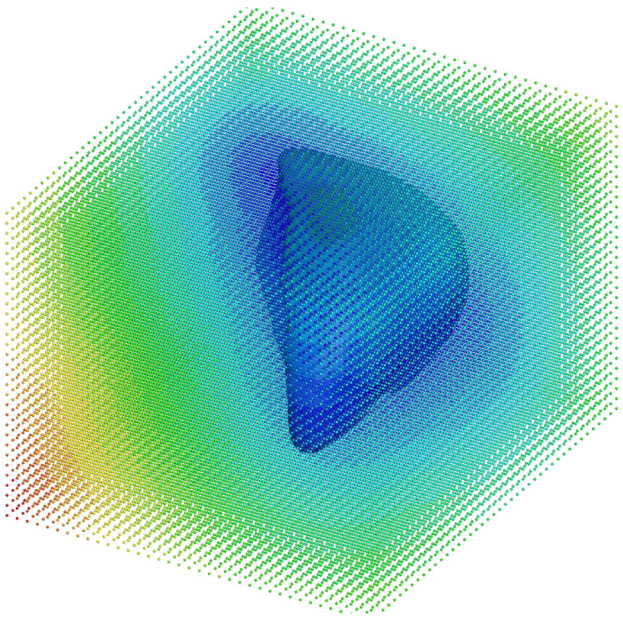

Fig. 13 Distance field computed for one particular instance of the 75 different livers in the sample

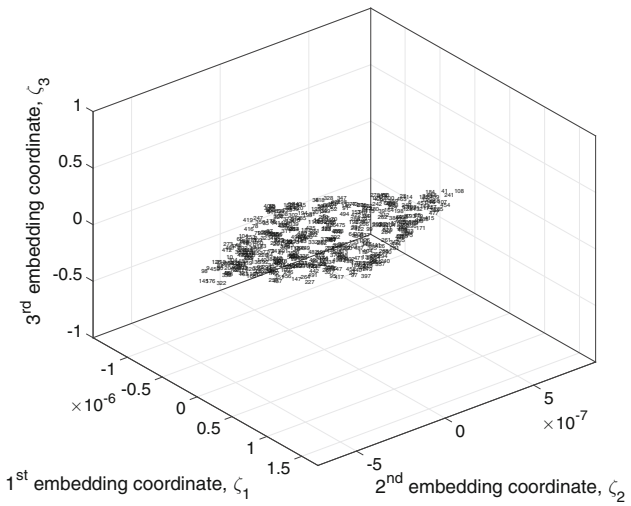

Fig. 14 Embedding of 500 different livers by kPCA techniques. Note the squared, flat, geometry of the resulting embedding, showing that there are two relevant parameters in the set

of well-stablished techniques such as the Empirical Interpolation method [12, 18] is also another possibility. Its application within the framework of PGD methods is deeply analyzed in [23].

Finally, as technical detail, it is worthy of mention that integration on the shape space $\Sigma = \mathrm{conv}(\upsilon_i)$ is done by employing the underlying Delaunay triangles shown in Fig. 15.

4.3 Results

To test the just presented technique, we have taken one of the livers as a reference anatomy and have calculated its own vademecum $\boldsymbol{u}_{\mathrm{ref}} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s})$ following standard PGD methods [54]. Hence, no shape dependence is considered.

This liver has then be eliminated from the set and the above procedure has been applied with the remaining 74 livers. We therefore check the accuracy of the just presented technique in providing an accurate approximation to this reference vademecum.

We have also calculated the distance field (in the form of a high-dimensional vector $\boldsymbol{y}_{\text{ref}}$) for the reference anatomy. When the kPCA algorithm is applied to $\boldsymbol{y}_{\text{ref}}$ vector so as to give $\boldsymbol{\upsilon}_{\text{ref}}$ coordinates in the embedded space, this point appears as the blue diamond in Fig. 16. On the other hand, given the nodal connectivity of the triangle to which $\boldsymbol{y}_{\text{ref}}$ pertains, one could employ standard finite element shape function so as to interpolate $\boldsymbol{y}_{\text{ref}}$ from its three neighbors. The resulting geometry is shown as a red square in the same Fig. 16, showing the accuracy of the kPCA projection onto the embedding space.

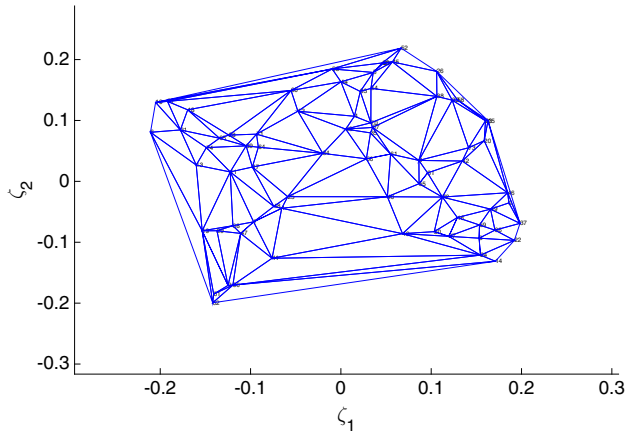

Fig. 15 Delaunay triangulation of the set of 75 livers in the embedding space

But the true interest of this method is to obtain, once particularized, a vademecum for the reference geometry of the form $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{\upsilon}_{\text{ref}})$ and to compare it with $\boldsymbol{u}_{\text{ref}}(\boldsymbol{x}, \boldsymbol{s})$. Both are defined over slightly different domains. In fact, if we compare the distance field generated by both anatomies (the reference one and the interpolated one), the obtained error on L^2-norm is 5.77 %. The error in the predicted displacement field, measured as

$$\|\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}_0, \boldsymbol{\upsilon}_{\text{ref}}) - \boldsymbol{u}_{\text{ref}}(\boldsymbol{x}, \boldsymbol{s}_0)\|_{L^2},$$

for a particular load position $\boldsymbol{s}_0$, was 8.539 %, which is judged enough for this type of applications where the dispersion in mechanical properties of living tissues, for instance, is much more than that. Load positions different to $\boldsymbol{s}_0$ give of course different errors, but of the same order of magnitude.

Figure 17 shows the difference in geometry between the reference geometry and the one computed by the vademecum. As can be noticed, both agree to a reasonable degree of accuracy, not easily distinguishable by the human eye. In Fig. 18 a comparison is made between deformed configurations for one particular load position $\boldsymbol{s}_0$. Again, the accuracy of the approximation is noteworthy.

If compared to our previous approach to the problem, based upon an interpolation of standard vademecums $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s})$ for different geometries, employing weights given by LLE embedding of the high-dimensional vectors $\boldsymbol{z}_i$, the obtained accuracy is of the same order, cf. [32]. The error in the predicted geometry is very similar (about 5 % in L^2-norm), while the error in the predicted displacement field is somewhat less in the present work.

We must also highlight the fact that we have worked with only 75 livers. The algorithm is prepared so as to be fed by a continuous stream of data coming from new

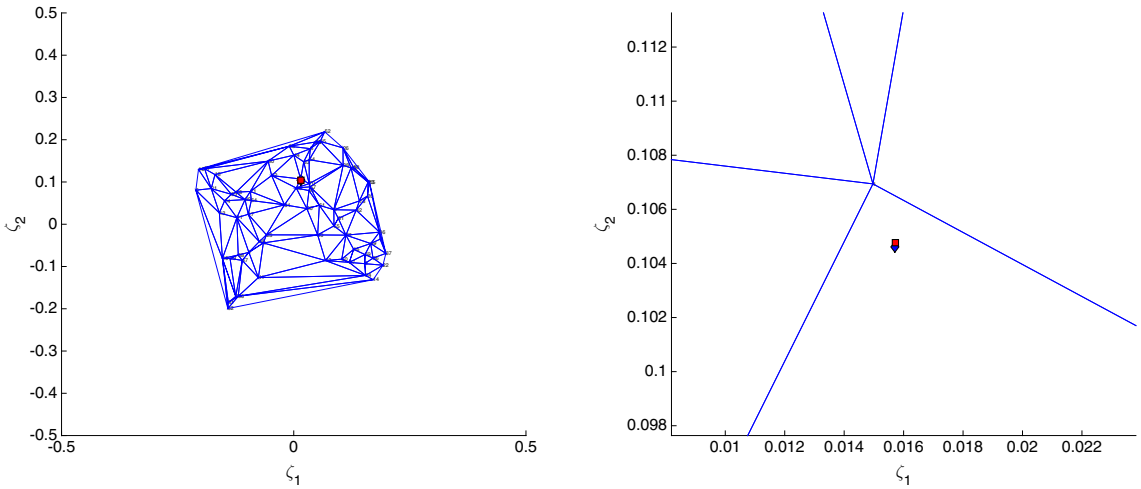

Fig. 16 Approximation of one particular liver anatomy. *Left* position of the reference liver in the embedding space. *Right* detail of the error versus the interpolated anatomy (*blue diamond*)

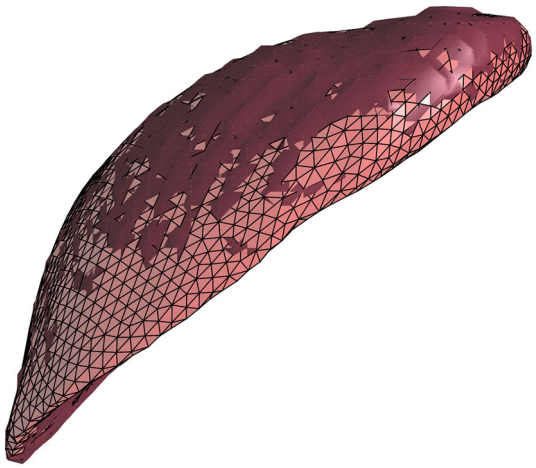

Fig. 17 Difference in geometry between the interpolated and the reference livers. The reference liver is represented in wireframe, while the approximated one is represented in *solid*

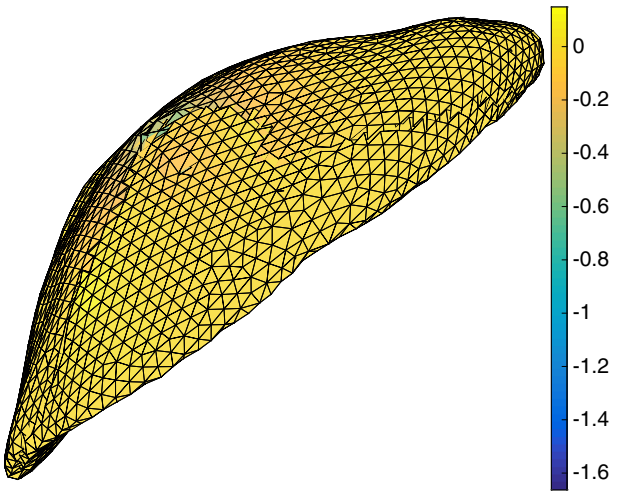

Fig. 18 Difference in the predicted displacement for a particular load position between the interpolated and the reference livers

patients that will very much improve the result by refining the sampling of the shape space.

5 Conclusions

In this work we proved the ability of kPCA to extract the relevant parameters associated to microstructures or shapes. As soon as nonlinear dimensionality reduction applies, a parametric solution using the just extracted parameters can be envisaged within the PGD framework. This PGD approximation is constructed on top of the just found relevant features of the geometric description of the domain. Thus, by combining nonlinear dimensionality reduction and Proper Generalized Decomposition powerful parametric solutions can be constructed, including parameters with full physical meaning and others that where extracted in a transparent way for the user.

The two numerical examples described and discussed prove the extremely high potential of the approaches here proposed. The resulting methods is a sort of mixed a priori/ a posteriori, linear/non-linear model order reduction method. Indeed, while the parametric space is identified a posteriori and non-linearly, the PGD part of the proposed algorithm construct a priori (and linearly) the high-dimensional approach to the parametric solution.

This mixed approach shows great promise in a wide variety of problems. Particularly, those in which shape, in its broadest sense, is one of the parameters of the solution.

Acknowledgments This work has been supported by the Spanish Ministry of Economy and Competitiveness through Grant number CICYT DPI2014-51844-C2-1-R and by the Regional Government of Aragon and the European Social Fund, research group T88. Professor Chinesta is also supported by the Institut Universitaire de France.

References

1. Aghighi MS, Ammar A, Metivier C, Chinesta F (2015) Parametric solution of the Rayleigh-Bénard convection model by using the pgd: application to nanofluids. Int J Numer Methods Heat Fluid Flows 25(6):1252–1281
2. Aguado JV, Huerta A, Chinesta F, Cueto E (2015) Real-time monitoring of thermal processes by reduced order modelling. Int J Numer Meth Eng 102(5):991–1017
3. Ammar A, Mokdad B, Chinesta F, Keunings R (2006) A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. J Non Newton Fluid Mech 139:153–176
4. Ammar A, Mokdad B, Chinesta F, Keunings R (2007) A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. part ii. J Non Newton Fluid Mech 144:98–121
5. Ammar A, Normandin M, Chinesta F (2010) Solving parametric complex fluids models in rheometric flows. J Non Newton Fluid Mech 165:1588–1601
6. Ammar A, Chinesta F, Cueto E, Doblaré M (2012) Proper generalized decomposition of time-multiscale models. Int J Numer Meth Eng 90(5):569–596
7. Ammar A, Huerta A, Chinesta F, Cueto E, Leygue A (2014) Parametric solutions involving geometry: a step towards efficient shape optimization. Comput Methods Appl Mech Eng 268:178–193
8. Amsallem D, Cortial J, Farhat C (2010) Towards real-time cfd-based aeroelastic computations using a database of reduced-order information. AIAA J 48:2029–2037
9. Amsallem D, Farhat C (2008) An interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA J 46:1803–1813
10. Athanasios C, Sorensen ADC, Gugercin S (2001) A survey of model reduction methods for large-scale systems. Contemp Math 280:193–220
11. Ballarin F, Manzoni A, Rozza G, Salsa S (2014) Shape optimization by free-form deformation: existence results and numerical solution for stokes flows. J Sci Comput 60(3):537–563
12. Barrault M, Maday Y, Nguyen NC, Patera AT (2004) An 'empirical interpolation' method: application to efficient reduced-

basis discretization of partial differential equations. CR Math 339(9):667–672
13. Bialecki RA, Kassab AJ, Fic A (2005) Proper orthogonal decomposition and modal analysis for acceleration of transient fem thermal analysis. Int J Numer Meth Engrg 62:774–797
14. Bognet B, Bordeu F, Chinesta F, Leygue A, Poitou A (2012) Advanced simulation of models defined in plate geometries: 3d solutions with 2d computational complexity. Comput Methods Appl Mech Eng 201–204:1–12
15. Bognet B, Leygue A, Chinesta F (2014) Separated representations of 3d elastic solutions in shell geometries. Adv Modell Simul Eng Sci 1(1):1–34
16. Bui-Thanh T, Willcox K, Ghattas O, Van Bloemen Waanders B (2007) Goal-oriented, model-constrained optimization for reduction of large-scale systems. J Comput Phys 224(2):880–896
17. Cancès E, Defranceschi M, Kutzelnigg W, Le Bris C, Maday Y (2003) Computational quantum chemistry: a primer. In: Handbook of numerical analysis, vol X, pp 3–270
18. Chaturantabut S, Sorensen DC (2010) Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 32:2737–2764
19. Chinesta F, Ammar A, Cueto E (2010) Proper generalized decomposition of multiscale models. Int J Numer Meth Eng 83(8–9):1114–1132
20. Chinesta F, Ammar A, Cueto E (2010) Recent advances in the use of the proper generalized decomposition for solving multidimensional models. Arch Comput Methods Eng 17(4):327–350
21. Chinesta F, Ammar A, Leygue A, Keunings R (2011) An overview of the proper generalized decomposition with applications in computational rheology. J Non Newton Fluid Mech 166(11):578–592
22. Chinesta F, Leygue A, Bordeu F, Aguado JV, Cueto E, Gonzalez D, Alfaro I, Ammar A, Huerta A (2013) PGD-based computational vademecum for efficient design, optimization and control. Arch Comput Methods Eng 20(1):31–59
23. Chinesta F, Keunings R, Leygue A (2014) The proper generalized decomposition for advanced numerical simulations. Springer, Switzerland
24. Chinesta F, Ladeveze P, Cueto E (2011) A short review on model order reduction based on proper generalized decomposition. Arch Comput Methods Eng 18:395–404
25. Dowell E, Hall K (2001) Modeling of fluid-structure interaction. Annu Rev Fluid Mech 33:445–490
26. Ghnatios C, Chinesta F, Cueto E, Leygue A, Poitou A, Breitkopf P, Villon P (2011) Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion. Compos A Appl Sci Manuf 42(9):1169–1178
27. Ghnatios C, Masson F, Huerta A, Leygue A, Cueto E, Chinesta F (2012) Proper generalized decomposition based dynamic data-driven control of thermal processes. Comput Methods Appl Mech Eng 213–216:29–41
28. Girault M, Videcoq E, Petit D (2010) Estimation of time-varying heat sources through inversion of a low order model built with the modal identification method from in-situ temperature measurements. Int J Heat Mass Transf 53:206–219
29. Gonzalez D, Masson F, Poulhaon F, Cueto E, Chinesta F (2012) Proper generalized decomposition based dynamic data driven inverse identification. Math Comput Simul 82:1677–1695
30. González D, Alfaro I, Quesada C, Cueto E, Chinesta F (2015) Computational vademecums for the real-time simulation of haptic collision between nonlinear solids. Comput Methods Appl Mech Eng 283:210–223
31. Gonzalez D, Cueto E, Chinesta F (2014) Real-time direct integration of reduced solid dynamics equations. Int J Numer Methods Eng 99(9):633–653
32. González D, Cueto E, Chinesta F (2015) Computational patient avatars for surgery planning. Ann Biomed Eng 44(1):35–45
33. El Halabi F, González D, Chico A, Doblaré M (2013) FE2 multiscale in linear elasticity based on parametrized microscale models using proper generalized decomposition. Comput Methods Appl Mech Eng 257:183–202
34. Hesthaven J, Rozza G, Stamm B (2015) Certified reduced basis methods for parametrized partial differential equations. Springer, New York
35. Heyberger C, Boucard P-A, Neron D (2013) A rational strategy for the resolution of parametrized problems in the PGD framework. Comput Methods Appl Mech Eng 259:40–49
36. Karhunen K (1946) Uber lineare methoden in der wahrscheinlichkeitsrechnung. Ann Acad Sci Fennicae Ser Al Math Phys 37:1–79
37. Ladeveze P (1985) On a family of algorithms for structural mechanics (in french). Comptes Rendus Académie Des Sci Paris 300(2):41–44
38. Ladeveze P (1989) The large time increment method for the analyze of structures with nonlinear constitutive relation described by internal variables. Comptes Rendus Académie Des Sci Paris 309:1095–1099
39. Lamari H, Ammar A, Cartraud P, Legrain G, Jacquemin F, Chinesta F (2010) Routes for efficient computational homogenization of non-linear materials using the proper generalized decomposition. Arch Comput Methods Eng 17(4):373–391
40. Lassila T, Manzoni A, Quarteroni A, Rozza G (2013) A reduced computational and geometrical framework for inverse problems in hemodynamics. Int J Numer Methods Biomed Eng 29(7):741–776
41. Lassila T, Rozza G (2010) Parametric free-form shape design with pde models and reduced basis method. Comput Methods Appl Mech Eng 199(23):1583–1592
42. Laughlin RB, Pines D (2000) The theory of everything. Proc Nat Acad Sci 97(1):28–31
43. Lee JA, Verleysen M (2007) Nonlinear dimensionality reduction. Springer, New York
44. Loève MM (1963) Probability theory. The university series in higher mathematics, 3rd edn. Van Nostrand, Princeton, NJ
45. Lopez E, Gonzalez D, Aguado JV, Abisset-Chavanne E, Lebel F, Upadhyay R, Cueto E, Binetruy C, Chinesta F (2016) A manifold learning approach for integrated computational materials engineering (in press)
46. Lorenz EN (1956) Empirical orthogonal functions and statistical weather prediction. MIT, Departement of Meteorology, Scientific Report Number 1, Statistical Forecasting Project
47. Maday Y, Patera AT, Turinici G (2002) A priori convergence theory for reduced-basis approximations of single-parametric elliptic partial differential equations. J Sci Comput 17(1-4):437–446
48. Maday Y, Ronquist EM (2004) The reduced basis element method: application to a thermal fin problem. SIAM J Sci Comput 26(1):240–258
49. Manzoni A, Quarteroni A, Rozza G (2012) Computational reduction for parametrized pdes: strategies and applications. Milan J Math 80:283–309
50. Manzoni A, Quarteroni A, Rozza G (2012) Model reduction techniques for fast blood flow simulation in parametrized geometries. Int J Numer Methods Biomed Eng 28(6–7):604–625
51. Manzoni A, Quarteroni A, Rozza G (2012) Shape optimization for viscous flows by reduced basis methods and free-form deformation. Int J Numer Meth Fluids 70(5):646–670
52. Mena A, Bel D, Alfaro I, Gonzalez D, Cueto E, Chinesta F (2015) Towards a pancreatic surgery simulator based on model order reduction. Adv Model Simul Eng Sci 2(1):31
53. Millán D, Arroyo M (2013) Nonlinear manifold learning for model reduction in finite elastodynamics. Comput Methods Appl Mech Eng 261–262:118–131

54. Niroomandi S, González D, Alfaro I, Bordeu F, Leygue A, Cueto E, Chinesta F (2013) Real-time simulation of biological soft tissues: a PGD approach. Int J Numer Methods Biomed Eng 29(5):586–600
55. Niroomandi S, Gonzalez D, Alfaro I, Cueto E, Chinesta F (2013) Model order reduction in hyperelasticity: a proper generalized decomposition approach. Int J Numer Meth Eng 96(3):129–149
56. Nouy A (2007) A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. Comput Methods Appl Mech Eng 196:4521–4537
57. Willcox K, Benner P, Gugercin S (2016) A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 57(4):483–531
58. Park HM, Cho DH (1996) The use of the Karhunen-Loève decomposition for the modeling of distributed parameter systems. Chem Eng Sci 51(1):81–98
59. Patera AT, Rozza G (2007) Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations. Technical report, MIT Pappalardo Monographs in Mechanical Engineering
60. Pruliere E, Chinesta F, Ammar A (2010) On the deterministic solution of multidimensional parametric models using the proper generalized decomposition. Math Comput Simul 81(4):791–810
61. Quarteroni A, Rozza G, Manzoni A (2011) Certified reduced basis approximation for parametrized pde and applications. J Math Ind 1(1):1–49
62. Quarteroni A, Rozza G (2003) Optimal control and shape optimization of aorto-coronaric bypass anastomoses. Math Models Methods Appl Sci 13(12):1801–1823
63. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
64. Rozza G (2014) Fundamentals of reduced basis method for problems governed by parametrized pdes and applications. In: Ladeveze P, Chinesta F (eds) CISM lectures notes "Separated Representation and PGD based model reduction: fundamentals and applications". Springer, New York
65. Ronquist EM, Maday Y (2002) A reduced-basis element method. C R Acad Sci Paris Ser I 335:195–200
66. Rozza G, Huynh DBP, Patera AT (2008) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations - application to transport and continuum mechanics. Arch Comput Methods Eng 15(3):229–275
67. Ryckelynck D (2003) A priori model reduction method for the optimization of complex problems. In: Workshop on optimal design of materials and structures, ecole polytechnique, Palaiseau, Paris (France)
68. Ryckelynck D (2005) A priori hyperreduction method: an adaptive approach. J Comput Phys 202(1):346–366
69. Ryckelynck D, Chinesta F, Cueto E, Ammar A (2006) On the a priori model reduction: overview and recent developments. Arch Comput Methods Eng 12(1):91–128
70. Ryckelynck D, Hermanns L, Chinesta F, Alarcon E (2005) An efficient 'a priori' model reduction for boundary element models. Eng Anal Bound Elem 29(8):796–801
71. Scholkopf B, Smola A, Muller KR (1999) Kernel principal component analysis. In: Advances in kernel methods—suport vector learning. MIT Press, New York, pp 327–352
72. Schölkopf B, Smola A, Müller K-R (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
73. Videcoq E, Quemener O, Lazard M, Neveu A (2008) Heat source identification and on-line temperature control by a branch eigenmodes reduced model. Int J Heat Mass Transf 51:4743–4752
74. Vitse M, Neron D, Boucard P-A (2014) Virtual charts of solutions for parametrized nonlinear equations. Comput Mech 54(6):1529–1539
75. Volkwein S (2001) Model reduction using proper orthogonal decomposition. Technical report, lecture notes. Institute of Mathematics and Scientific Computing, University of Graz
76. Wang Q (2012) Kernel principal component analysis and its applications in face recognition and active shape models. ArXiv preprint arXiv:1207.3538
77. Zimmer VA, Lekadir K, Hoogendoorn C, Frangi AF, Piella G (2015) A framework for optimal kernel-based manifold embedding of medical image data. Comput Med Imaging Graph 41:93–107 Machine Learning in Medical Imaging