# Engineering empowered by physics-based and data-driven hybrid models: A methodological overview

Victor Champaney, Francisco Chinesta, Elias Cueto

# Engineering empowered by physics-based and data-driven hybrid models: A methodological overview

**Victor Champaney**[1] · **Francisco Chinesta**[1] · **Elias Cueto**[2]

**Abstract**
Smart manufacturing implies creating virtual replicas of the processing operations, taking into account the material dimension and its multi-physics transformation when forming processes operate. Performing efficient, that is, online accurate predictions of the induced properties (including potential defects) of the formed part (to optimally control the process parameters) needs moving beyond usual offline simulation based on nominal models, and proceeds by assimilating data. This will serve, from one side, to keep the model calibrated, and from the other, to enrich the model and its associated predictions, to avoid bias, to improve accuracy or for performing online diagnosis, by advertising on preventive maintenance. For all these purposes, a new alliance between physics-based and data-driven modelling approaches seems a very valuable route for empowering engineering in general, and smart manufacturing in particular. The present paper revisits the main methodologies involved in the construction of the component or system Hybrid Twins.

**Keywords** Smart manufacturing · Physics-based modelling · Model order reduction · PGD · Data-driven modelling · Artificial intelligence · Hybrid twins · Diagnosis and prognosis

## Introduction

The main aim of engineering is moving from the product itself to the management of its performance all along its operational life. Even if the main aim of engineering was always the decision making, in virtue of accurate diagnosis and prognosis, it was not possible in the past due to the limitation in monitoring large components or systems. The irruption of sensors, communication, data-analytics, ... in all the domains of science, technology and society, within the so-called Internet of Things (IoT), originated major changes in engineering practices.

✉ Victor Champaney
victor.champaney@ensam.eu

Francisco Chinesta
francisco.chinesta@ensam.eu

Elias Cueto
ecueto@unizar.es

[1] PIMM Laboratory & ESI Group Chair, Arts et Métiers Institute of Technology, CNRS, Cnam, HESAM Université, 151 boulevard de l'Hôpital, 75013 Paris, France

[2] Aragon Institute of Engineering Research, Universidad de Zaragoza, Maria de Luna s/n, 50018 Zaragoza, Spain

The last century engineering—very successful from multiple points of view—employed models expected to represent the physical system itself (which we will refer to as *nominal models*), calibrated by using data appropriately collected using sophisticated testing machines. Then, several loading scenarios, expected spanning the one that the component will experience during its real life, were applied to obtain as main output the component or system performances.

When the mathematical models became too complex for envisaging their analytical solution (within an almost analogical world) computers came to rescue for efficiently solving them. Despite of the numerous simplifying hypotheses involved in the model derivation, in its numerical treatment as well as in the definition of the loadings considered for the performances analyses, the success stories were abundant and some of them simply impressive, with most of the XX century technology as definitive proof. The considered engineering workflow is sketched in Fig. 1(left).

However, at the beginning of the third millennium, as just commented, managing the performance seems a real opportunity. Today when buying an electrical drill, we are in fact buying a good quality hole; nowadays engineering is more concerned by the number of hours of fly than by the

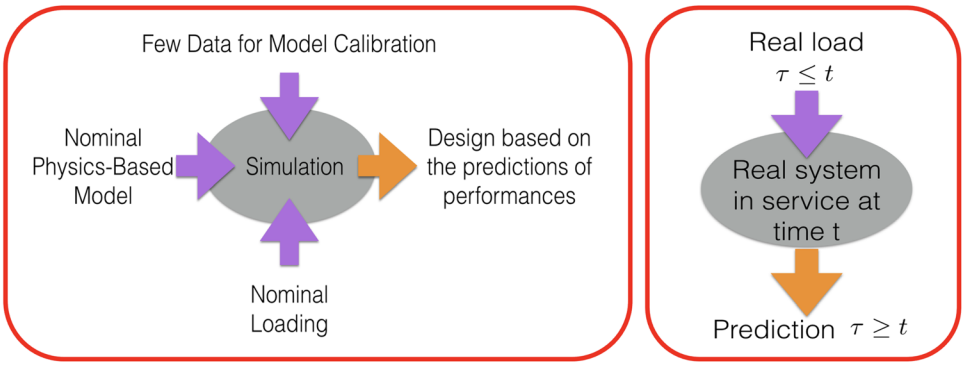


**Fig. 1** The past (left) versus the present (right) engineering practices

aircraft engines themselves, ... This is the new face of the new performances-based engineering.

Engineering is concerned by the real system at the present time, with the knowledge of the loading that it experienced during its past life (fully described by the collected data). Thus, from all this data and knowledge we would like to make efficient diagnosis (at the present time) and prognosis (by projecting the present knowledge into the future) for anticipating and taking the right decision at the right time. The workflow of this new engineering is sketched in Fig. 1(right).

The domain in which engineering operates is expanding, expecting to address systems the more and more large, complex, and involving the more are more variability and uncertainty. As just mentioned, engineering is not only addressing the design of components or systems, but monitoring them lo learn from their functioning, to make efficient (accurate and fast) diagnosis and prognosis, for the optimal decision making. To make all this possible, we need new techniques able to predict very accurately and very quickly complex system responses.

The two main words have been introduced, in a very simple and natural way: fast and accurate. In the past these two words were almost immiscible. When searching for fast responses models were degraded, simplified, ... and an amount of accuracy definitively lost. When searching for accuracy, models were kept rich enough, but then, the responses had to wait, or very powerful computers were used, but the last solution is not attainable in a context of democratizing simulation for rendering it accessible to small industries.

The two existing paradigms, the one of the past, mainly focused on physics and analogical knowledge, and the more recent and digital one, the one based on the employ of data manipulated by advanced artificial intelligence techniques, both present their inherent limitations. Both paradigms are sketched in Fig. 2.

As it can be noticed from Fig. 2, the physics-based modelling framework produces responses that approximate quite well the real ones, as soon as such an accurate model exist. The main difficulties when considering such a physics-based approach are: (i) the precision of the model itself; (ii) the impact that variability and uncertainty have in the predictions; and (iii) the computing time required for solving the complex and intricate mathematical models.

On the other hand, the data-driven framework is not fully satisfactory, because as it can be noticed in that figure, even if the data (assumed noise-free and symbolized by the blue small squares) represent well the reality, in between, the use of a low degree interpolation (in this image piecewise linear) produces a quite poor approximation of the real system response. Of course, by increasing the number of collected data, one could expect approximating the real solution better, however, data is not always

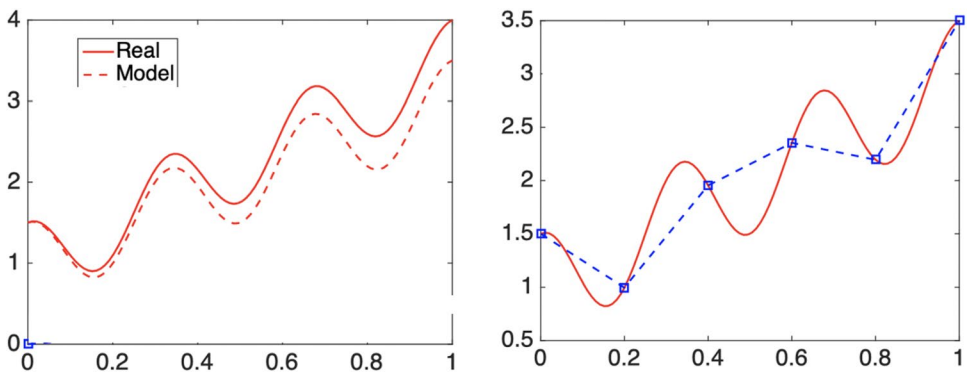


**Fig. 2** The physics-based (left) versus the data-driven (right) modelling paradigms

simple to collect, not always possible to have access to it, and in all the cases collecting data is expensive (cost of sensors, cost of communication and analysis, ...). Equipping a very large industrial or civil infrastructure with millions of sensors to cover all its spatial dimension seems simply unreasonable. Moreover, even when the solution will be well approximated, two difficulties persist: (i) the solution explainability, compulsory to certify solutions and decisions; and (ii) the domain of validity, because even data can be interpolated in a quite safe manner, extrapolating them becomes extremely risky.

In view of the limitations of both existing frameworks, a gateway consists of allying both of them to conciliate these two targets: fast and accurate. The hybrid paradigm, sketched in Fig. 3, is a valuable and appealing option. The hybrid paradigm considers the reality expressible from the addition of two contributions: the existing knowledge (the state-of-the-art physics-based models or any other kind of knowledge-based models) and the part of the reality that our models ignore, the so-called ignorance (also called deviation, gap, discrepancy, ...). As can be noticed in Fig. 3, the model represents quite accurately, even if not perfectly, the reality.

In some cases, the discrepancy becomes simpler in the sense of cheaper to describe (less data) or to explain (compared with a fully data-driven approach), with most of its richness captured and explained by the physics-based model. Thus, as in transfer learning, or reinforced transfer learning, we try to approximate a richer behavior from another close enough and well stablished. This hybrid approach that combines the use of physics and data, can be applied in different ways, some of them reported below:

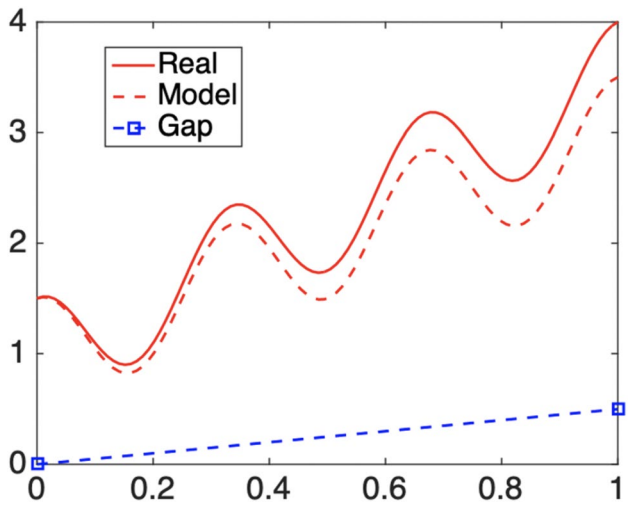


**Fig. 3** The Hybrid paradigm: the real systems is expressed from a physics-based model enriched from the data-driven model of the discrepancy

- In some cases, when the model captures most of the solution complexity, the correction must describe a discrepancy that could exhibits much smaller nonlinearities, as was the case treated in [40, 41], where the same amount of data performed better within the hybrid than within the fully data-driven framework.
- Sometimes the physics-based model operates very accurately in a part of the domain, whereas the nonlinear behavior localizes in a very small region that can, in that case, be captured by a data-driven learned model, as considered in [39] for addressing the inelastic behavior of spot-welds.
- When considering constitutive modeling of materials, different frameworks exist:
  - The first is based on the collected data, complemented with first principles (and their associated variational formulations), as considered in the seminal work of Ortiz [29] with a lot of extensions.
  - A second approach considers the collected data to lie on a manifold embedded into the higher-dimensional behavior space [24, 27, 33].
  - A third approach includes a number of quite general rules representing the material description, for constructing the so-called *constitutive manifold* that intimately embraces data and existing knowledge [32].
  - Other physics-informed approaches, within a thermodynamical dissipative setting [17].
  - Within the augmented rationale (or hybrid paradigm) the real behavior can be assumed represented by a first order one (calibrated at best from the available data) complemented by an enrichment (or correction) filling the gap between the collected data and the predictions obtained from the chosen and calibrated model [18].
- The hybrid modeling can also transfer the existing knowledge slightly outside its domain of applicability with small amount of collected data, as performed in [35] for correcting the usual structural beam models.
- Sometimes the discrepancy concerns an imperfect alignment of the solution between the prediction and the measures. That discrepancy seems very high when evaluating it at each location, however, a small transport allows aligning both solutions. Optimal transport is very suitable in these situations, where the hybrid model consists of the usual nominal model enriched from a parametric correction formulated in an optimal transport setting as described in [45, 46].
- In [26] a correction of a given yield function was proposed and obtained from the deviation between the results predicted by using it and the measures obtained

at the structure level. Being the correction much simpler it can be described from few approximation functions, much less that the ones needed for approximating the resulting yield function.
– Finally, when addressing processing and performances, the hybrid formulation, of different granularities, can exhibit advantages (amount of data, ability to explain, knowledge transfer, ...), at the heart of the digital twin developments [3, 7, 12, 16, 28, 47].

Moreover, as bonus, the major contribution of the hybrid model, the one based on the state-of-the-art physics-based model, can be explained and is nowadays certifiable, fact that allows leaving the blackbox image characteristic of fully data-driven models, to obtain a light-gray box. It is important also to remark, that even if it seems conclusive the fact that data is helping the models to be more accurate, the existing knowledge also transform the big-data paradigm into a smarter one, where thanks to the existing knowledge, one can conceive the data to be collected and the optimal locations and time to perform the measurements.

Despite of the undoubtable advantages in using the hybrid framework, three difficulties persist:

1. Addressing complex and heterogeneous data, for instance, the manufacturing card of a part, containing processing data involving categorial, qualitative and codified informations, like the commercial designations of materials, ... Sometimes, processing data also concerns time series or images (microstructures, ...), both with rich topological content.
2. Addressing physics in almost real-time, compulsory for constructing the so-called Hybrid Twin, to take into account the contribution (sometimes the most important) of state-of-the art physics-based models, that now must be solved in almost-real time, enabling efficient diagnosis and prognosis.
3. Finally, the online construction of a data-driven model, rich enough, from few data and in real-time, constitutes a challenge for most experienced artificial intelligence and machine learning techniques.

Reconciliation and allying physics-based and data-driven models, leads to the so-called Hybrid Twin [12], that constitutes a digital twin variant. Digital twins are extensively used for a diversity of engineering applications, among them we could cite [16, 28, 47]. It constitutes a digital replica of a material, process, structure, component, systems or systems of systems, able to replace or substitute the real system for anticipating future responses, to access to its intimate state, while retaining the following features, qualities and functionalities: (i) accuracy guaranteed by the hybrid approach; (ii) frugal, being based on the smart-data paradigm; (iii) holistic throughout all the involved physics and description scales; (iv) providing real-time responses by invoking advanced model order reduction techniques; (v) explicable and certifiable; (vi) adaptable; (vii) reliable; (viii) resilient; (ix) informed -IoT; (x) systemic to address the system within its environment; and (xi) usable.

## On the description of complex data and the associated metrics

Manipulating data for clustering or classification purposes is very common. It is expected that as soon a collected data is found belonging to a certain group, it will share some properties that characterize the members of the group, that is, proximity implies similar responses (in a certain sense).

The main issue is precisely how evaluating the proximity between data, because such proximity implies the use of adequate metrics, and sometime those metrics are not straightforward defined. This is for example the case when manipulating data associated with manufacturing processes described by their processing cards, consisting of the name of the employee involved in the operation, the designation of the employed materials, the temperature of the oven used for curing the part and the processing time. In that case, comparing two parts, from their respective processing cards, becomes a tricky issue. In these circumstances the comparison metrics itself must be learned from the existing training data, as is the case when using decision-trees (or its random forest counterpart) [5, 30], code-to-vector [2] or neural networks [19].

In other cases, the data exhibits a huge topological content, as it is the case when processing time-series or images of complex microstructures (foams, ...). The question is again how comparing two time-series or two images, that seem similar in a certain intuitive sense, but are impossible to match point to point. In many circumstances, techniques that operate by trying to align the data, like DTW [36, 43], fail to accomplish the task. Thus be most valuable route seems the one of extracting valuable statistical descriptors, to be then used in unsupervised clustering, supervised classification or for modeling purposes, as inputs of advance nonlinear regressions.

One of the most appealing techniques for performing that extraction consists of using the so-called topological data analysis (TDA). It has been successfully considered in our former works for addressing complex mesostructures [48], time-series [13], rough surfaces [14] and shapes [15], with the aim of classifying and also constructing robust regressions expressing properties or performances from the input data expressed from its topological description.

To briefly summarize the main steps concerned by the application of TDA, we consider below its application in the analysis of time series and images.

### Time series

We consider the time-series $\mathcal{Y} = \{y_1, y_2, ..., y_k, ...\}$, with $y_k \equiv y(t_k)$ (or $y_k \equiv y(x_k)$ when considering a profile $y$ parametrize by the coordinate $x$). Now, the local maximums and local minimums are identified, and taking into account its respective local-minimum to local-maximum proximity, matched, giving rise to the $\mathtt{P}$ pairs $(m_p, M_p)$, $p = 1, ..., \mathtt{P}$, where the lower case $m$ is used for the local minimum and the capital letter $M$ for the maximum. In the TDA terminology, minimum and maximum pairs are referred by birth and death respectively.

When reporting the pairs $(m_p, M_p)$ into a 2D diagram, with the birth in the abscissa axis and the death if the ordinate axis, the $\mathtt{P}$ points will constitute the so-called persistence diagram.

The persistence diagram can be transformed into the so-called lifetime diagram, by representing in the ordinate axis the lifetime, i.e. $M_p - m_p$, $\forall p$, instead of the death $M_p$.

Now, by associating to each point in the lifetime diagram a bivariate normal distribution and by embedding the lifetime diagram into a lager 2D domain covering (up-to a certain tolerance) the support of all the regularized points in the life-time diagram, after some technical manipulations, the so-called persistence image is obtained. It can viewed as a regularized form of the lifetime diagram.

The greatest merit of such a representation is its inherited invariance, that implies that different time series with similar topological content (similar persistence diagram) and even if they seems very different when expressed in the time or $x$-domain, their persistence image will be almost the same. Adequate metrics can be applied on both, the persistence diagram and the persistence images, mostly on the persistence images, making possible comparing, classifying and using them for modelling (regressions) purposes.

### Complex miscrostructures

In the case of complex microstructures described from 2D images, a set of structured or unstructured points is distributed on the analyzed phase (e.g. the material when assuming a cellular material, like a foam). Now, the evolution of points, edges and triangles is evaluated with respect to an increasing distance, is evaluated.

When this distance starts with a null value only points exist. Then, when the distance increases points connect, to create edges. When the edges create a closed loop (a hole), a topological feature appears, and the current value of the distance represents the birth of the feature. Then, for certain value of the distance the hole disappears, fully filled by the just created triangles inside. This value correspond with the death. The birth and the death represents a point to be reported into the persistence diagram.

Then, moving from the persistence diagram to the lifetime diagram and finally to the persistence image, follows the same rationale previously described for the time series.

## Physics in real-time

When looking for an approximation of the solution $u(\boldsymbol{x}, t)$ of a given partial differential equation –PDE– expected governing a particular physical phenomenon, here assumed scalar without loss of generality, the standard finite element method considers the approximation

$$u(\boldsymbol{x}, t) = \sum_{i=1}^{\mathtt{N}} U_i(t) N_i(\boldsymbol{x}), \tag{1}$$

where $U_i$ denotes the value of the unknown field at node $i$, $\boldsymbol{x}_i$, and $N_i(\boldsymbol{x})$ represents the its associated approximation function—shape function in the finite element method (FEM) terminology—. Here, $\mathtt{N}$ refers to the number of nodes considered to approximate the field in the domain $\Omega$ where the problem is defined.

This approximation results in an algebraic problem of size $\mathtt{N}$ in the linear case, or the solution of many of them in the general transient and nonlinear cases. In order to alleviate the computational cost, model order reduction techniques have been proposed and are nowadays intensively used.

When considering POD-based model order reduction [11], a learning stage allows extracting the significant modes $\phi_i(\boldsymbol{x})$ that best approximate the solution. Very often a reduced number of modes $\mathtt{R}$ ($\mathtt{R} \ll \mathtt{N}$) suffices to approximate the solution of problems similar to the one that served to extract the modes at the learning stage. In other words, while finite element shape functions are general and can be employed in virtually any problem, the reduced-order basis is restricted to the domain where the learned process was accomplished.

By expressing the solution $u(\boldsymbol{x}, t)$ onto the reduced basis $\{\phi_1(\boldsymbol{x}), \ldots, \phi_{\mathtt{R}}(\boldsymbol{x})\}$

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

the resulting discrete problem will now require the solution of a linear system of equations of size $\mathtt{R}$, instead of size $\mathtt{N}$, which is the actual size of the finite element solution. This often implies impressive savings in computing time. Addressing nonlinear models requires the use of specific strategies to ensure solution efficiency [8].

Equation 1 or 2 involve a finite sum of products composed by time-dependent coefficients multiplied by space functions. These space function are the well-known finite element shape functions when no prior knowledge about the structure of the problem exists, or the modes extracted by applying POD.

A generalization of this procedure consists in assuming that space functions are also unknown. This makes it necessary to compute both time and space functions, on the fly. Thus, the resulting approximation reads

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

Since the pairs of space and time functions in Eq. 3 are unknown, their determination will define a nonlinear problem. Obviously, it will require some form of linearization. This linearization procedure has been studied in some of the author's former works [9, 10]. The final approximation, Eq. 3, will require the solution of about $\mathtt{M}$ problems, with $\mathtt{M} \ll \mathtt{N}$ and $\mathtt{M} \sim \mathtt{R}$.

Degenerate geometries (beams, plates, shells, layered domains such as composite materials) are specially well suited for a space domain separation. If the domain $\Omega$ can be decomposed as $\Omega = \Omega_x \times \Omega_y \times \Omega_z$, the solution $u(x,y,z)$ could be approximated in turn by a separated representation of the type [10]

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

which is specially advantageous, since it gives rise to a sequence of one-dimensional problems instead of the typical three-dimensional complexity. For some geometries, like plates or shells, in-plane/out-of-plane this separated representation becomes specially interesting,

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

where the obtained complexity of the problem is roughly the typical of a two-dimensional problem, i.e., the calculation of in-plane functions $X_i(x,y)$.

A very interesting case is that of space-time-parameter separated representations [9]. In this framework a so-called *computational vademecum* (also known as abacus, virtual charts, nomograms, ...) can be developed so as to provide a sort of computational response surface for the problem at hand, but without the need for a complex sampling in high dimensional domains. It has been successfully employed in problems like simulation, optimization, inverse analysis, uncertainty propagation and simulation-based control, to cite a few [9]. Once constructed off-line, this sort of response surface provides results under very stringent real-time constraints—in the order of milliseconds—by just invoking this response surface instead of simulating the whole problem.

Thus, when the unknown field is a function of space, time and a number of parameters $\mu_1, \dots, \mu_{\mathtt{Q}}$, the subsequent separated representation could be established as

$$u(\boldsymbol{x},t,\mu_1,\dots,\mu_{\mathtt{Q}}) \approx \sum_{i=1}^{\mathtt{M}} X_i(\boldsymbol{x})T_i(t)\prod_{j=1}^{\mathtt{Q}} M_i^j(\mu_j). \tag{6}$$

## The standard, intrusive, PGD constructor

We assume the generic model governing the evolution of the field under consideration $u(\boldsymbol{x},t)$

$$\mathcal{R}(u(\boldsymbol{x},t);\mu) \equiv \mathcal{L}(u(\boldsymbol{x},t);\mu) - F(\boldsymbol{x},t;\mu) = 0, \text{ in } \Omega \tag{7}$$

where $\mathcal{L}(\bullet)$ represents a generic linear or nonlinear differential operator, $F(\bullet)$ the so-called forcing term, and $\mu$ is a parameter (here a single parameter for the sake of simplicity) that could affect the domain $\Omega$ in which the problem is defined, the physical model itself or the forcing term. Thus, it is expected that the solution will depend on the considered value of the parameter $\mu$, i.e. $u(\boldsymbol{x},t;\mu)$.

The Proper Generalized Decomposition (PGD) is based on the solution separated representation, originally proposed for defining non-incremental transient solutions [31]. In the parametric setting, the PGD proceeds by assuming a fully separated representation of the problem solution, where parameters are assumed extra-coordinates [9]. Thus, the parametric solution approximation $u^{\mathtt{M}}(x,t,\mu)$ reads

$$u^{\mathtt{M}}(x,t,\mu) = \sum_{i=1}^{\mathtt{M}} X_i(x)T_i(t)M_i(\mu), \tag{8}$$

with $x \in \Omega$, $t \in \mathcal{T}$ and $\mu \in \mathcal{I}$.

To compute the different unknown functions involved in the separated representation Eq. 8 the usual weighted residual form is extended according to

$$\int_{\Omega\times\mathcal{T}\times\mathcal{I}} u^*(x,t,\mu)\mathcal{R}(u(x,t,\mu))\,dx\,dt\,d\mu = 0. \tag{9}$$

Within a Galerkin framework, when looking for the functional product $m$, the trial and test functions to be employed within the integral form (9), read respectively

$$\begin{aligned} u^m(x,t,\mu) &= \sum_{i=1}^{m-1} X_i(x)T_i(t)M_i(\mu) + X_q(x)T_q(t)M_q(\mu) \\ &= u^{m-1}(x,t,\mu) + X_m(x)T_m(t)M_m(\mu), \end{aligned} \tag{10}$$

and

$$u^*(x,t,\mu) = X^*(x)T_m(t)M_m(\mu) + X_m(x)T^*(t)M_m(\mu) + X_m(x)T_m(t)M^*(\mu). \quad (11)$$

The separated representation constructor deeply described in [10] proceeds by using an alternate direction fixed point algorithm that computes the unknown function at the enrichment iteration $m$: $X_m(x)$ from $T_m(t)$ and $M_m(\mu)$ taken at the previous iteration, $T_m(t)$ from $X_m(x)$ and $M_m(\mu)$, and finally $M_m(\mu)$ from $X_m(x)$ and $T_m(t)$. The iteration continues until reaching the fixed point, and then the next functional product, $X_{m+1}(x)T_{m+1}(t)M_{m+1}(\mu)$, is considered.

The main difficulties in applying the just described procedure are the necessity of performing an affine decomposition of the problem residual involved in Eq. 9 for making possible the sequential calculation of each one of the functions involving the corresponding problem coordinate. Such affine decomposition is not direct, mainly in the case of nonlinear models [1, 9]. The other difficulty is related to the procedure intrusiveness that makes difficult its use in tandem with usual commercial software.

## Non-intrusive PGD constructor

To overpass the just referred difficulties, one option consists in constructing metamodels (also known as surrogates, response surfaces, virtual charts or vademecums).

The construction of these metamodels can be performed by considering a quite simple workflow: (i) defining a sampling of the parametric space; (ii) computing a high fidelity solution for each parameters choice; (iii) using an adequate regression for extending the solution known at the points in the sampling everywhere in the parametric space.

There are many alternatives for performing the first and third tasks, trying to conciliate: (i) a sparse and very reduced sampling, considering the best sampling points location, and with the number of points roughly linearly scaling with the dimension of the parametric space; (ii) the high fidelity solutions post-compression by extracting first and using then (by projection) reduced bases; (iii) rich enough approximation bases while avoiding overfitting, based on the use of sparse regularizations for enforcing parsimony [6] (elastic-net, ridge, lasso, ...); (iv) using orthogonal basis for evaluating sensibilities in a direct manner; and (v) efficiently addressing the high-dimensional spaces induced by the multi-parametric models, where the use of separated representations are specially suitable ... [4, 25, 42, 44].

In what follows the Sparse Subspace Learning (SSL) and the sparse PGD (sPGD) constructors, widely employed later, will be revisited.

### Sparse Subspace Learning

Again for the sake of simplicity, we assume that only one parameter is involved in the model, $\mu \in \mathcal{I} \equiv [\mu_{\min}, \mu_{\max}]$. The parametric solution $u(\boldsymbol{x}, t, \mu)$ is searched in the separated form

$$u(\boldsymbol{x}, t, \mu) \approx \sum_{i=1}^{\mathtt{M}} X_i(\boldsymbol{x}, t) M_i(\mu).$$

SSL [4] consists first in choosing a hierarchical basis of the parametric domain. The associated collocation points (the Gauss-Lobatto-Chebyshev) and the associated functions will be noted by: $(\mu_i^j, \xi_i^j(\mu))$, where indexes $i$ and $j$ refer to the $i$-point at the $j$-level.

At the first level, $j = 0$, there are only to points, $\mu_1^0$ and $\mu_2^0$, that correspond to the minimum and maximum value of the parameters that define the parametric domain, i.e. $\mu_1^0 = \mu_{\min}$ and $\mu_2^0 = \mu_{\max}$.

If we assume that a direct solver is available, i.e., a computer software able to compute the transient solution as soon as the value of the parameter has been specified, these solutions read $u_1^0(\boldsymbol{x}, t) = u(\boldsymbol{x}, t, \mu = \mu_1^0)$ and $u_2^0(\boldsymbol{x}, t) = u(\boldsymbol{x}, t, \mu = \mu_2^0)$ respectively.

Thus, the solution at level $j = 0$ could be approximated from

$$u^0(\boldsymbol{x}, t, \mu) = u_1^0(\boldsymbol{x}, t)\xi_1^0(\mu) + u_2^0(\boldsymbol{x}, t)\xi_2^0(\mu),$$

that in fact consists of a standard linear approximation since at the first level, $j = 0$, the two approximation functions read $\xi_1^0(\mu) = (\mu_2^0 - \mu)/(\mu_2^0 - \mu_1^0)$ and $\xi_1^0(\mu) = (\mu - \mu_1^0)/(\mu_2^0 - \mu_1^0)$, respectively.

At level $j = 1$ there is only one point located just in the middle of the parametric domain, i.e. $\mu_1^1 = 0.5(\mu_{\min} + \mu_{\max})$, being its associated interpolation function $\xi_1^1(\mu)$. It defines a parabola that takes a unit value at $\mu = \mu_1^1$ and vanishes at the other collocation points of level $j = 0$, $\mu_1^0$ and $\mu_2^0$ in this case. The associated solution reads $u_1^1(\boldsymbol{x}, t) = u(\boldsymbol{x}, t, \mu = \mu_1^1)$.

This solution contains a part already explained by the just computed approximation at the previous level, $j = 0$, expressed by

$$u^0(\boldsymbol{x}, t, \mu_1^1) = u_1^0(\boldsymbol{x}, t)\xi_1^0(\mu_1^1) + u_2^0(\boldsymbol{x}, t)\xi_2^0(\mu_1^1).$$

Thus, we can define the so-called *surplus* as

$$\tilde{u}_1^1(\boldsymbol{x}, t) = u_1^1(\boldsymbol{x}, t) - u^0(\boldsymbol{x}, t, \mu_1^1),$$

from which the approximation at level $j = 1$ reads

$$u^1(\boldsymbol{x}, t, \mu) = u^0(\boldsymbol{x}, t, \mu) + \tilde{u}_1^1(\boldsymbol{x}, t)\xi_1^1(\mu). \quad (12)$$

The process continues by adding surpluses when going-up with the hierarchical approximation level. An important aspect is that the norm of the surplus can be used as a local error indicator, and then when adding a level does not contribute sufficiently, the sampling process can stop.

The computed solution, as noticed in Eq. 12, ensures a separated representation. However, it could contain too many terms. In that circumstances a post-compression takes place by looking for a more compact separated representation [10].

When the model involves more parameters $\mu_1$ and $\mu_2$, here noted for the sake of notational simplicity $\mu$ and $\eta$, the hierarchical 2D basis, defined in the parametric space $(\mu, \eta)$ is composed by the cartesian product of the collocations points and the tensor product of the approximation bases.

Thus, the first level $j = 0$, is composed by the four points:

$$(\mu_1^0, \eta_1^0), (\mu_2^0, \eta_1^0), (\mu_2^0, \eta_2^0), (\mu_1^0, \eta_2^0),$$

with the associated interpolation functions

$$\xi_1^0(\mu)\varphi_1^0(\varphi), \xi_2^0(\mu)\varphi_1^0(\eta), \xi_2^0(\mu)\varphi_2^0(\eta), \xi_1^0(\mu)\varphi_2^0(\eta).$$

When moving to the next level, $j = 1$, the collocation points and approximation functions result from the combination of the zero-level of one parameter and the first level of the second one, i.e., the points are now: $(\mu_1^0, \eta_1^1), (\mu_2^0, \eta_1^1)$ and $(\mu_1^1, \eta_1^0), (\mu_1^1, \eta_2^0)$. In what concerns the interpolation functions they result from the product of the zero level in one coordinate and the level one in the other. It is worth noting that the point $(\mu_1^1, \eta_1^1)$ and its associated interpolation function is in fact a term of level $j = 2$.

The main issue when using the SSL is that even when limiting the approximation to the first level (the zero level), if $\mathtt{D}$ dimensions are involved, the number of collocation points becomes $2^{\mathtt{D}}$, that limits the use of the SSL when both, the number of parameters, and/or the degree to be considered (the so-called level) increase. For alleviating this issue the sPGD and its regularized variants were proposed, and will be summarized below.

## From the sparse PGD to its regularized variants: rsPGD and s2PGD

For the the sake of simplicity we consider two parameters, again noted by $\mu$ and $\eta$, and a field that depend on both them, $u(\boldsymbol{x}, t, \mu, \eta)$, with $n_t$ known solutions associated with $n_t$ different choices of the parameters, i.e. $u(\boldsymbol{x}, t, \mu_i, \eta_i) \equiv u^i(\boldsymbol{x}, t)$, $i = 1, ..., n_t$.

In general these solutions are expressed in a discrete manner, in the form of a matrix, where each column contains the solution at each node (rows) at a given time step. Thus, the discrete form of the solution $u(\boldsymbol{x}, t, \mu_i, \eta_i) \equiv u^i(\boldsymbol{x}, t)$ is noted $\mathbb{U}^i$, with the component $\mathbb{U}^i_{rs}$ referring to the solution at node $\boldsymbol{x}_r$ at time $t_s$, for the parameters choice $(\mu_i, \eta_i)$, i.e. $\mathbb{U}^i_{rs} \equiv u(\boldsymbol{x}_r, t_s, \mu_i, \eta_i)$.

However, finding a parametric regression of each component of the discrete solution, leading to $\mathbb{U}(\mu, \eta)$, is too expensive. For this reason, more than working with the discrete solution itself, reduced space and time bases are extracted: $\mathbf{X}_k$, $k = 1, ..., \mathtt{K}$, and $\mathbf{T}_l$, $l = 1, ..., \mathtt{L}$. Then, each discrete solution $\mathbb{U}^i$ can be expressed as

$$\mathbb{U}^i \approx \mathbb{X}\mathbb{D}^i\mathbb{T}, \tag{13}$$

where $\mathbb{X} = (\mathbf{X}_1 \ ... \ \mathbf{X}_{\mathtt{K}})$, $\mathbb{T}^T = (\mathrm{T}_1 \ ... \ \mathrm{T}_{\mathtt{L}})$ and $\mathbb{D}$, whose size is $\mathtt{K} \times \mathtt{L}$, is obtained by minimizing $\|\mathbb{U}^i - \mathbb{X}\mathbb{D}^i\mathbb{T}\|$.

Now, knowing $\mathbb{D}^i_{pq}$, i.e. each component of it for each choice $i$, $i = 1, ..., n_t$, of the parameters, $(\mu_i, \eta_i)$, $\mathbb{D}^i_{pq} \equiv \mathbb{D}_{pq}(\mu_i, \eta_i)$, the more general parametric expression $\mathbb{D}_{pq}(\mu, \eta)$ is searched.

For the sake of notational simplicity, we will describe in what follows the construction of regressions of a generic scalar $f$ that depends on the couple of parameters $(\mu, \eta)$, with $f^i \equiv f(\mu_i, \eta_i)$ known at the $n_t$ points in the sampling. Here, the scalar $f$ represents each component $\mathbb{D}_{pq}$, $p = 1, ..., \mathtt{K}$ and $q = 1, ..., \mathtt{L}$.

The goal is therefore to find a function $f^{\mathtt{M}}(\mu, \eta)$, expressible as a finite sum of $\mathtt{M}$ terms, according to

$$f^{\mathtt{M}}(\mu, \eta) = \sum_{j=1}^{\mathtt{M}} \mathcal{G}_j(\mu)\mathcal{H}_j(\eta), \tag{14}$$

able to approximate the known data $f^i \equiv f(\mu_i, \eta_i), i = 1, ..., n_t$.

Even if in the 2D case addressed here (with only two parameters) the separation of variables is not compulsory, we would like to propose a technique general enough and able to operate in highly multi-parametric settings.

At iteration $m$ the algorithm looks for the update $\mathcal{G}_m(\mu)\mathcal{H}_m(\eta)$ such that $f^m(\mu, \eta) = f^{m-1}(\mu, \eta) + \mathcal{G}_m(\mu)\mathcal{H}(\eta)$. For computing the searched functions, they are first approximated as $\mathcal{G}_m(\mu) = \mathbf{N}_m^{\mu,T}(\mu)\mathbf{a}_m$ and $\mathcal{H}_m(\eta) = \mathbf{N}_m^{\eta,T}(\eta)\mathbf{b}_m$, where $\mathbf{N}_m^{\mu}$ represents the basis considered for approximating the $m$-mode depending on the $\mu$-parameter, being $\mathbf{a}_m$ the associated weights, and similarly for the other direction ($\eta$-parameter): $\mathbf{N}_m^{\eta}$ and $\mathbf{b}_m$.

The update comes from the minimization problem:

$$\mathcal{G}_m(\mu)\mathcal{H}_m(\eta) = \underset{(G_m(\mu)\mathcal{H}_m(\eta))^*}{\operatorname{argmin}} \sum_{i=1}^{n_t} \|f^i - f^{m-1} + (\mathcal{G}_m(\mu)\mathcal{H}_m(\eta))^*\|_2^2, \tag{15}$$

that by using the notation below:

$$\boldsymbol{r} = \begin{pmatrix} f^1 - f^{m-1}(\mu_1, \eta_1) \\ \vdots \\ f^{n_t} - f^{m-1}(\mu_{n_t}, \eta_{n_t}) \end{pmatrix},$$

$$\mathbb{M}_\mu = \begin{pmatrix} \mathbf{N}_m^{\eta,T}(\eta_1)\boldsymbol{b}_m\mathbf{N}_m^{\mu,T}(\mu_1)) \\ \vdots \\ \mathbf{N}_m^{\eta,T}(\eta_{n_t})\boldsymbol{b}_m\mathbf{N}_m^{\mu,T}(\mu_{n_t})) \end{pmatrix},$$

$$\mathbb{M}_\eta = \begin{pmatrix} \mathbf{N}_m^{\mu,T}(\mu_1)\boldsymbol{a}_m\mathbf{N}_m^{\eta,T}(\eta_1)) \\ \vdots \\ \mathbf{N}_m^{\mu,T}(\mu_{n_t})\boldsymbol{a}_m\mathbf{N}_m^{\eta,T}(\eta_{n_t})) \end{pmatrix},$$

results in the two problems:

$$\boldsymbol{a}_m = \underset{\boldsymbol{a}_m^*}{\text{argmin}} \left\{ \|\boldsymbol{r} - \mathbb{M}_\mu \boldsymbol{a}_m^*\|_2^2 \right\}, \tag{16}$$

$$\boldsymbol{b}_m = \underset{\boldsymbol{b}_m^*}{\text{argmin}} \left\{ \|\boldsymbol{r} - \mathbb{M}_\eta \boldsymbol{b}_m^*\|_2^2 \right\}, \tag{17}$$

that are solved iteratively until reaching the fixed point.

This schema constitutes the heart of the sPGD constructor. However, when combining rich approximation bases $\mathbf{N}_m^\mu(\mu)$ and $\mathbf{N}_m^\eta(\eta)$, with data-sets $n_t$ not sufficiently rich (as it will be always the case when operating in highly multi-parametric settings) the just described procedure produces overfitting.

To alleviating overfitting, an adaptive procedure was proposed in [25] that consists in adapting the approximation bases, whose degree increases when advancing in the modal enrichment, that is, with the degree increasing with $m$.

Looking for a more versatile and automatic procedure, the *ridge* regularization was employed, giving rise to

$$\boldsymbol{a}_m = \underset{\boldsymbol{a}_m^*}{\text{argmin}} \left\{ \|\boldsymbol{r} - \mathbb{M}_\mu \boldsymbol{a}_m^*\|_2^2 + \lambda \|\boldsymbol{a}_m^*\|_2^2 \right\}, \tag{18}$$

$$\boldsymbol{b}_m = \underset{\boldsymbol{b}_m^*}{\text{argmin}} \left\{ \|\boldsymbol{r} - \mathbb{M}_\eta \boldsymbol{b}_m^*\|_2^2 + \lambda \|\boldsymbol{b}_m^*\|_2^2 \right\}. \tag{19}$$

When looking for sparsity in order to employ extremely rich approximations while exploiting parsimony, like proposed in the Sindy [6], it is well known that the lower is the employed norm the more intense results the sparsity enforcement. Many times the L1-norm (the so-called *Lasso* regularization) becomes a good compromise between sparsity enforcement and computational efficiency, and when combining *ridge* and *Lasso* the so-called *elastic net* regularization results, that when combined with the separated representation constructor reads [42]

$$\boldsymbol{a}_m = \underset{\boldsymbol{a}_m^*}{\text{argmin}} \left\{ \|\boldsymbol{r} - \mathbb{M}_\mu \boldsymbol{a}_m^*\|_2^2 + \lambda \left[ (1-\alpha)\|\boldsymbol{a}_m^*\|_2^2 + \alpha \|\boldsymbol{a}_m^*\|_1 \right] \right\}, \tag{20}$$

$$\boldsymbol{b}_m = \underset{\boldsymbol{b}_m^*}{\text{argmin}} \left\{ \|\boldsymbol{r} - \mathbb{M}_\eta \boldsymbol{b}_m^*\|_2^2 + \lambda \left[ (1-\alpha)\|\boldsymbol{b}_m^*\|_2^2 + \alpha \|\boldsymbol{b}_m^*\|_1 \right] \right\}. \tag{21}$$

Figure 4 summarizes the regression techniques usually employed to conciliate: (i) a sparse and very reduced sampling; (ii) rich enough approximation bases while avoiding overfitting, based on the use of sparse regularizations; (iii) using orthogonal basis for evaluating sensibilities in a direct manner; and (iv) efficiently addressing the high-dimensional spaces induced by the multi-parametric models, where the use of separated representations are specially suitable.

## Physics-aware artificial intelligence

Machine Learning (ML), a major protagonist of AI, is able to create predictive models from available or collected data, with an additional added value, the fact of providing that prediction in almost real-time.

However, creating models from scratch, simply because the existing models based on physics were not accurate enough, is not the best choice. Creating a model based on data from scratch needs a lot of data, and in engineering and technology, data is synonym of cost, and sometimes the data collection also implies to consider an ethical dimension, to fulfill existing regulations, or must address technical difficulties.

In our work we advocate by the alliance between both (i) the former analogical world of knowledge and physics-based models and (ii) the more recent digital world of data, manipulated by the more and more powerful (accurate, frugal and explainable -certifiable-) techniques of Artificial Intelligence.

This new alliance is materialized in the so-called Digital or Hybrid Twins, in which physics-based models are enriched, to decrease their intrinsic ignorance, in a pragmatic way, from the data representing the deviation between predictions and measurements. The hybrid paradigm not only allows reducing the amount of needed data that now is only expected describing the gap between the reality and the physics-based predictions, but also the ability of explaining the part of the model based on the existing physics or knowledge, and then facilitating the design or decision certification.

In order to reduce the amount of data to model the observable phenomena (when models do not exist or are too inaccurate) or for enriching the existing physics-based models within the hybrid paradigm previous introduced, physics-aware (also known as physics-informed) Artificial Intelligence (Machine Learning or Neural Networks) seems being the most appealing route.

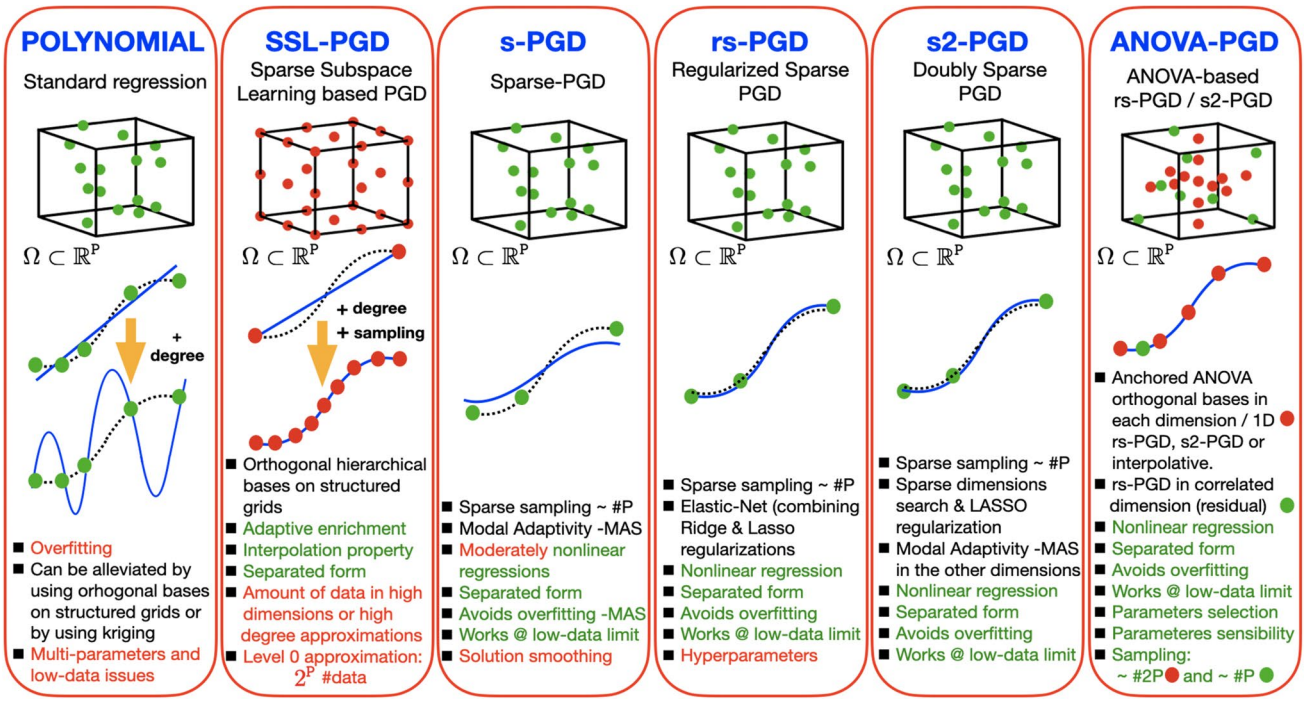


**Fig. 4** Advanced nonlinear regressors combining different sparse regularization. They enable addressing nonlinear behaviors with a low amount of data, while avoiding overfitting, in the multi-parametric setting efficiently addressed by the separated representations at the heart of the so-called PGD [42]

The so-called Physics Informed Neural Networks (PINN) [37, 38] considers the approximation of the unknown function $u(x, t)$ as a regression problem defined on an adapted neural network. Then, as soon as the physics is assumed fully known and adequately described by a partial differential equation, the derivatives involved in the differential operator can be applied on the NN and the residual nullity is enforced from the NN loss function.

Sometimes, everything concerning the physics is not fully known. If we consider an hyper-elastic material, the best option consists in learning the free energy by constructing a regression linking it to the state variables, in such a way that its derivatives leads to the constitutive equation. Then, the free energy is learned to be consistent with the collected data on the structural component, under the equilibrium constraints.

Similar procedures apply in the so-called Structure Preserving NN (also known as Thermodynamic Informed NN) where the free energy and the dissipation potential are computed in such a way that energy balance and entropy production are ensured [17, 18, 21, 22, 34, 35].

In those works, inspired from the GENERIC framework [20], the state $\mathbf{Z}$ evolution, $\dot{\mathbf{Z}}$, reads

$$\dot{\mathbf{Z}} = \mathbb{L}\nabla_Z \mathcal{H} + \mathbb{M}\nabla_Z \mathcal{S}, \tag{22}$$

where the first term of the right-hand member represents the reversible evolution (Hamiltonian contribution) whereas the second one represents the dissipative contribution, with $\mathcal{H}$ and $\mathcal{S}$ the energy and the entropy respectively.

Learning matrices $\mathbb{L}$ and $\mathbb{M}$, skew-symmetric the former, and symmetric and positive semi-definite the last, as well as both potentials $\mathcal{H}$ and $\mathcal{S}$ (subjected to some constraints: the Jacobi identities as well as the consistency conditions $\mathbb{L}\nabla_Z \mathcal{S} = \mathbf{0}$ and $\mathbf{M}\nabla_Z \mathcal{H} = \mathbf{0}$) is performed from the existing data concerning the state time evolution $\mathbf{Z}(t_i), i = 1, 2, ...$ The learned model has very interesting properties, as are the ones related to energy conservation and positive dissipation, enabling stable and accurate time integrators.

The main issue found when learning such thermodynamic-aware models is not the regression implementation, but the choice of the variables in the state vector, issue discussed in [27]. In very small systems the state variables are easily identified, however in large (continuous) systems such a choice is far of being trivial. In that case, many options exist. One among them consists of performing a dimensionality reduction. The use of most of manifold learning dimensionality reduction has as main drawback the unavailability of performing the inverse mapping for coming back from the reduced space to the departure one.

In the general nonlinear case, an appealing alternative consists in the use of NN-based autoencoders [23], where encoding and decoding is learned, while models operate

(and are learned) in a transparent way in the internal layer of reduced dimension where active reduced coordinates (the so-called latent variables or latent space) act. Autoencoders allow learning the best model representation, that in many cases, other than reducing the dimension of the model, allow reducing the model complexity (and then its nonlinearity) enabling the use of simpler and cheaper regression techniques operating at the level of the latent space, where sometimes linear regressions suffice.

## Use case

The present section illustrates the just introduced concepts, methodologies and procedures, on a forming process use-case involving stamping, and more particularly multi-stage stamping, where the deviations, due to inevitable uncertainties, accumulate throughout the whole process compromising the geometrical tolerancing of the final formed parts.

In what follows we consider a two-stage stamping process, where the first stage, depicted in Fig. 5, imprints a spherical shape on the original planar metallic sheet, whereas the second one acts on the shape that resulted from the first-stage, to imprint a final cylindrical shape, as illustrated in Fig. 6.

Both stages involve three main parameters: (i) the clamping force $F$; (ii) the tool-sheet friction coefficient $\xi$ and (iii) the punch-sheet friction coefficient $\eta$.

A parametric solution of the first stage was computed by using the non-intrusive PGD constructor previously introduced, operating on the solutions provided by the software PAM-STAMP (ESI Group) associated with the design of experiments associated with the SSL regression up-to level two.

The second stage operates on the shape that results from the first one. Thus, the parametric model of the second stage should include as parameters the ones describing the initial shape and its themomechanical state, representing the final state of the first stage.

As the final sate of the first stage is parametrized by the triplet $(F^I, \xi^I, \eta^I)$ (where the superscript $\bullet^I$ refers the first stage), one could expect having that triplet into the parametric space of the second-stage, and then expressing the nodal displacement according to $\mathbf{u}^{II}(F^I, \xi^I, \mu^I, F^{II}, \xi^{II}, \mu^{II})$. Such a parametric solution is depicted in Fig. 7.

Such a procedure faces three major difficulties: (i) the increase of the number of parameters when several stages are involved; (iii) the dependence of each stage model on the whole process; and (iii) the errors, inaccuracies and uncertainties accumulation from one stage to the next.

A way of increasing generality and accuracy consists in creating a generic parametric solution applicable to any stage. For that purpose, from the simulation of the whole process for different parameters, the final state of each stage is computed and the principal modes extracted by using the POD. Thus, in our case, from the $\mathtt{S}$ available snapshots $\mathbf{u}_i^I(F_i^I, \xi_i^I, \mu_i^I), i = 1, ..., \mathtt{S}$, the $\mathtt{R}$ most relevant POD modes $\phi_j$, $j = 1, ..., \mathtt{R}$, are extracted. In our case three modes ($\mathtt{R} = 3$) suffice for describing the final state of the first stage. Accordingly, each snapshot can be expressed as

$$\mathbf{u}_i^I \equiv \mathbf{u}^I(F_i^I, \xi_i^I, \mu_i^I) \approx \sum_{j=1}^{\mathtt{R}} \alpha_j^i \phi_j, \quad i = 1, ..., \mathtt{S}, \tag{23}$$

where taking into account the orthonormality of the reduced basis $\phi_j$, the coefficients read

$$\alpha_j^i = \mathbf{u}_i^I \cdot \phi_j. \tag{24}$$

More generally, any state resulting from the first stage could be expressed from

$$\mathbf{u}^I(F^I, \xi^I, \mu^I) \approx \sum_{j=1}^{\mathtt{R}} \alpha_j \phi_j. \tag{25}$$

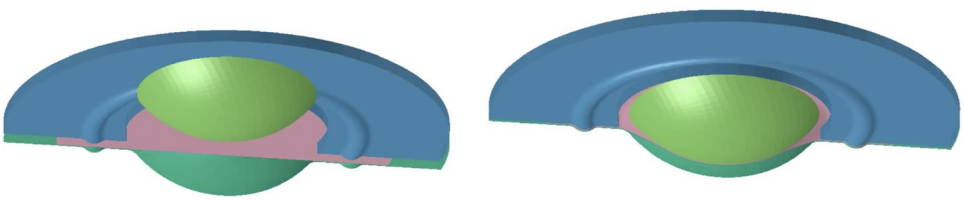

**Fig. 5** Two-stage stamping. First-stage: (left) initial configuration and (right) almost final configuration

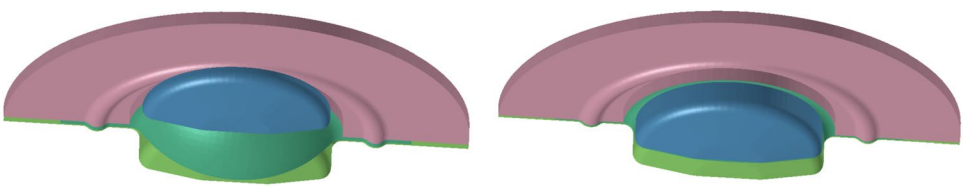

**Fig. 6** Two-stage stamping. Second-stage: (left) initial configuration and (right) almost final configuration

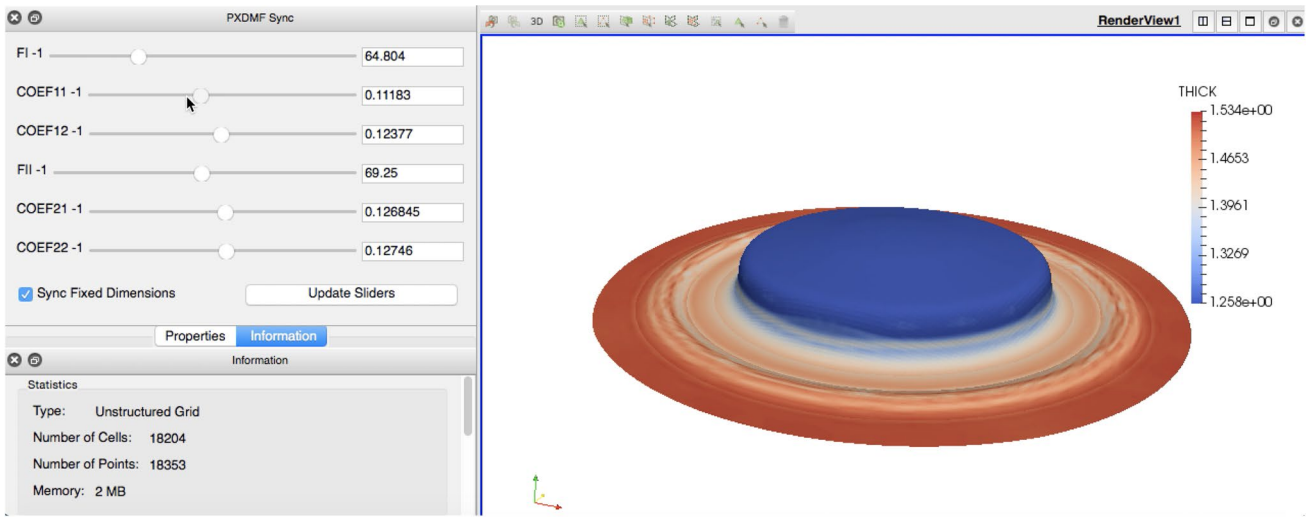


**Fig. 7** Parametric solution $\mathbf{u}^{II}(F^I, \xi^I, \mu^I, F^{II}, \xi^{II}, \mu^{II})$

This means that as considered in the POD by interpolation, one can express parametrically the coefficients $\alpha_j$, i.e. $\alpha_j(F^I, \xi^I, \mu^I)$.

Such a rationale becomes more general, and enables the solution of the second stage being expressed parametrically with respect to $\alpha_j$, $j = 1, ..., \mathtt{R}$, and the process triplet $(F^{II}, \xi^{II}, \mu^{II})$, i.e.

$$\mathbf{u}^{II}(\alpha_1, ..., \alpha_{\mathtt{R}}, F^{II}, \xi^{II}, \mu^{II}). \tag{26}$$

Other than gaining generality, because of the fact that Eq. 26 can be applied to any stage independently of the place that it occupies in the whole process, as soon as the reduced basis $\{\phi_1, \ldots, \phi_{\mathtt{R}}\}$ is able to express the incoming part, such a rationale enables also improving accuracy.

Within the fully physics-based rationale, the part resulting from the first stage, when enforcing the process parameters $(F^I, \xi^I, \mu^I)$, is expressed from

$$\mathbf{u}^I(F^I, \xi^I, \mu^I) \approx \alpha_1\phi_1 + \ldots + \alpha_{\mathtt{R}}\phi_{\mathtt{R}},$$

with $\alpha_j = \alpha_j(F^I, \xi^I, \mu^I)$, $j = 1, \ldots, \mathtt{R}$.

If these coefficients are then inserted into the second stage parametric solution, given by Eq. 26, the computed solution will correspond approximately with $\mathbf{u}^{II}(F^I, \xi^I, \mu^I, F^{II}, \xi^{II}, \mu^{II})$.

However, if the real part after the first stage, represented by $\hat{\mathbf{u}}^I$, differs from $\mathbf{u}^I$, i.e. $\|\hat{\mathbf{u}}^I - \mathbf{u}^I\| > \epsilon$, a fully physics-based approach will propagate such a deviation to the stages that follow.

The hybrid paradigm aims at correcting the intermediate solutions. If we assume that the reduced basis continues spanning the real solution we can express it from

$$\hat{\mathbf{u}}^I \approx \hat{\alpha}_1\phi_1 + ... + \hat{\alpha}_{\mathtt{R}}\phi_{\mathtt{R}}, \tag{27}$$

where $\hat{\alpha}_j = \hat{\mathbf{u}}^I \cdot \phi_j$.

The real state $\hat{\mathbf{u}}^I$ is rarely fully known, however, a partial knowledge suffices for determining the $\mathtt{R}$ parameters $\hat{\alpha}_j$. Few amount of data, scaling with $\mathtt{R}$, is in general sufficient.

Such a data-driven correction allows updating the prediction related to the second stage, $\hat{\mathbf{u}}^{II}(\hat{\alpha_1}, ..., \hat{\alpha}_{\mathtt{R}}, F^{II}, \xi^{II}, \mu^{II})$ instead of $\mathbf{u}^{II}(\alpha_1, ..., \alpha_{\mathtt{R}}, F^{II}, \xi^{II}, \mu^{II})$.

In the numerical example addressed in the present section, as soon as the solution of the second stage is updated, the solution at the end for $(F^{II}, \xi^{II}, \mu^{II})$ results the one depicted in Fig. 8(left) where the damage exceed the maximum acceptable value. Thus, the control proceeds by changing the clamping force $\tilde{F}^{II}$, action that leads to the solution depicted in Fig. 8(right) that exhibits an acceptable risk criterion.

## Conclusion

This paper revisited the main methodologies employed in the hybrid engineering and hybrid twins, main protagonists of the incipient performance based engineering, where the necessity of predicting very fast and very accurately, while reducing the amount of needed data, leaded to the hybridization of physics-based models, efficiently manipulated thanks to the use of advanced non-intrusive model order reduction techniques, enriched, in view of the collected data, by using low-cost physics-aware artificial intelligence techniques.

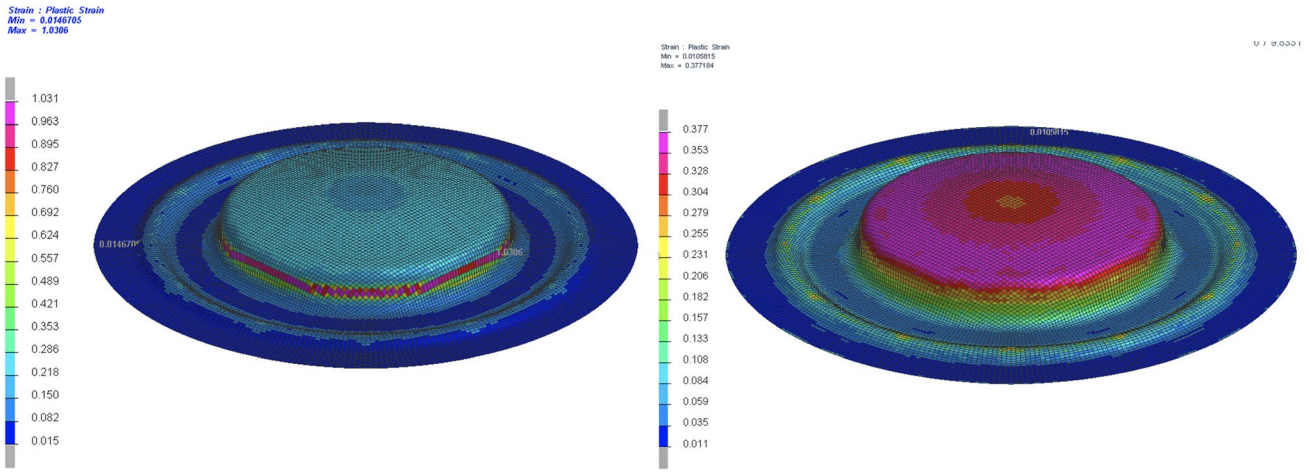


**Fig. 8** Updated solution: (left) $\hat{\mathbf{u}}^{II}(\hat{\alpha}_1, ..., \hat{\alpha}_R, F^{II}, \xi^{II}, \mu^{II})$ and (right) $\hat{\mathbf{u}}^{II}(\hat{\alpha}_1, ..., \hat{\alpha}_R, \tilde{F}^{II}, \xi^{II}, \mu^{II})$ (the color map is not the same in both figures)

**Acknowledgements** Authors acknowledge the support of ESI Group from their chairs at the University of Zaragoza and at Arts et Metiers Institute of Technology.

## Declarations

**Conflicts of interest** The authors declare that they have no conflict of interest.

## References

1. Aguado JV, Borzacchiello D, Kollepara KS, Chinesta F, Huerta A (2019) Tensor representation of on-linear models using cross approximations. Journal of Scientific Computing 81:22–47
2. Argerich C, Ibanez R, Barasinski A, Chinesta F (2019) Code2vect: An efficient heterogenous data classifier and nonlinear regression technique. C.R. Mecanique 347:754–761
3. Argerich C, Carazo A, Sainges O, Petiot E, Barasinski A, Piana M, Ratier L, Chinesta F (2020) Empowering Design Based on Hybrid Twin: Application to Acoustic Resonators. Designs 4:44. https://doi.org/10.3390/designs4040044
4. Borzacchiello D, Aguado JV, Chinesta F (2019) Non-intrusive sparse subspace learning for parametrized problems. Archives of Computational Methods in Engineering 26(2):303–326
5. Breiman L (2001) Random forests. Machine Learning 45:5–32
6. Brunton S, Proctor JL, Kutz N (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. PNAS 113(15):3932–3937
7. Casteran F, Delage K, Cassagnau P, Ibanez R, Argerich C, Chinesta F (2020) Application of Machine Learning tools for the improvement of reactive extrusion simulation. Macromolecular Materials and Engineering. https://doi.org/10.1002/mame.202000375
8. Chaturantabut S, Sorensen D (2010) Nonlinear model order reduction via discrete empirical interpolation. SIAM J. Sci. Comput. 32(5):2737–2764
9. Chinesta F, Leygue A, Bordeu F, Aguado JV, Cueto E, Gonzalez D, Alfaro I, Ammar A, Huerta A (2013) Parametric PGD based computational vademecum for efficient design, optimization and control. Archives of Computational Methods in Engineering 20(1):31–59
10. Chinesta F, Keunings R, Leygue A (2014) The proper generalized decomposition for advanced numerical simulations. A primer, Springerbriefs, Springer
11. Chinesta F, Huerta A, Rozza G, Willcox K (2015) Model order reduction. In: Stein E, de Borst R, Hughes T (eds) The encyclopedia of computational mechanics. Second Edition, John Wiley & Sons Ltd
12. Chinesta F, Cueto E, Abisset-Chavanne E, Duval JL, El Khaldi F (2020) Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. Archives of Computational Methods in Engineering 27:105–134
13. Frahi T, Chinesta F, Falco A, Badias A, Cueto E, Choi HY, Han M, Duval JL (2021) Empowering advanced driver-assistance systems from topological data analysis. Mathematics 9:634
14. Frahi T, Yun M, Argerich C, Falco A, Chinesta F (2020) Tape surfaces characterization with persistence images. AIMS Materials Science 7(4):364–380
15. Frahi T, Falco A, Vinh Mau B, Duval JL, Chinesta F (2021) Empowering advanced parametric modes clustering from topological data analysis. Appl. Sci. 11:6554
16. Ghanem R, Soize C, Mehrez L, Aitharaju V (2020) Probabilistic learning and updating of a digital twin for composite material systems. IJNME. https://doi.org/10.1002/nme.6430
17. Gonzalez D, Chinesta F, Cueto E (2019) Thermodynamically consistent data-driven computational mechanics. Continuum Mech. Thermodynamics 31:239–253
18. Gonzalez D, Chinesta F, Cueto E (2019) Learning corrections for hyper-elastic models from data. Frontiers in Materials - section Computational Materials Science, 6 https://www.frontiersin.org/article/10.3389/fmats.2019.00014
19. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
20. Grmela M, Ottinger HC (1997) Dynamics and thermodynamics of complex fluids. I. development of a general formalism. Phys. Rev. E 56:6620–6632
21. Hernandez Q, Badias A, Gonzalez D, Chinesta F, Cueto E (2021) Deep learning of thermodynamics-aware reduced-order models from data. Computer Methods in Applied Mechanics and Engineering 379:113763

22. Hernadez Q, Gonzalez D, Chinesta F, Cueto E (2021) Learning non-Markovian physics from data. Journal of Computational Physics, In press
23. Hinton GE, Zemel RS (1994) Autoencoders, minimum description length and Helmholtz free energy. Advances in neural information processing systems 6 (NISP 1993). Morgan-Kaufmann, 3-10
24. Ibanez R, Borzacchiello D, Aguado JV, Abisset-Chavanne E, Cueto E, Ladeveze P, Chinesta F (2017) Data-driven non-linear elasticity. Constitutive manifold construction and problem discretization. Computational Mechanics 60/5:813–826
25. Ibanez R, Abisset-Chavanne E, Ammar A, Gonzalez D, Cueto E, Huerta A, Duval JL, Chinesta F (2018) A multi-dimensional data-driven sparse identification technique: the sparse Proper Generalized Decomposition. Complexity, Article ID, p 5608286
26. Ibanez R, Abisset-Chavanne E, Gonzalez D, Duval JL, Cueto E, Chinesta F (2019) Hybrid Constitutive Modeling: Data-driven learning of corrections to plasticity models. International Journal of Material Forming 12:717–725
27. Ibanez R, Gilormini P, Cueto E, Chinesta F (2020) Numerical experiments on unsupervised manifold learning applied to mechanical modeling of materials and structures. CRAS 348(10–11):937–958
28. Kapteyn MG, Willcox KE (2020 ) From physics-based models to predictive digital twins via interpretable machine learning. arXiv:200411356v3
29. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Computer Methods in Applied Mechanics and Engineering 304:81–101
30. Kirkwood CW (2002) Decision tree primer. http://creativecommons.org/licenses/by-nc/3.0/
31. Ladeveze P (1999) Nonlinear computational structural mechanics. Springer Verlag, New Approaches and Non-Incremental Methods of Calculation
32. Ladeveze P, Neron D, Gerbaud P-W (2019) Data-driven computation for history-dependent materials. Comptes Rendus Mecanique 347/11:831–844
33. Latorre M, Montans FJ (2014) What-you-prescribe-is-what-you-get orthotropic hyperelasticity. Computational Mechanics 53(6):1279–1298
34. Moya B, Gonzalez D, Alfaro I, Chinesta F, Cueto E (2019) Learning slosh dynamics by means of data. Computational Mechanics 64:511–523
35. Moya B, Badias A, Alfaro I, Chinesta F, Cueto E (2020) Digital twins that learn and correct themselves. International Journal for Numerical Methods in Engineering. https://doi.org/10.1002/nme.6535
36. Muller M Information retrieval for music and motion. Springer-Verlag Berlin Heidelberg
37. Raissi M, Perdikaris P, Karniadakis GE (2017) Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations. arXiv:1711.10561
38. Raissi M, Perdikaris P, Karniadakis GE (2017) Physics informed deep learning (part II): data-driven discovery of nonlinear partial differential equations. arXiv:1711.10566
39. Reille A, Champaney V, Daim F, Tourbier Y, Hascoet N, Gonzalez D, Cueto E, Duval JL, Chinesta F (2021) Learning data-driven reduced elastic and inelastic models of spot-welded patches. Mechanics & Industry 22:32
40. Sancarlos A, Cameron M, Abel A, Cueto E, Duval JL, Chinesta F (2020) From ROM of electrochemistry to AI-based battery digital and hybrid twin. Archives of Computational Methods in Engineering. https://doi.org/10.1007/s11831-020-09404-6
41. Sancarlos A, Cameron M, Le Peuvedic JM, Groulier J, Duval JL, Cueto E, Chinesta F (2021) Learning stable reduced-order models for hybrid twins. Data Centric Engineering 2:E10. https://doi.org/10.1017/dce.2021.16
42. Sancarlos A, Champaney V, Duval JL, Cueto E, Chinesta F (2021) PGD-based advanced nonlinear multiparametric regressions for constructing metamodels at the scarce-data limit. arxiv:2103.05358
43. Senin P (2008) Dynamic time warping algorithm review. Technical report
44. Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58:1:267–288
45. Torregrosa S, Champaney V, Ammar A, Hebert V, Chinesta F (2022) Surrogate Parametric Metamodel based on Optimal Transport. Mathematics and Computers in Simulation 194:36–63
46. Torregrosa S, Champaney V, Ammar A, Herbert V, Chinesta F (2007) Hybrid twins based on optimal transport. Computers and Mathematics with Applications, Submitted
47. Tuegel EJ, Ingraffea AR, Eason TG, Spottswood SM (2011) Reengineering aircraft structural life prediction using a digital twin. International Journal of Aerospace Engineering, 154798
48. Yun M, Argerich C, Cueto E, Duval JL, Chinesta E (2020) . Nonlinear regression operating on microstructures described from Topological Data Analysis for the real-time prediction of effective properties. Materials 13:10:2335