

Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data

Francisco Chinesta, Elías G. Cueto, Emmanuelle Abisset-Chavanne, Jean Louis Duval, Fouad El Khaldi

▶ **To cite this version:**

Francisco Chinesta, Elías G. Cueto, Emmanuelle Abisset-Chavanne, Jean Louis Duval, Fouad El Khaldi. Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data. Archives of Computational Methods in Engineering, 2019, 10.1007/s11831-018-9301-4 . hal-02292413

HAL Id: hal-02292413
https://hal.science/hal-02292413

Submitted on 19 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data

Francisco Chinesta[1] · Elias Cueto[2] · Emmanuelle Abisset-Chavanne[3] · Jean Louis Duval[4] · Fouad El Khaldi[4]

Abstract
Engineering is evolving in the same way than society is doing. Nowadays, data is acquiring a prominence never imagined. In the past, in the domain of materials, processes and structures, testing machines allowed extract data that served in turn to calibrate state-of-the-art models. Some calibration procedures were even integrated within these testing machines. Thus, once the model had been calibrated, computer simulation takes place. However, data can offer much more than a simple state-of-the-art model calibration, and not only from its simple statistical analysis, but from the modeling and simulation viewpoints. This gives rise to the the family of so-called *twins*: the virtual, the digital and the hybrid twins. Moreover, as discussed in the present paper, not only data serve to enrich physically-based models. These could allow us to perform a tremendous leap forward, by replacing big-data-based habits by the incipient smart-data paradigm.

1 Introduction

As models involved in science and engineering become too complex, their analytical solution is often compromised. On the other hand, computers are able to perform very efficiently only elementary operations. Consequently, it is necessary to transform complex mathematical objects (derivatives, …) into simpler objects, i.e., elementary operations. At the same time, it is necessary to reduce the number of points and time instants at which the solution of the model is evaluated, by replacing the continuum by a discrete system, treatable by digital computers. Such a procedure is known as numerical simulation and constitutes one of the three pillars of twentieth century engineering—modeling and experiments being the other two pillars—. This age has been coined as the *third paradigm* of science [1].

In the previous (third) industrial revolution, "virtual twins" (emulating a physical system by one, or more, mathematical models to describe its complex behavior) were major protagonists.[1] Nowadays, numerical simulation is present in most scientific fields and engineering domains, making accurate designs and virtual evaluation of systems responses possible—drastically cutting the number of experimental tests.

The usual numerical model in engineering practice (which we will denote here as virtual twin) is something static. This kind of (finite element, finite volume, finite difference) models is nowadays ubiquitous in the design of complex engineering systems and their components. We say that they are static because they are not expected to be

✉ Francisco Chinesta
Francisco.Chinesta@ensam.eu

Elias Cueto
ecueto@unizar.es

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

Jean Louis Duval
Jean-Louis.Duval@esi-group.com

Fouad El Khaldi
Fouad.El-Khaldi@esi-group.com

[1] ENSAM ParisTech, 151 Boulevard de l'Hôpital, 75013 Paris, France

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

[3] ESI Group Chair, Ecole Centrale de Nantes, 1 rue de la Noe, 44300 Nantes, France

[4] ESI Group, 3bis rue Saarinen, 94528 Rungis Cedex, France

[1] There seems to be no consensus on the definition of the concepts of virtual, digital and hybrid twins. In this paper we suggest one possible distinction, that seems feasible, attending to their respective characteristics. It is not the sole possibility, of course, nor do we pretend to create any controversy on it.

continuously fed by data so as to assimilate them. This would be what is today understood as a Dynamic Data-Driven Application System (DDDAS) [2]. The characteristic time of standard engineering simulation strategies can not accommodate the stringent real-time constraints posed by DDDAS, specially for control purposes. Real-time simulation for control is typically ensured by techniques based on the use of ad hoc—or black box—models of the system (in the sense that they relate some inputs to some outputs, encapsulated into a transfer function). This adapted representation of the system allows proceeding in real-time. However, it becomes too coarse when compared with rich, high fidelity simulations, such as the ones performed using, for example, Finite Element techniques.

Although science was preeminently data-based at the early years (think of Tycho Brahe, for instance), it was at the end of the 20th century that data irrupted massively in most scientific fields, and in particular in the one we are specially interested in: engineering. For many years data have been widely incorporated to usual practice in many disciplines where models were more scarce or less reliable—with respect to engineering sciences—. Thus, massive data were classified, visualized (despite its frequent multi-dimensional nature), curated, analyzed, …thanks to the powerful techniques recently developed in the wide areas of artificial intelligence and machine learning. When correlations are removed from data, a certain simplicity emerges from the apparent complexity, as proved by advanced nonlinear dimensionality reduction techniques based on manifold learning. Moreover, a number of techniques were proposed to establish the relations between outputs of interest and certain inputs, assumed to be sufficient to explain and infer the outputs. These are the so-called "model learners", based on the use of linear and nonlinear regressions, decision trees, random forests, neural networks—inevitably linked to deep-learning techniques—, among many others.

The solution of physically-based models, very well established and largely validated in the last century, was partially or totally replaced by these data-based models, due to their computational complexity. This is especially true for applications requiring real-time feedback. Thus, massively collected and adequately curated data, as just discussed, provided interpretation keys to advise on an imminent fortuitous event. This makes possible improved data-based predictive maintenance, efficient inspection and control, …that is, allows for real-time decision making. The price to pay is an as rich as possible learning stage. This takes considerable time and efforts, as the establishment of validated models took in the previous engineering revolution.

Important success was reported, many possibilities imagined, …justifying the exponential increase in popularity of these "digital twins". There has been a fast development of data-driven models for representing a system with all its richness while ensuring real-time enquiries to its governing model. However, replacing the rich history of engineering sciences (that proved their potential during more than a century with spectacular successes) led to feelings of bitterness and of waste of acquired knowledge.

This new incipient engineering consists of "virtual twins", that operate offline in the design stage, and their digital counterparts, based on data, taking over in online operations. However, the domain of applicability of the last, even if they are superior in what concerns their rate of response, continues to be narrower. A combination of both, the "virtual" and "digital" twins seems to be the most appealing solution. However, prior to combining both, a major difficulty must be solved: the real-time solution of physically-based models.

All the just introduced problems can not be overcome by simply employing more powerful computers—in other words, by employing modern supercomputing facilities—. Even when this is a valuable route, it strongly limits the accessibility to the appropriate simulation infrastructure. This is true also in what concerns to its integration in deployed platforms. Recent history has proved that this is a prohibitive factor for small and medium-sized companies. An effort must be paid towards the to democratization of simulation.

Again, it was at the end of the past century and the beginning of the 21st century, that major scientific accomplishments in theoretical and applied mathematics, applied mechanics, and computer sciences contributed to new modeling and simulation procedures. Model Order Reduction (MOR) techniques were one of these major achievements [3]. These techniques do not proceed by simplifying the model, models continue to be well established and validated descriptions of the physics at hand. Instead, they rely on an adequate approximation of the solution that allows simplifying the solution procedure without any sacrifice on the model solution accuracy, in view of accommodating real-time constraints.

A feasible alternative within the MOR framework consists of extracting offline the most significant modes involved in the model solution, and then projecting the solution of similar problems in that reduced space. Consequently, a discrete problem of very small size must be solved at each iteration or time step. Thus, MOR-based discretization techniques provide significant savings in computing-time. Another MOR-based route consists of computing offline, using all the needed computational resources and computing time, a parametric solution that contains the solution of all possible scenarios. This parametric solution can then be online particularized to any

scenario using deployed computational facilities, including tablets or even smartphones. It allows then to perform efficient simulation, optimization, inverse analysis, uncertainty propagation and simulation-based control, all under real-time constraints. Such a solution has been demonstrated on many applications where the Proper Generalized Decomposition (PGD) method is used [4–9].

The next generation of twins was born, the so-called "hybrid twin^{TM}", that combines physically-based models within a MOR framework (for accommodating real-time feedback) and data-science.

On one hand, real-time solution of physically-based models allows us to assimilate data collected from physical sensors, to calibrate them. Therefore, it also exhibits predictive capabilities to anticipate actions. Thus, simulation-based control was made possible, and successfully implemented in many applications, often by using deployed computing devices (e.g., Programmable Logic Controllers). Despite an initial euphoric and jubilant period in which high-fidelity models were exploited in almost real-time by using standard computing platforms, unexpected difficulties appeared as soon as they were integrated into data-driven application systems.

Significant deviations between the predicted and observed responses have been detected, nevertheless, by following this approach. The origin of these deviations between predictions and measurements can be attributed to inaccuracy in the employed models, in parameters or in their time evolution. These often continue to be crude descriptions of the actual systems. Attacking this ignorance can done by developing on-the-fly data-driven models that could eventually correct this deviation between data and model predictions.

Indeed, a DDDAS consists of three main ingredients: (1) a simulation core able to solve complex mathematical problems representing physical models under real-time constraints [10]; (2) advanced strategies able to proceed with data-assimilation, data-curation and data-driven modeling; and (3) a mechanism to online adapt the model to evolving environments (control). The Hybrid Twin^{TM} [11] embraces these three functionalities into a new paradigm within simulation-based engineering sciences (SBES).

2 From Virtual to Hybrid Twins

A given physical system is characterized by a number of continuous or discrete variables. In general, to manipulate these variables in a computer, continuous variables are discretized, i.e., more than looking for those variable at any point, it is assumed that variables at any point can be expressed from the ones existing in some particular locations (the nodes, if we employ the finite element terminology) by using adequate interpolations.

In what follows the discrete form of the variables defining the system state at time t is denoted by $\mathbf{X}(t)$. As just indicated, they could include, depending on the considered physics, nodal temperatures, velocities, displacements, stresses, etc.

The system evolution is described by its state $\mathbf{X}(t)$, evolving from its initial state at the initial time $t = t_0 = 0$, denoted by $\mathbf{X}_0$. Numerical models based on well established physics allow making this prediction of the system state at time t from the knowledge of it at the initial time t_0, by integrating its rate of change (coming from the physical laws adequately discretized) given by $\dot{\mathbf{X}}(\tau)$ at $\tau \in (0, t]$.

This contribution will be expressed by $\dot{\mathbf{X}}(t;\boldsymbol{\mu}) = \mathbf{A}(\mathbf{X}, t;\boldsymbol{\mu})$—we emphasize its parametric form—, where $\boldsymbol{\mu}$ represents the set of involved parameters that should be identified offline or online.

Remark 1 In the previous expression the semicolon $(\cdot;\cdot)$ makes a distinction between the coordinates before the semicolon—in this case, time—and the model parameters after it—here, $\boldsymbol{\mu}$.

Thus, if we assume a model to accurately represent the subjacent physics involved in the system, predictions are easily performed by integrating $\mathbf{A}(\mathbf{X}, t;\boldsymbol{\mu})$. Here, if real-time feedbacks are needed, standard integration (based on the use of well experienced numerical techniques like finite elements, finite differences, finite volumes, spectral methods, meshless (or meshfree) techniques, …) of the dynamical system expressed by $\mathbf{A}(\mathbf{X}, t;\mu)$, turns out to be unsatisfactory. As previously discussed, the employ of model reduction techniques opened new routes in this sense. In particular, the Proper Generalized Decomposition—PGD—precomputes (offline) the parametric solution, thus making possible to accommodate real-time constraints.

When model calibration is performed online, model parameters $\boldsymbol{\mu}$ are calculated by enforcing that the associated model prediction fits as much as possible to the experimental measurements, at least at the measurement points. In the context of process or system control, external actions can be applied to drive the model towards the given target. Thus, the state rate of change (if we neglect noise for the moment) is composed by two terms,

$$\dot{\mathbf{X}}(t;\boldsymbol{\mu}) = \mathbf{A}(\mathbf{X}, t;\boldsymbol{\mu}) + \mathbf{C}(t), \tag{1}$$

that expresses the physical and forced (external goal-oriented actions) contributions, $\mathbf{A}$ and $\mathbf{C}$, respectively.

Remark 2 In general, control actions, here represented by the term $\mathbf{C}$ could depend on measures and/or on the inferred model parameters, but here, and without loss of

generality, we only indicate explicitly its dependence on time.

2.1 Model Updating

When models represent the associated physics poorly, a non negligible deviation between their predictions and the actual evolution of the system, acquired from collected data, is expected. This deviation is expected to be biased, because it represents the modeler's ignorance on the subjacent physics. The unbiased deviation contribution is associated to modeling or measurement noise and is easily addressed by using adequate filters. However, biased deviations express hidden physics and required a particular treatment, that is, their online modeling by assimilating collected data.

Indeed, the deviation (gap between the model prediction $\mathbf{X}(t;\boldsymbol{\mu})$ and measurements $\mathbf{X}^{\exp}(t)$) when considering the optimal choice of the model parameters $\boldsymbol{\mu}$, and, more precisely its time derivative, should be used for the online construction (under the already mentioned severe real-time constraints) of the so-called data-based correction model. This correction, also referred as deviation model, is here denoted by $\mathbf{B}(\mathbf{X},t)$. Even if, in what follows, the presence of unbiased noise is ignored, its inclusion is straightforward.

Thus, the fundamental equation governing a hybrid twins writes

$$\dot{\mathbf{X}}(t;\boldsymbol{\mu}) = \mathbf{A}(\mathbf{X},t;\boldsymbol{\mu}) + \mathbf{B}(\mathbf{X},t) + \mathbf{C}(t) + \mathbf{R}(t), \tag{2}$$

expressing that the rate of change of the system state at time t contains four main contributions:

1. the model contribution, whose rate of change related to the model parameters $\boldsymbol{\mu}$ reads $\mathbf{A}(\mathbf{X},t;\boldsymbol{\mu})$. MOR is crucial at this point to ensure real-time feedback;
2. a data-based model $\mathbf{B}(\mathbf{X},t)$ describing the gap between prediction and measurement;
3. external actions $\mathbf{C}(t)$ introduced into the system dynamics in order to drive the model solution towards the desired target. It also includes any other kind of decision based on the collected and analyzed data;
4. the unbiased noise $\mathbf{R}(t)$, that has been traditionally addressed using appropriate filters [12]. This terms also includes external actions for which there is no possible prediction. Typically, human intervention on the system.

Here we have omitted a very important distinction, the necessity of collecting appropriate data with different aims: (1) to calibrate the considered physically-based model, assumed to represent the first-order contribution to predictions and for explicative purposes; (2) to construct on-the-fly the data-driven model update; and (3) for decision-making proposes (control) by using data-analytics on the collected data.

It is also worth noting that the better locations and frequency of acquisition for collecting data could differ, given the volume of data to treat and data-assimilation rate, depending on the purpose: calibration, modeling and control. In the present framework, the model could help to infer the *smartest* data to acquire, and when and where they should be collected. Thus, Big Data could be replaced by Smart Data in the framework of a new multi-scale data science and theory of information, bridging the gap between data (microscopic), information (mesoscopic) and knowledge (macroscopic).

The construction of the data-based model deserves some additional comments:

1. Deviations inform us about the model possibly becoming inaccurate. In our approach, model updating is based on the deviation model, and then, it is added to the first-order model when it exists. Other authors suggested to update the model itself. Thus for example in [13], the authors proceed by perturbing in a random way the discrete matrix $\mathbf{A}$, that results in $\tilde{\mathbf{A}}$, within a stochastic framework.
2. In some cases the first order, physically-based model, $\mathbf{A}$, does not exist, or simply it is ignored as was the case in most digital twins, motivated by difficulties related to its real-time solution, to its accuracy, etc …In that case, the model consists of a unique contribution, the data-based model [14, 15]. In this case, when constructed from scratch, many data are required to reach a sufficient accuracy. However, when the data-driven model is only expected to fill the gap between the first-order model predictions and the measurements, the higher is the model accuracy, the smaller the data-driven contribution, implying that the required volume of data significantly decreases. It is worth mentioning that collecting data and processing them is expensive, and could compromise the real-time constraints.
3. The recent exponential growing of machine learning techniques (data-mining, deep-learning, manifold learning, linear and nonlinear regression techniques …to cite but a few) makes it possible to construct on-the-fly such a data-based deviation model;
4. Another possibility consists of expressing the deviation within a parametric form within the PGD framework. To that end, a sparse-PGD is constructed—here viewed as an advanced powerful nonlinear regression technique [16]—, operating on the deviations. These deviations are the difference between the physically-based prediction and the measurements. The main advantage of this procedure is that the parametric expression of the deviation can be added to the

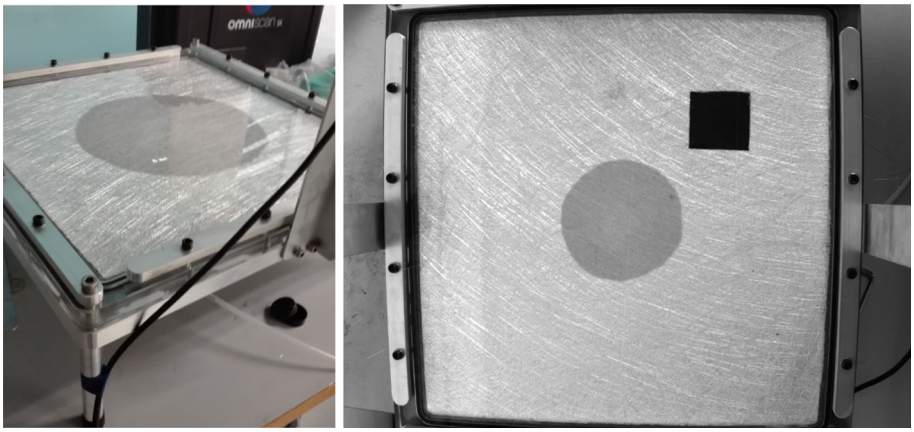


Fig. 1 Square mould filled with an isotropic reinforcement and containing an impermeable square insert (black small square)

expression of the model based on the known physics, **A**, that was already expressed using the same format (parametric PGD separated representation).

Thus, the resulting solution contains some modes coming from the discretization of the equations representing the known physics, while the remaining ones are associated to the detected deviations. In any case, both together represent the actual system that contains hidden physical mechanisms, more complex that the ones retained in the first-order model, pragmatically captured even when ignoring its real nature.

If the real solution evolves in a manifold, its projection on the manifold defined by the physical model, **A**, allows computing the best choice for the involved parameters, i.e., $\boldsymbol{\mu}$ (calibration). The orthogonal complement represents the deviation model. All of them, the real, the physical and the deviation models can be cast into a parametric PGD separated representation form.

Remark 3 In the previous expression, Eq. (2), the data-based contribution $\mathbf{B}(\mathbf{X}, t)$ justifies the "hybrid" appellation, because the model is composed of two contributions, one coming from well established and validated physics, the other based on data. This double nature makes the difference between usual digital twins and their hybrid counterparts.

Remark 4 When enriching a dynamical system with a data-based contribution, before reaching a sufficient accuracy, a stable system can become unstable, thus compromising long-time predictions. In this case the control term could encompass a numerical stabilization to ensure that the enriched dynamical systems remains stable.

Remark 5 Deep learning, based on the use of deep neural networks, allows impressive accomplishments. However, it generates nowadays a certain frustration in a scientific community that for centuries tried to explain reality through models. That aim is almost lost when using deep learning. Even if many efforts are being paid with the purpose of explaining these machine learning techniques, today their impressive performance is not fully understood. However, within the hybrid twin rationale, things become less uncomfortable, since these techniques, whose predictions are difficult to explain, are being used to model a physics that escapes to our understanding, what we have called ignorance.

2.2 Illustrating Hybrid Twin Features

Hereafter, the construction and functioning of a simple hybrid twin of resin transfer moulding (RTM) processes is presented. For the sake of simplicity, realistic complexity has been sacrificed in favor of description simplicity. The problem consists in filling a square mould from its central point. An impermeable square insert is placed in the right-upper zone in order to break the solution symmetry. The experimental device is depicted in Fig. 1.

In what follows, the construction and the use of the two first contributions of the hybrid twin—the physical (**A**) and the data-based (**B**) models—is described through four steps:

- First, the parametric solution of the flow problem related to the mould filling process—where the chosen parameter is the preform permeability κ—is carried out by coupling the commercial software PAM-RTM (ESI Group, France) and the PGD constructor. In particular, we use a non-intrusive formulation based on the sparse subspace learning (which we will refer to as SSL-PGD in what follows) or its sparse counterpart sPGD (both reviewed in Sect. 3). Thus, every field (pressure, velocity, filling factor, …) will be accessible in a parametric way, that is, for any possible value of the permeability κ. Here, without loss of generality, it is assumed to be constant and isotropic in the whole

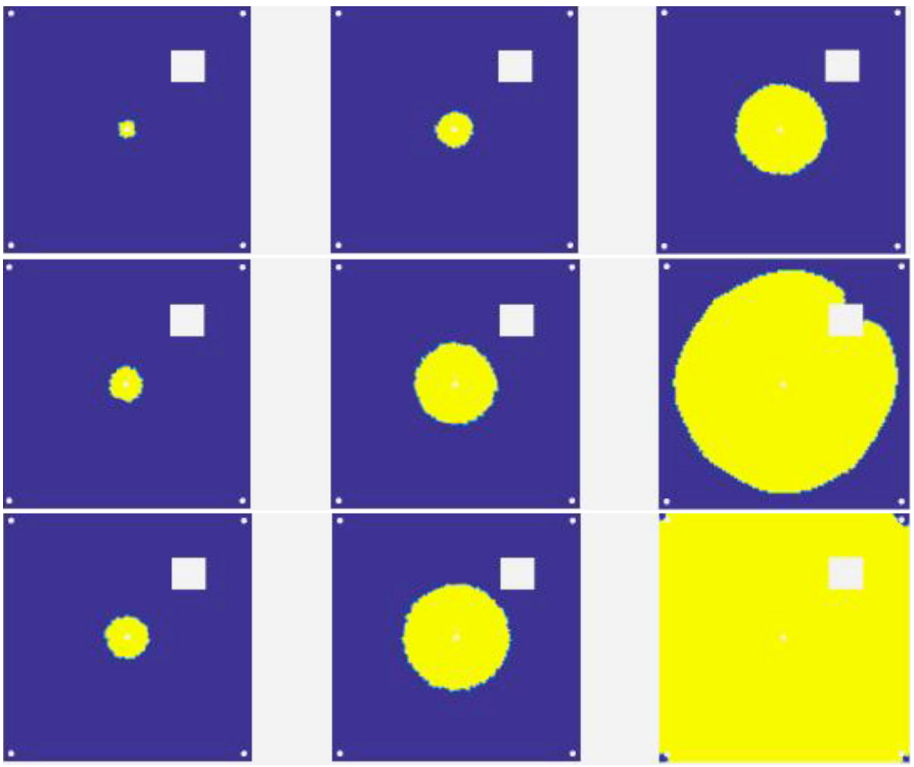


Fig. 2 Particularizing the PGD-based mould filling solution for three different permeabilities (low at the left, intermediate at the center and high at the right) at three different time steps (from top to bottom)

preform. As soon as the parametric solution has been computed offline, it can be particularized online almost in real-time. Figure 2 depicts the flow front at different instants and for three different permeabilities.

- Second, the permeability is identified by comparing the actual flow front—recorded by a camera—with the ones obtained by using different permeabilities. The reinforcement permeability is identified as the one that, inserted into the parametric model, allows the best fit between the predicted flow front position and the recorded one at different filling times. Once permeability has been determined, the simulated filling process agrees in minute to the one experimentally observed, as revealed by Fig. 3.
- Permeability is thus identified at the beginning of the filling procedure. However, the system ignores that the permeability in the neighborhood of the mould wall is lower than the just identified one. If this is the case, the model represented by $\mathbf{A}(t;\kappa)$ will significantly deviate from the measurements when the flow reaches the regions of reduced permeability. Figure 4 compares the simulation with the experimental recording of the modified permeability case. Note how, at the beginning, the simulation is in perfect agreement with the recording. But as soon as the flow front reaches the region with modified, lower permeability, important errors are noticed.
- Finally, by using dictionary learning or by constructing a PGD form of the correction, the deviation can be perfectly represented by the data-based contribution $\mathbf{B}(\mathbf{X},t)$, as illustrated in Fig. 5. This ensures the model predictability all along the filling process.

Through this simple example we would like to highlight how a hybrid twin is able to detect discrepancies with respect to the built-in model, and to correct them on the fly. Let us review the main difficulties associated to the practical implementation of this concept.

2.3 Implied Methodological Needs

As previously discussed the most complete member of the twin family involves many different methodologies that are revisited in the present paper, in particular:

1. Real-time simulation based on Model Order Reduction;
2. Real-time calibration;
3. Real-time data-assimilation and data completion;
4. Real-time data-analytics;
5. Real-time data-driven modeling.

The previous requirements will be addressed in the next section by using advanced model order reduction techniques for solving state-of-the-art physical models under stringent real-time constraints. Then, in Sect. 4 different

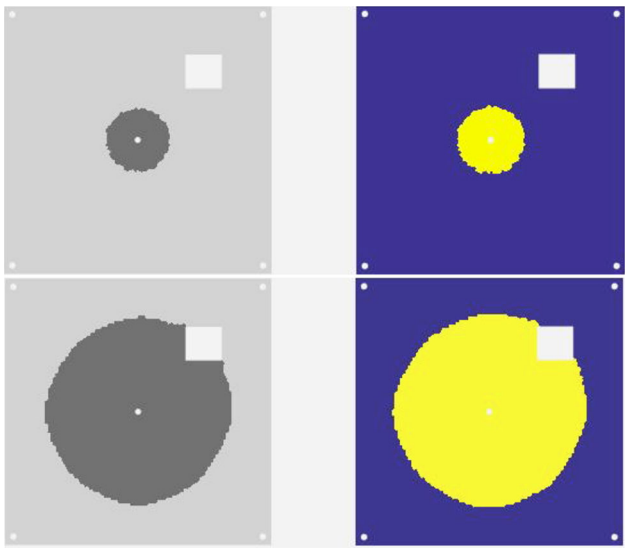


Fig. 3 Identifying the fibrous medium permeability and comparing predicted (right) and measured flow front (left) at two different time steps (top and bottom)

methodologies based on data-science will be described for addressing data-driven modeling.

3 Methods Based on Model Order Reduction with Special Emphasis on the Proper Generalized Decomposition

When looking for an approximation of the solution $u(\boldsymbol{x}, t)$ of a given PDE, here assumed scalar and linear without loss of generality, the standard finite element method considers the approximation

$$u(\boldsymbol{x}, t) = \sum_{i=1}^{\mathtt{N}} U_i(t) N_i(\boldsymbol{x}), \tag{3}$$

where U_i denotes the value of the unknown field at node i and $N_i(\boldsymbol{x})$ represents the its associated shape function. Here, $\mathtt{N}$ refers to the number of nodes considered to approximate the field in the domain Ω where the physical 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 [3], 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 are specific for the problem at hand and similar ones, but precisely because of this, they are much less numerous, thus minimizing the final number of degrees of freedom.

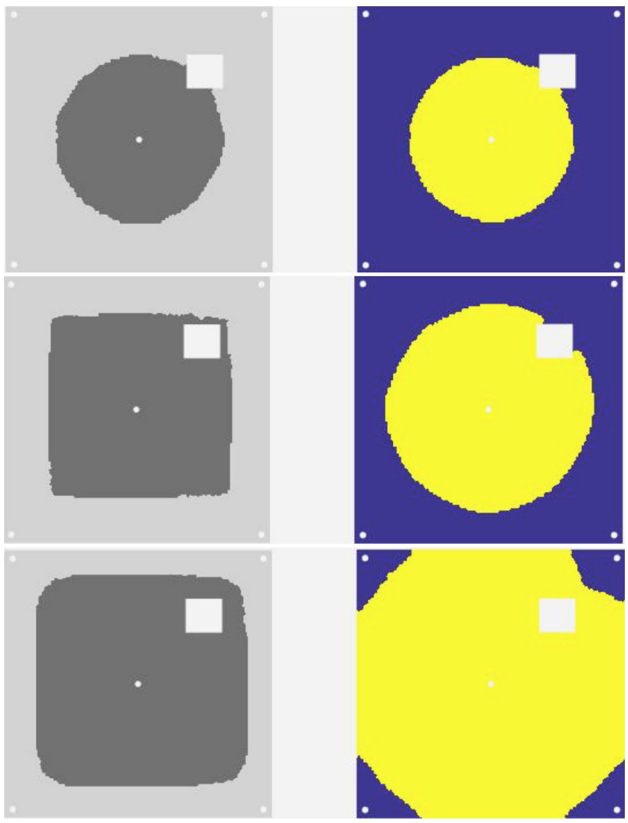


Fig. 4 Introducing a permeability reduction in the mould wall neighborhood in absence of data-based deviation model

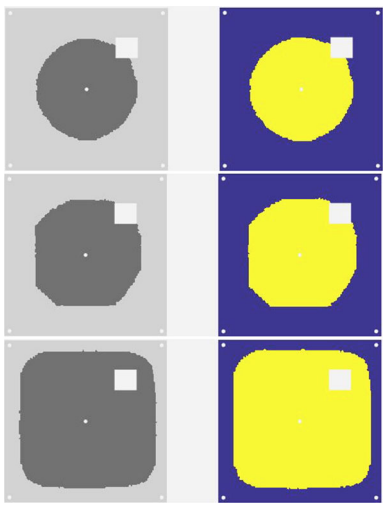


Fig. 5 Introducing a permeability reduction in the mould wall neighborhood while activating the data-based deviation model correction

Thus, by projecting the solution $u(\boldsymbol{x},t)$ onto the reduced basis composed by $\{\phi_1(\boldsymbol{x}),\ldots,\phi_{\mathtt{R}}(\boldsymbol{x})\}$, according to

$$u(\boldsymbol{x},t) \approx \sum_{i=1}^{\mathtt{R}} \xi_i(t)\phi_i(\boldsymbol{x}), \tag{4}$$

the resulting problem will now require the solution of a linear system of equations of size R, instead of size 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 [17, 18].

Equations (3) or (4) involve a finite sum of products composed by time-dependent coefficients multiplied by space functions. These space function are well-known finite element shape functions when no prior knowledge about the structure of the problem exists. Or can be substituted by a series of modes extracted by applying POD, if solutions of similar problems are available (i.e., snapshots of similar systems). 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 [19]. Thus, the resulting approximation reads

$$u(\boldsymbol{x},t) \approx \sum_{i=1}^{\mathtt{M}} T_i(t)X_i(\boldsymbol{x}). \tag{5}$$

Since the pairs of space and time functions in Eq. (5) 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, such as, for instance [7] or [8] and the references therein.

The final approximation, Eq. (5), will require the solution of about M problems, with $\mathtt{M} \ll \mathtt{N}$ and $\mathtt{M} \sim \mathtt{R}$. Usually the actual number will be slightly bigger than that. This is due to the nonlinearity induced by separated representations but also to the structure itself of the separated constructor. To compute the space functions $X_i(\boldsymbol{x})$ will require, at each iteration, to solve problems involving the spatial coordinates (in general three-dimensional, whose associated discrete systems are of size N) and also some M one-dimensional problems to calculate the time functions $T_i(t)$. The CPU cost of the solution of 1D problems is negligible, if compared to the solution of 3D problems. Thus, the resulting computational complexity reduces drastically, and will scale roughly with M instead of P (P being the number of time-steps employed in the time domain discretization).

Degenerate geometries (beams, plates, shells, layered domains such as composite materials) are specially well suited for a space domain separation [20–23]. If the domain Ω 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

$$u(x,y,z) \approx \sum_{i=1}^{\mathtt{M}} X_i(x)Y_i(y)Z_i(z), \tag{6}$$

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,

$$u(x,y,z) \approx \sum_{i=1}^{\mathtt{M}} X_i(x,y)Z_i(z), \tag{7}$$

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. 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. 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 [5, 24].

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

$$u(\boldsymbol{x},t,\mu_1,\ldots,\mu_{\mathtt{Q}}) \approx \sum_{i=1}^{\mathtt{M}} X_i(\boldsymbol{x})T_i(t)\prod_{j=1}^{\mathtt{Q}} M_i^j(\mu_j). \tag{8}$$

The use of a separated representation allows circumventing the combinatorial explosion. The solution of a sequence of low-dimensional problems allows calculating the parametric solution that can be viewed as a chart, abacus or vademecum—or, simply, as a high-dimensional response surface—, to be used online in a variety of applications.

3.1 The Standard, Intrusive, PGD Constructor

For the sake of completeness, we start addressing the original, intrusive, version of the PGD-based parametric solver [7] before considering its non-intrusive counterparts, that will be discussed in the following sections. For that purpose, we consider the parametric heat transfer equation

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0, \tag{9}$$

with homogeneous initial and boundary conditions. Here, $(\boldsymbol{x}, t, k) \in \Omega_x \times \Omega_t \times \Omega_k$. A completely new approach to the problem arises by simply considering the conductivity k as a new coordinate, which will be defined within some interval of interest Ω_k.

This new approach, instead of sampling the solution space for given values of the conductivity, consist in solving a new, more general problem. This new problem will be obtained after extending the weighted residual form related to Eq. (9),

$$\int_{\Omega_x \times \Omega_t \times \Omega_k} u^* \left(\frac{\partial u}{\partial t} - k\Delta u - f \right) d\boldsymbol{x}\, dt\, dk \;=\; 0. \tag{10}$$

If we look for a PGD approximation to the solution, it will look like

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

In other words, at iteration $n < \mathtt{M}$ the solution $u^n(\boldsymbol{x}, t, k)$ will be approximated by

$$u^n(\boldsymbol{x}, t, k) = \sum_{i=1}^{n} X_i(\boldsymbol{x}) T_i(t) K_i(k),$$

so that an improvement of this approximation, $u^{n+1}(\boldsymbol{x}, t, k)$, will be

$$u^{n+1}(\boldsymbol{x}, t, k) = u^n(\boldsymbol{x}, t, k) + X_{n+1}(\boldsymbol{x}) T_{n+1}(t) K_{n+1}(k). \tag{11}$$

The test function $u^\star$ for this extended weak form, Eq. (10), will therefore be given by

$$\begin{aligned} u^\star(\boldsymbol{x}, t, k) = {} & X^\star(\boldsymbol{x}) T_{n+1}(t) K_{n+1}(k) + X_{n+1}(\boldsymbol{x}) T^\star(t) K_{n+1}(k) \\ & + X_{n+1}(\boldsymbol{x}) T_{n+1}(t) K^\star(k). \end{aligned} \tag{12}$$

As usual, trial and test functions, Eqs. (11) and (12) respectively, are substituted into the weak form, Eq. (10). After an appropriate linearization, finite element approximations to functions $X_{n+1}(\boldsymbol{x})$, $T_{n+1}(t)$ and $K_{n+1}(k)$ are found. The simplest linearization strategy is the alternated directions, fixed point algorithm. It proceeds through the following steps (the interested reader can refer to [7] for more details, or to [8] for a thorough description of a Matlab code):

- Arbitrarily initialize at the first iteration $T^0_{n+1}(t)$ and $K^0_{n+1}(k)$.
- With $T^{p-1}_{n+1}(t)$ and K^{p-1}_{n+1} given at the previous, $p-1$, iteration of the non linear solver, all the integrals in $\Omega_t \times \Omega_k$ are computed, leading to a boundary value problem for $X^p_{n+1}(\boldsymbol{x})$.
- With $X^p_{n+1}(\boldsymbol{x})$ just computed and K^{p-1}_{n+1} given at the previous iteration of the nonlinear solver, all the integrals in $\Omega_x \times \Omega_k$ are computed, leading to an one-dimensional initial value problem for $T^p_{n+1}(t)$.
- With $X^p_{n+1}(\boldsymbol{x})$ and T^p_{n+1} just updated, all the integrals in $\Omega_x \times \Omega_t$ are performed, leading to an algebraic problem for $K^p_{n+1}(k)$.

3.2 Non-intrusive PGD Constructors

To circumvent the intrusivity of standard PGD algorithms so as to be able to construct parametric solutions by using commercial simulation softwares, two efficient procedures have been proposed that showed promise in a variety of case studies:

3.2.1 Sparse Subspace Learning—SSL

We consider the general case in which a transient parametric solution is searched. For the sake of notational simplicity, we assume that only one parameter is involved in the model, $\mu \in [\mu_{\min}, \mu_{\max}]$. The generalization to several, potentially many parameters is straightforward. The parametric solution $u(\boldsymbol{x}, t, \mu)$ is searched in the separated form

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

to circumvent the curse of dimensionality when the number of parameters increases. In this expression both functions involved in the finite sum representation, $X_i(\boldsymbol{x}, t)$ and $M_i(\mu)$, are a priori unknown.

SSL consists first in choosing a hierarchical basis of the parametric domain [25]. 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, μ_1^0 and μ_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 $\mu_2^0 = \mu_{\max}$ ($\Omega_\mu = [\mu_{\mathtt{min}}, \mu_{\mathtt{max}}]$). 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

$$\xi_1^0(\mu) = 1 - \frac{\mu - \mu_1^0}{\mu_2^0 - \mu_1^0},$$

and

$$\xi_1^0(\mu) = \frac{\mu - \mu_1^0}{\mu_2^0 - \mu_1^0},$$

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 $j = 0$, μ_1^0 and μ_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

$$u^1(\boldsymbol{x}, t, \mu) = u^0(\boldsymbol{x}, t, \mu) + \tilde{u}_1^1(\boldsymbol{x}, t)\xi_1^1(\mu). \tag{13}$$

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. (13), 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, that will be described later.

When the model involves more parameters, e.g., μ and η, the hierarchical 2D basis, defined in the parametric space (μ, η) is composed by the cartesian product of the collocations points and the tensor product of the approximation bases $\xi_i^0(\mu)$ and $\varphi_j^0(\eta)$.

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: (μ_1^0, η_1^1), (μ_2^0, η_1^1) and (μ_1^1, η_1^0), (μ_1^1, η_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 (μ_1^1, η_1^1) and its associated interpolation function is in fact a term of level $j = 2$.

3.2.2 PGD-Based Regression (rPGD) and Sparse PGD (sPGD)

The main drawbacks of the technique just presented are from one side, the difficulty to address the case of multiple parameters and, from the other, the necessity of expressing the parametric space as a hyper-hexahedron.

An alternative procedure consists in defining a sparse approximation in high dimensional settings [16]. For the ease of exposition and, above all, representation, but without loss of generality, let us begin by assuming that the unknown objective function $f(x, y)$ lives in $\mathbb{R}^2$ and that it is to be recovered from sparse data. For that purpose we consider the Galerkin projection

$$\int_\Omega w(x, y)(u(x, y) - f(x, y))dxdy = 0, \tag{14}$$

where $\Omega \subset \mathbb{R}^2$ and $w^*(x, y) \in \mathcal{C}^0(\Omega)$ is an arbitrary test function.

Following the Proper Generalized Decomposition (PGD) rationale, the next step is to express the approximated function $u^{\mathrm{M}}(x, y) \approx u(x, y)$ in the separated form and look for the enriched approximation $u^n(x, y)$ assuming known $u^{n-1}(x, y)$,

$$u^n(x, y) = u^{n-1}(x, y) + X_n(x)Y_n(y). \tag{15}$$

with

$$u^{n-1}(x, y) = \sum_{k=1}^{n-1} X_k(x)Y_k(y).$$

It is worth noting that the product of the test function $w(x, y)$ times the objective function $f(x, y)$ is only evaluated at few locations (the ones corresponding to the available sampled data). Since information is just known at these P sampling points (x_i, y_i), $i = 1, \ldots, P$, it seems reasonable to

express the test function not in a finite element context, but to express it as a set of Dirac delta functions collocated at the sampling points,

$$w(x,y) = u^*(x,y)\sum_{i=1}^{P}\delta(x_i,y_i) \\ = (X^*(x)Y_n(y) + X_n(x)Y^*(y))\sum_{i=1}^{P}\delta(x_i,y_i). \tag{16}$$

In the expressions above nothing has been specified about the basis in which each one of the one-dimensional modes was expressed. An appealing choice ensuring accuracy and avoiding spurious oscillations consists of using interpolants based on Kriging techniques.

The just described procedure defines a powerful non-linear regression called rPGD. Following our recent works on multi-local-PGD representations [26], local approximations ensuring continuity could be defined.

The rPGD-based regression technique could be applied to interpolate fields obtained through commercial software, allowing a drastic reduction of the sampling size, with respect to the SSL technique. When applied for that purpose it is called sPGD (for sparse PGD).

If we consider a set of $\mathtt{S}$ points in the parametric space, here assumed one-dimensional for the sake of simplicity , i.e. μ_j, and the solution calculated at those points: $u^j(\boldsymbol{x},t) = u(\boldsymbol{x},t,\mu_j)$, the parametric solution $u(\boldsymbol{x},t,\mu,\eta)$ expressed by

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

is constructed by employing the same procedure that in the regression case described above.

3.3 Miscellaneous

3.3.1 Compressing the Resulting Separated Representations

The main drawback of the non-intrusive separated representation constructor with respect to the intrusive one, is that the former produces too many terms in the finite sum, that is, too many modes, much more than those needed to approximate the solution at the same accuracy.

Imagine for a while that the SSL (or the sPGD) procedure leads to the $\mathtt{M}$-term representation

$$u(x,y) = \sum_{i=1}^{\mathtt{M}} X_i(x)Y_i(y),$$

for a given residual. Assume that this residual is known to accept a more compact representation, i.e., one with a smaller number of modes $\tilde{\mathtt{M}}$, with $\tilde{\mathtt{M}}<\mathtt{M}$. In this case, PGD can be efficiently used for post-compression [7], by simply to applying the PGD approximation algorithm to any non-optimal PGD solution, $f(x, y)$, in the form

$$f(x,y) = \sum_{i=1}^{\mathtt{M}} X_i(x)Y_i(y),$$

and then looking for a new separated expression of $u(x, y)$ according to

$$\int_\Omega u^*(u(x,y) - f(x,y))dxdy = 0,$$

where $u(x, y)$ is searched in the separated form

$$u(x,y) = \sum_{i=1}^{\tilde{\mathtt{M}}} X_i^c(x)Y_i^c(y).$$

Here, the super-index $\bullet^c$ refers to the *compressed* separated representation.

3.3.2 Quantities of Interest and Their Sensitivities

We consider the generic problem

$$\mathcal{L}(u(\boldsymbol{x},t,\boldsymbol{\mu})) = 0,$$

with $\mathcal{L}(\cdot)$ a linear or nonlinear differential operator, acting on a parametric field. In our case this field will be denoted by $u(\boldsymbol{x},t,\boldsymbol{\mu})$, where $\boldsymbol{\mu}$ is the vector of model parameters $\mu_1,\ldots,\mu_{\mathtt{Q}}$. By using the standard PGD, or its nonintrusive counterparts, we are able to write the parametric solution in the separated form

$$u(\boldsymbol{x},t,\mu_1,\ldots,\mu_{\mathtt{Q}}) \approx \sum_{i=1}^{\mathtt{M}} X_i(\boldsymbol{x})T_i(t)M_i^1(\mu_1)\cdots M_i^{\mathtt{Q}}(\mu_{\mathtt{Q}}),$$

or in its equivalent tensor form

$$\mathbf{U} \approx \sum_{i=1}^{\mathtt{M}} \boldsymbol{X}_i \otimes \mathbf{T}_i \otimes \mathbf{M}_i^1 \otimes \cdots \otimes \mathbf{M}_i^{\mathtt{Q}},$$

with $\mathbf{U}$ the multi-tensor whose entry $k,l,m_1,\ldots,m_{\mathtt{Q}}$ contains the value of the field u at point, time and parameters referred by these indexes, i.e., $u(\boldsymbol{x}_k,t_l,\mu_{1_{m_1}},\ldots,\mu_{\mathtt{Q}_{m_{\mathtt{Q}}}})$. Obviously, in any other point that does not coincide with a node of the mesh of space ($\boldsymbol{x}_k$), time (t_l) or parameters ($\mu_{1_{m_1}},\cdots$), the solution is computed by interpolation.

We assume now that we are not directly interested in the field involved in the physical model $u(\boldsymbol{x},t,\boldsymbol{\mu})$ itself, but in another output field of interest $\mathcal{O}$, that, for the sake of simplicity, is assumed scalar and depending on every model coordinate $(\boldsymbol{x},t,\mu_1,\ldots,\mu_{\mathtt{Q}})$. Assume that it could be derived from the former according to

$$\mathcal{O}(\boldsymbol{x}, t, \boldsymbol{\mu}) = \mathcal{G}(u(\boldsymbol{x}, t, \boldsymbol{\mu})).$$

Thus, we can compute the output at the collocation points when using the SSL technique or in the points of a sparse sampling (e.g., carried out by using the Latin Hyper–Cube method) so as to define, or better, learn, the model

$$\mathcal{O}(\boldsymbol{x}, t, \boldsymbol{\mu}) \approx \sum_{i=1}^{\mathtt{O}} \mathcal{M}_i(\boldsymbol{x}) \mathcal{T}_i(t) \mathcal{M}_i^1(\mu_1) \cdots \mathcal{M}_i^{\mathtt{Q}}(\mu_{\mathtt{Q}}).$$

Remark 6

- This separated representation can be easily obtained by using the SSL or the sPGD previously presented.
- The model $\mathcal{O}(\boldsymbol{x}, t, \boldsymbol{\mu})$ can be also constructed by making use of machine learning techniques, from the known output in a large enough number of points.

The sensitivity of the output to a given parameter, in the expression below to μ_1 reads [27]

$$\frac{\partial \mathcal{O}(\boldsymbol{x}, t, \boldsymbol{\mu})}{\partial \mu_1} \approx \sum_{i=1}^{\mathtt{O}} \mathcal{M}_i(\boldsymbol{x}) \mathcal{T}_i(t) \frac{\partial \mathcal{M}_i^1(\mu_1)}{\partial \mu_1} \mathcal{M}_i^2(\mu_2) \cdots \mathcal{M}_i^{\mathtt{Q}}(\mu_{\mathtt{Q}}).$$

3.3.3 Uncertainty Propagation

We recall here the model of the quantity of interest

$$\mathcal{O}(\boldsymbol{x}, t, \boldsymbol{\mu}) \approx \sum_{i=1}^{\mathtt{O}} \mathcal{M}_i(\boldsymbol{x}) \mathcal{T}_i(t) \mathcal{M}_i^1(\mu_1) \cdots \mathcal{M}_i^{\mathtt{Q}}(\mu_{\mathtt{Q}}).$$

If parameters are totally uncorrelated, the probability distribution of all them becomes independent, so that the probability density function can be expressed as

$$\Xi(\mu_1, \ldots, \mu_{\mathtt{Q}}) = \xi_1(\mu_1) \cdots \xi_{\mathtt{Q}}(\mu_{\mathtt{Q}}).$$

When correlations cannot be totally avoided, we can express the joint probability density $\Xi(\mu_1, \ldots, \mu_{\mathtt{Q}})$ in a separated form (by invoking the SSL or the sPGD):

$$\Xi(\mu_1, \ldots, \mu_{\mathtt{Q}}) \approx \sum_{i=1}^{\mathtt{R}} \mathcal{F}_i^1(\mu_1) \cdots \mathcal{F}_i^{\mathtt{Q}}(\mu_{\mathtt{Q}}).$$

With both the output and joint probability density expressed in a separated form, the calculation of the different statistical moments becomes straightforward. Thus, the first moment, the average field results in

$$\overline{\mathcal{O}}(\boldsymbol{x}, t) = \int_{\Omega_1 \times \cdots \times \Omega_{\mathtt{Q}}} \mathcal{O}(\boldsymbol{x}, t, \mu_1, \ldots, \mu_{\mathtt{P}}) \; \Xi(\mu_1, \ldots, \mu_{\mathtt{Q}}) \; d\mu_1 \cdots d\mu_{\mathtt{Q}},$$

where Ω_k denotes the domain of parameter μ_k. The separated representation is a key point for the efficient evaluation of this multidimensional integral, that becomes a series of one dimensional integrals. The calculation of higher order statistical moments (variance, …) proceeds in a similar manner.

Remark 7

- Monte-Carlo strategies can be also used in a very efficient way since the solution is available for any parameter choice.
- The knowledge of the parameter distribution can be used in a parametric stochastic setting [28].
- When addressing stochastic fields, appropriate spatial parametrization can be introduced based for example on the Karhunen–Loève expansions or the use of polynomial chaos.
- Parametric solutions are also very valuable when addressing Bayesian inference, for example.

3.3.4 Data-Assimilation and Advanced Virtual and Augmented Reality

Data assimilation is the process by which experimental measurements are incorporated into the modeling process of a given system. Data assimilation becomes a key player in dynamic data-driven application systems (DDDAS), as well as for mixed or augmented reality applications, for instance.

Both applications need real-time feedbacks. Depending on the latency of the particular system, these can oscillate from a few seconds to some milliseconds, for instance, if haptic (tactile) feedback is sought. To achieve these impressive feedback rates, the model and its solution play a fundamental role. If, as is nearly always the case, nonlinear problems are considered, such feedback rate restrictions can only be achieved by employing some for of model order reduction. In our previous works we have employed PGD strategies.

If we assume that the vademecum (PGD) solution of the parametric problem is available, given a set of measurements, the precise value of every parameter can be identified in almost real time by using inverse methodologies, e.g., Kalman filters [12], Tikhonov regularization [29], gradient methods [10, 30, 31], or Bayesian inference [32], to cite but a few.

The use of parametric solutions for immersive virtual reality purposes has been successfully accomplished [33]. Two examples developed by ESI on crash and stamping are sketched in Fig. 6. More spectacularly, a combined strategy integrating parametric solutions, computer vision and inverse analysis allowed unique performances in both feedback rates and realism in augmented reality applications [34]. The same techniques are now being employed for using simulated reality for intelligence augmentation.

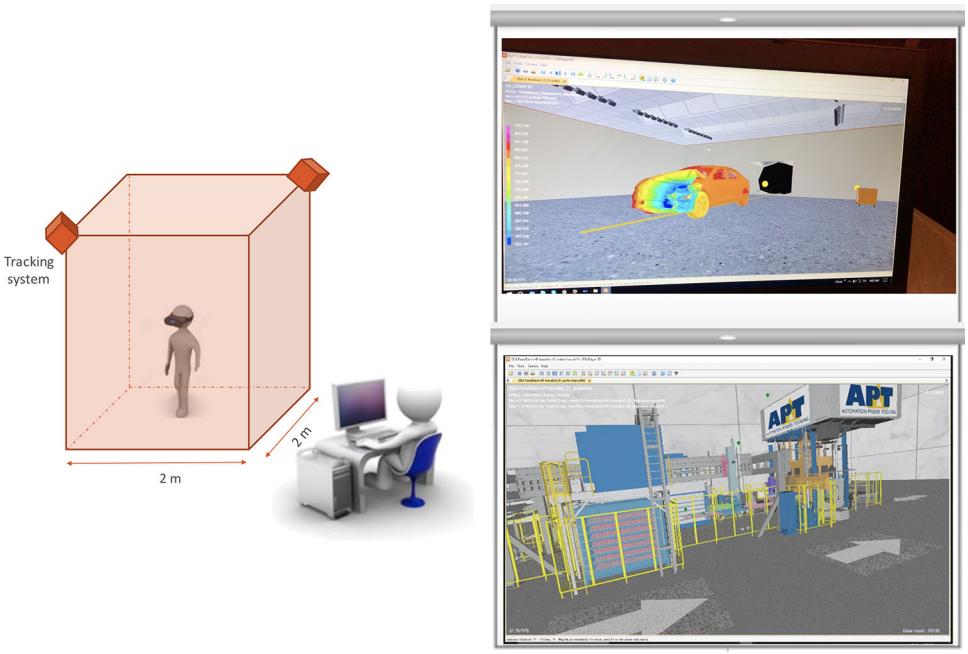


Fig. 6 Crash and stamping immersive virtual reality platform by ESI

4 Methods Based in Data-Science

As widely discussed in Sect. 1, engineering is evolving in the same way than society. However, data could offer much more than a simple state-of-the-art model calibration, and not only from a simple statistical analysis, but from the artificial intelligence perspective:

- Data can be used to produce data-based models, by relating the selected outputs of interest to uncorrelated inputs.
- Data can be used to create data-based models to enrich state-of-the-art models based on well-established physics (first principle or largely accepted phenomenological constitutive equations). Thus, the data contribution is expected to compensate (in a pragmatic way) the modeler ignorance, or the excessive system complexity impossible to capture for some reason.
- Data can be used to classify behaviors, tendencies, features. Special attention must be paid to the considered metrics and induced invariance (Euclidean, fuzzy, topological persistence, …).
- Data can offer the possibility to extract patterns with high information contents. This is crucial in predictive maintenance, inspection, supervision, control, etc.
- Multi-dimensional data can be visualized (using a particular manifold reduced representation) in order to extract hidden relations.
- By extracting the existing correlations and then by removing them, data results in valuable, sufficient and explicative information. Models constructed from information more than from the raw data, result in knowledge, key for real-time decision making purposes.
- The *smart-data paradigm* should replace the—in many cases irrational—big data-based habits and procedures. First-order physics and their associated models could inform on the most pertinent data to be collected, the places and time instants to perform that measurements, and the most adequate observation scale(s). This is extremely important because data is expensive to collect and also expensive to treat.
- After data collection, it must be assimilated into models using adequate procedures. Sometimes, missing data must be completed (data-completion) to offer a global map or to infer measures in regions/places where measures cannot be directly performed.
- Data filtering (models are excellent filters, but when proceeding directly from data, noise is a real inevitable issue), the exclusion of outliers (even if sometimes outliers are crucial, since they are related to fortuity defects), become compulsory.
- Data must be compressed, mainly if it is involved in streaming procedures. This implies the use of specific technologies (tensor formats, compressed sensing, gappy ROMs, …).
- Data "V's" (variability, veracity, volume, value, …) must be addressed from a computational perspective.
- The statistical nature of data represents an added difficulty, since uncertainty must be quantified and its propagation evaluated for addressing reliability.
- And many other known and still unknown possibilities. The domains is expanding exponentially.

From the above list, it seems clear that the use of data, nowadays and, more importantly, in the future, drastically differs from the use of it in the past. It seems clear that two competences/expertises must be considered independently (but without a total dissociation, since both should continue interacting intimately): data-collection and data-analysis.

Even if as just mentioned both should intimately interact, the intrinsic nature, tools, procedures, …of each become more and more different to the other, and consequently they require different approaches. The former will be centered in measurements, the second on data, both with their own science and technological contents and specificities.

4.1 Extracting Embedded Manifolds: Manifold Leaning Based in Linear and Nonlinear Dimensionality Reduction

Very often, our system evolves on manifolds of reduced dimension (d) embedded into the high-dimensional phase space $\mathbb{R}^D$ in which the problems is defined. This is the so-called *slow manifold*. By extracting these manifolds, the computational complexity of discretization techniques reduces significantly. This fact is at the roots of model order reduction techniques. Proper Orthogonal Decomposition or Reduced Bases techniques extract first this manifold and then proceed to solve problems by exploiting the low dimensionality of this manifold ($d \ll D$). On the contrary, PGD constructs the manifold and its approximation at the same time.

In the same way, in the case of a parametric model, dimensionality reduction allows to extract the number of informative, uncorrelated parameters (that depends linearly or nonlinearly on the original model parameters). This way of doing things becomes extremely useful when solving a parametric problem, since the lower is the number of significant parameters, the simpler becomes its parametric solution, its offline construction, and its online manipulation.

It is well known that the human brain consumes only 4 watts of power to perform some tasks for which today's computers will require the power of several nuclear power plants. Therefore, our usual way of doing simulation, despite the impressive progress in our computers and algorithms, must be definitively suboptimal. In everyday life, we distinguish and recognize, almost instantaneously, a tree or a human being, even those that we never met before. This means that, despite the diversity and apparent complexity, few parameters should suffice to accomplish the task of classifying. In other words, if recognizing something will depend on thousands of parameters, the human being will have to spend hours performing the task. In that case, his or her survival will be compromised. Since we have survived all along the long history of evolution, it is because, without any doubt, in general the answer lies only in few almost uncorrelated data. Big data is accompanied most of the time by small information. The apparent diversity is hidden in the small scales, but the largest scales suffice for having a useful image of the nature and our environment, and then to make adequate decisions nearly in real-time.

The big challenge is how to remove these intricate correlations and how to express reality in this new resulting frame? How to discover the frame in which complexity disappears in favor of simplicity? How to visualize reality in that new frame? in which coordinate axes? what is the physical meaning of these new axes?

Accumulated learning, starting from our infancy, provided us the capacity of pattern recognition in its more general sense. To adapt ourselves faster, learning should be sped up by replacing the human brain by powerful computers based on electrons, very soon in quantum effects, that proceed much faster [35]. One second of a standard laptop calculation is equivalent to the calculations that a human brain could perform during a long life devoted to the same task. Today some routine calculations, e.g., crash test simulations, require tens of millions of computing hours, equivalent to thousands of years on a single core computer. These unimaginable calculations can be performed in only few days by using high-performance computing platforms, making use of thousands of cores working in parallel.

The only need to this end is adequate (robust and efficient) algorithms to recognize and extract simplicity from the apparent complexity so as to proceed from it. Manifold learning techniques, few of them summarized in what follows, is a valuable route. Imagine for a while that we are interested in solving the mechanical problem related to liver deformation in biomechanics. The main issue, is that each patient has its own liver whose shape (anatomy) defining the domain Ω in which the mechanical problem must be solved, is "similar" qualitatively but "different" quantitatively to any other liver. Thus, one is tempted to introduce parameters defining the liver shape as model parameters and then compute the mechanical problem solution for any choice of these parameters. But, how many geometrical parameters define a liver? Each one of us could propose a different number related to different geometrical features, probably many tens, even hundreds. In [36] it was proved, using nonlinear dimensionality reduction and, more concretely, manifold learning techniques, that few almost uncorrelated parameters (2–4) largely suffices to represent accurately any human liver. Thus, in [37] parametric models based on the PGD were developed and successfully used.

For the sake of completeness, even if many papers and books deeply address the foundations and applications of these techniques, in what follows some popular linear and nonlinear dimensionality reduction techniques, widely employed in our works, are summarized.

4.1.1 Principal Component Analysis and Its Locally Linearly Counterpart

Let us consider a vector $\boldsymbol{y} \in \mathbb{R}^D$ containing some experimental results. These results are often referred to as *snapshots* of the system. If they are obtained by numerical simulation, they consist of nodal values of the essential variable along time. Therefore, these variables will be somehow correlated and, notably, there will be a linear transformation $\boldsymbol{W}$ defining the vector $\boldsymbol{\xi} \in \mathbb{R}^d$, with $d < D$, which contains the still unknown *latent variables*, such that

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{\xi}. \tag{17}$$

The transformation matrix $\boldsymbol{W}$, $D \times d$, satisfies 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 $\boldsymbol{I}_D$). This transformation is the key ingredient of the principal component analysis (PCA) [38].

Assume that there exist M different snapshots $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M$, which we store in the columns of a $D \times M$ matrix $\boldsymbol{Y}$. The associated $d \times M$ reduced matrix $\boldsymbol{\Xi}$ contains the associated vectors $\boldsymbol{\xi}_i$, $i = 1, \ldots, M$.

PCA works usually with centered variables. In other words,

$$\begin{cases} \sum_{i=1}^{M} \boldsymbol{y}_i = \boldsymbol{0} \\ \sum_{i=1}^{M} \boldsymbol{\xi}_i = \boldsymbol{0} \end{cases}.$$

Otherwise, observed variables must be centered by removing the expectation of $\mathrm{E}\{\boldsymbol{y}\}$ to each observation $\boldsymbol{y}_i$, $i = 1, \ldots, M$. This is done by subtracting the sample mean, given the fact that the expectation is not known, in general.

What is remarkable about PCA is its ability to calculate both d—the dimensionality of the embedding space—and the associated transformation matrix, $\boldsymbol{W}$. PCA proceeds by guaranteeing maximal preserved variance and decorrelation in the latent variable set $\boldsymbol{\xi}$. The latent variables in $\boldsymbol{\xi}$ will therefore be uncorrelated, thus constituting a basis. In other words, the covariance matrix of $\boldsymbol{\xi}$,

$$\boldsymbol{C}_{\xi\xi} = \mathrm{E}\{\boldsymbol{\Xi}\boldsymbol{\Xi}^T\},$$

will be diagonal.

Observed variables will most likely be correlated. PCA will then extract the d uncorrelated latent variables by resorting to

$$\begin{aligned} \boldsymbol{C}_{yy} &= \mathrm{E}\{\boldsymbol{Y}\boldsymbol{Y}^T\} = \mathrm{E}\{\boldsymbol{W}\boldsymbol{\Xi}\boldsymbol{\Xi}^T\boldsymbol{W}^T\} = \boldsymbol{W}\mathrm{E}\{\boldsymbol{\Xi}\boldsymbol{\Xi}^T\}\boldsymbol{W}^T \\ &= \boldsymbol{W}\boldsymbol{C}_{\xi\xi}\boldsymbol{W}^T. \end{aligned}$$

Pre- and post-multiplying by $\boldsymbol{W}^T$ and $\boldsymbol{W}$, respectively, and making use of the fact that $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$, gives us

$$\boldsymbol{C}_{\xi\xi} = \boldsymbol{W}^T\boldsymbol{C}_{yy}\boldsymbol{W}. \tag{18}$$

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

$$\boldsymbol{C}_{yy} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T, \tag{19}$$

with $\boldsymbol{V}$ containing the orthonormal eigenvectors and $\boldsymbol{\Lambda}$ the diagonal matrix containing the eigenvalues, sorted in descending order.

Substituting Eq. (19) into Eq. (18), we arrive at

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

This equality holds when the d columns of $\boldsymbol{W}$ are taken collinear with d columns of $\boldsymbol{V}$.

We then conserve those eigenvectors associated with the d nonzero eigenvalues,

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

which gives

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

We therefore conclude that the eigenvalues in $\boldsymbol{\Lambda}$ represent the variance of the latent variables (diagonal entries of $\boldsymbol{C}_{\xi\xi}$).

Noise may often corrupt experimental observations. If this is the case, every eigenvalue of $\boldsymbol{C}_{\xi\xi}$ is strictly positive, and the choice of the d most representative columns in $\boldsymbol{V}$ becomes intricate. For that to be useful, latent variables must have variances larger than noise. In that case, it is enough to choose the eigenvectors associated with the d largest eigenvalues.

There is a clear geometrical interpretation of all this: the columns of $\boldsymbol{V}$ indicate the vectors in $\mathbb{R}^D$ that span the subspace of latent variables. In Fig. 7 this fact can be observed. In the left figure a set of points in $\mathbb{R}^2$ is represented. Notice however that these points show some pattern, as they are ordered along a diagonal line, that constitutes the already mentioned slow manifold. PCA is able to find an alternative representation, by expressing these points in a new coordinate system, defined by $\boldsymbol{V}$ (axes in red). In this new coordinate system, all these points lie clearly in a one-dimensional space.

PCA has been re-discovered several times in recent times, under different names, in different scientific specialities. It relies, nevertheless, in the basic assumption of linear dependency expressed by Eq. (17) between observed and latent variables. This is precisely one of its most

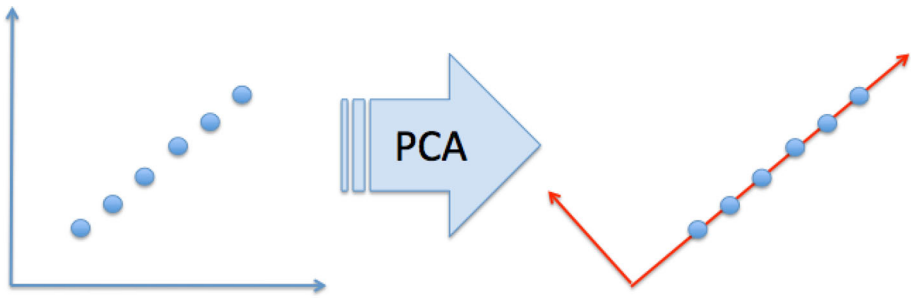


Fig. 7 Geometrical interpretation of PCA

relevant limitations, that lead recently to a growing interest on the so-called non-linear dimensionality reduction (NLDR) techniques.

Latent variables move frequently around a so-called slow manifold. If this manifold is not flat, as is frequently the case, the projection in Eq. (17) will simply not exist. Examples of this situation include, for instance, non-linear, large strain solid dynamics. NLDR methods are of course more general than linear ones, allowing for richer relationships between latent variables and the experimental ones. This is shown in Fig. 8, where the reader can notice how no rotation will give us the desired one-dimensional embedding of Fig. 7. PCA does not see this situation, and perceives points as pertaining to a two-dimensional manifold, even if they pertain to a spiral-like curve, which is in fact a one-dimensional manifold.

Local-PCA (ℓ-PCA) constitutes an alternative to standard PCA. It simply consists of PCA applied locally, i.e., to each data point and its closest neighbors, see Fig. 9 [39]. This gives rise to additional difficulties, such as finding the way to align the different basis for every patch in the data [40].

ℓ-PCA has another appealing property: if all the dimensions are kept, that is $d = D$, ℓ-PCA allows aligning locally the reduced manifold with the transformed coordinates, but since no coordinate axis is removed, points out of the reduced manifold can be placed and transported to the initial space by using the inverse mapping.

4.1.2 Multidimensional Scaling

PCA works with the covariance matrix of the experimental results, $\boldsymbol{Y}\boldsymbol{Y}^T$. However, multidimensional scaling, MDS, (like k-PCA, which will be described hereafter) works with the the Gram matrix containing scalar products, i.e., $\boldsymbol{S} = \boldsymbol{Y}^T\boldsymbol{Y}$ [38].

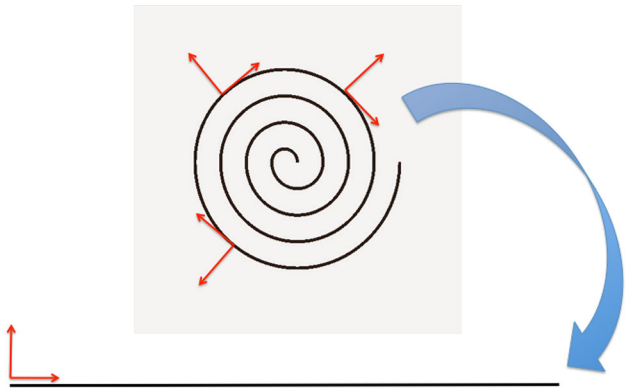


Fig. 9 Sketch of local-PCA

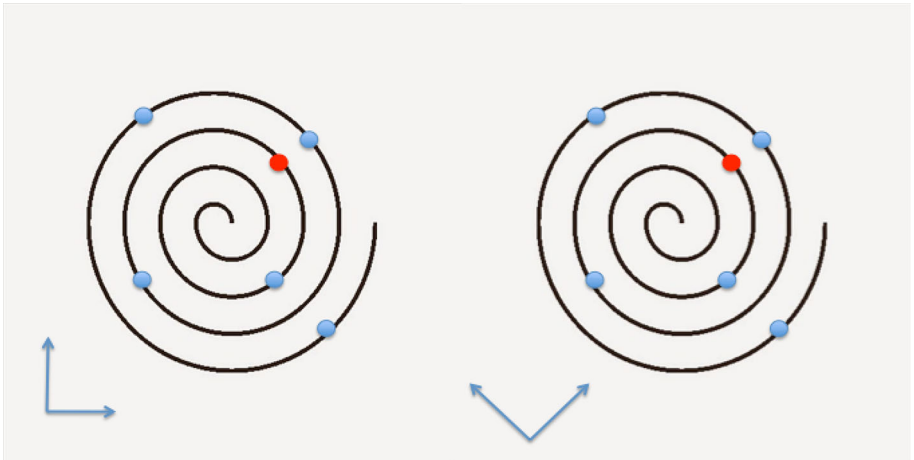


Fig. 8 PCA limits in presence of strongly-nonlinear manifolds

Multidimensional scaling methods construct a configuration of points in a target metric space from information about point distances. MDS preserves pairwise scalar products instead of pairwise distances. They are nevertheless closely related:

$$\boldsymbol{S} = \boldsymbol{Y}^T\boldsymbol{Y} = \boldsymbol{\Xi}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{\Xi} = \boldsymbol{\Xi}^T\boldsymbol{\Xi}.$$

Computing the eigenvalues of $\boldsymbol{S}$, we arrive at

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

which in turn gives

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

Proving the equivalence between MDS and PCA is therefore straightforward [38].

4.1.3 Kernel Principal Component Analysis

The origin of kernel Principal Component Analysis, k-PCA methods is very appealing for its intuitiveness. It adds, however, some technical difficulties that will be described next. In fact, it is easy to understand that data not linearly separable in D dimensions, could be linearly separated if previously projected to a space in $Q > D$ dimensions [38]. It may appear surprising that k-PCA projects the data to a higher dimensional space, in an attempt to linearize the underlying manifold $\mathcal{M}$. Therefore, a mapping

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

is constructed, where Q may be an arbitrary number of dimensions. The true advantage comes, however, from the fact that it is not necessary to write down the analytical expression of the mapping ϕ.

The symmetric matrix $\boldsymbol{\Phi} = \boldsymbol{Z}^T\boldsymbol{Z}$ has to be decomposed in eigenvalues and eigenvectors. Previously, the mapped data $\boldsymbol{z}_i$ involved in $\boldsymbol{\Phi}$ must be centered. Since the mapping is unknown, this centering process may seem difficult. However, centering can be done in an implicit way. The interested reader should consider to consult classical references in the field such as [41, 42].

The eigenvector decomposition can now be performed on the doubly-centered matrix,

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

giving rise to

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

The mapping ϕ could provoke scalar products to become prohibitive, given the fact that the vectors will now be expressed in a space of a high number of dimensions, Q. To avoid this high-dimensional multiplication and even the search for ϕ, a kernel function κ is employed that, based upon Mercer's theorem—also knwon as the *kernel trick*—, directly gives the value of the scalar product $\kappa(\boldsymbol{y}_i, \boldsymbol{y}_j) = \boldsymbol{z}_i \cdot \boldsymbol{z}_j$. Mercer's theorem states that if $\kappa(\boldsymbol{u}, \boldsymbol{v})$ is continuous, symmetric and positive definite, then it defines an inner-product in the mapped space.

Many different kernels exist that fulfill Mercer's condition, such as, for instance:

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

No practical tip can be offered to choose any particular mapping ϕ. The goal is simply to linearize the manifold to be embedded. If this goal is met, then the application of PCA will suffice to unveil the nonlinear principal components of the data set, that now lives in a flat space.

4.1.4 Locally Linear Embedding

From the set of points $\boldsymbol{y}_i \in \mathbb{R}^D$, $i = 1, \ldots, M$, Locally Linear Embedding, LLE, methods proceed in two steps [43]:

1. Interpolate each point $\boldsymbol{y}_i$, $i = 1, \ldots, M$ linearly by choosing a number K of its nearest neighbors. Note that this interpolation is local (is performed only among its nearest neighbors) and linear. One of the most cited limitations of LLE is precisely to have to choose K. In principle, it should be greater that the expected dimension d of the embedding manifold, while the neighbors should be close enough so as to ensure the validity of linear approximation. In sum, we exploit the classical definition of what a manifold is: a geometric structure homeomorphic to a plane in the neighborhood of each point. Choosing a small number of neighbors K and a large sampling M provides almost always a satisfactory reconstruction.

   This linear reconstruction of each data point $\boldsymbol{y}_i$ can be expressed as:

   $$\boldsymbol{y}_i = \sum_{j\in\mathcal{S}_i} W_{ij}\boldsymbol{y}_j,$$

   with W_{ij} the sought weights and $\mathcal{S}_i$ the set of the K-nearest neighbors of $\boldsymbol{y}_i$.

   The set of weights that best approximates the manifold structure of the data will be obtained by minimizing the functional

$$\mathcal{F}(\boldsymbol{W}) = \sum_{i=1}^{M} \left\| \boldsymbol{y}_i - \sum_{j=1}^{M} W_{ij} \boldsymbol{y}_j \right\|^2,$$

where W_{ij} is zero if $\boldsymbol{y}_j$ is not one of the K-nearest neighbors of $\boldsymbol{y}_i$.

2. Every linear patch around $\boldsymbol{y}_i$, $\forall i$, is mapped onto a lower dimensional embedding space of dimension $d \ll D$. The key ingredient of LLE methods is to assume that the same weights will hold in the new, low-dimensional embedding space. If the weights remain, the problem reduces now to find the particular coordinates of each point $\boldsymbol{y}_i$ in the embedding space, $\boldsymbol{\xi}_i \in \mathbb{R}^d$ that make it possible to maintain the value of the weights.

   This is achieved by defining a second functional $\mathcal{G}$, as a function of the sought coordinates, $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M$

$$\mathcal{G}(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M) = \sum_{i=1}^{M} \left\| \boldsymbol{\xi}_i - \sum_{j=1}^{M} W_{ij} \boldsymbol{\xi}_j \right\|^2.$$

   In this functional the weights are assumed known while we look for the reduced coordinates $\boldsymbol{\xi}_i$. Minimization of $\mathcal{G}$ gives rise to a $M \times M$ eigenvalue problem whose d lowest non-zero eigenvalues define the basis of the space in which the manifold is embedded.

   It is worth noting that $\mathcal{G}(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M)$, with the different coordinates $\boldsymbol{\xi}_i$ already determined, allows us to obtain a local error estimator as

$$\mathcal{E}(\boldsymbol{\xi}_i) = \left\| \boldsymbol{\xi}_i - \sum_{j=1}^{M} W_{ij} \boldsymbol{\xi}_j \right\|. \tag{20}$$

4.2 Data-Driven Mechanics: Data-Based Constitutive Equations

In an environment in which large scientific infrastructures produce petabytes of data every day, it was unavoidable that computational mechanics succumbed under the tsunami of big data. Science was first experimental (the so-called first paradigm of science), then was able, by means of models, to establish a theoretical paradigm. In the last decades it has become heavily computational, so as to make predictions by simulating the already established physical laws. However, very recently, the fourth paradigm of science is that of data exploration, the one that unifies data, theory and simulation [1].

We are far from an epoch of *hypothesis-neutral research* [44]. It is not either a question of finding correlations among data. What data-driven computational mechanics is all about is to be able to abandon the cumbersome times of data fitting to complex, phenomenological constitutive equations and to be able to perform simulations on top of large sets of experimental data without the need of over-simplifying assumptions. In other words: it is a question of bringing *computation to the data, rather than data to the computation,fourth-paradigm*.

The word *genome*, when applied out of the context of biological systems, refers to a *fundamental building block toward a larger purpose*. The materials genome—see https://mgi.nist.gov/—is an initiative set forth by the White House in USA. to face the challenge of incorporating new, designed materials to the market twice as fast at a fraction of the nowadays cost. This initiative emphasizes the need for the design of more advanced computational techniques able to supplement physical experiments. This will be possible if data are shared and integrated across the "materials continuum" process of design. The materials genome initiative highlights the need for an integrated workflow of experiments, simulation and theory and the *development of advanced simulation tools* that are validated through experimental data [45]. It also emphasizes the need to make digital data accessible, including combining data from experiment and computation into a searchable materials data infrastructure. This need has revealed, however, being totally insufficient. For instance, data produced in one week by the Spallation Neutron Source in the USA used to take one year of graduate student's time to analyze [46]. Now, this research installation is producing data one hundred times faster.

Therefore, it is absolutely necessary to go substantially beyond: to develop simulation methods able to integrate and perform data acquisition, reduction, assimilation and analysis so as to be able to seamlessly integrate them in the design and fabrication processes of products involving radically new materials.

Existing computational tools still posses some other fundamental limitations. One of the biggest is the difficulty of integrating disparate time and length scales. For instance, we can model and predict the vibration of atoms in a lattice at time scales on the order of picoseconds. But this information is not suitable for the prediction of materials behavior across the course of the years. If a computational tools is needed to cope with this challenge, it will need to acquire and reduce all this huge amount of data and convert it in knowledge. Therefore, the need for model order reduction techniques is seen as a must.

Materials Informatics is a new scientific discipline that applies the principles of informatics to the design of new materials. It shares much of the spirit of the materials genome initiative. Indeed, it envisages the design of "specialized informatics tools for data capture, management, analysis, and dissemination" and the need for "advances in computing power, coupled with computational modeling and simulation and materials properties

databases" [47]. Again, the possibility of sifting vast amounts of data reveals to be the bottleneck of a suitable strategy.

In an attempt to incorporate the huge possibilities of Big Data to the field of scientific computing, some proposals have been proposed very recently. The first one represents an attempt of working without constitutive laws [14]. In fact, they propose a method that works directly with balance equations and seeks for the experimental point that gives the state closest to equilibrium. To that end, it employs an optimization procedure.

This method re-opens the *epistemic controversy* between the scientific approach followed by Kepler—who, with the help of "big" data, was able to accurately describe planet's orbits—or the one by Newton, who unveiled the laws of physics behind gravitation that could finally explain why the computations done by Kepler were right.

The other approach, closer to the one of Newton, is to discover governing equations from data [48–50]. These methods need for some assumptions on the form of the particular sought physical laws, but determines a precise form of governing laws even in the presence of noised data.

The main limitation that can be envisaged about these two approaches is their ability to cope with large amounts of data. In particular, the approach in [14] performs an optimization procedure to find the experimental point closest to satisfying balance equations that could be very expensive in the presence of big data. Furthermore, in an ICME approach we want to create new materials, still inexistent, by extrapolating the conclusions obtained by experimental and computational data. This is not possible without employing some form of *machine learning*, able to extract trends from data and to foresight the properties of materials yet to come.

In this framework, computational mechanics is hold on top of three cornerstones: equilibrium, compatibility and constitutive equations. It is obvious that, as pointed out by Ortiz et al. [14], the later is of a lower epistemic character. It is simply nonsense to capture, curate and analyze petabytes of data just to verify equilibrium during an experiment or to check if compatibility is satisfied. Therefore, data-driven computational mechanics deals naturally with the issue of correctly reproducing from data the constitutive behavior of the material.

4.2.1 Early Times of Data-Driven Approaches

Of course, data-driven approaches in computational mechanics trace back to early parameter identification methods, that had an important popularity after the mid-nineties [51–57]. Essentially, this approach consisted of an inverse problem solving by finite elements so as to determine the value of the material parameter that best fits with the experimental results. However, this approach needs a pre-defined constitutive model and is therefore very intrusive in the process of material characterization.

By data-driven approaches, however, one tends to think of an approach that does no presuppose any form of constitutive equation. In fact, the work that is often considered as the first in the field, the one by Kirchdoerfer and Ortiz [14], does not employ any constitutive equation, and arose in an attempt to employ data directly in the computations. There exist, however, some previous works that, in the framework of numerical homogenization, tried to obtain a sort of response surface for a representative volume element subjected to any possible boundary condition, see for instance [58–61]. These response surface approaches avoided the employ of any form of constitutive equation while also avoided the always cumbersome task of a microstructure analysis at every Gauss point of the macroscopic model.

Recent works by W. K. Liu and coworkers share important similarities with this rationale. For instance, in [62] a method is developed that works by designing a sort of *response database* for material RVEs, such that it very much eases the task of designing new materials by simply interpolating among selected microstructures. Of course, in this approach, very much like in the one by Yvonnet and coworkers, the issue of the curse of dimensionality (given the vas amount of design parameters that exists in the problem) is of utmost importance. To circumvent this curse, Liu et al. [63] developed a technique coined as *self-consistent clustering analysis* (SCA). Basically, it relies on k-means clustering techniques to characterize the macroscopic response of similar material microstructures [64]. This technique has recently been extended to elasto-plastic materials with strain softening [65].

4.2.2 Working Without Constitutive Equations

While the work of Yvonnet et al. [58] assumes that input data comes from numerical simulations at the scale of the representative volume element (RVE) of the material, the work of Kirchdoerfer and Ortiz assumes experimental results. While the former employs high-dimensional interpolation so as to obtain a sort of response surface for the RVE, the one by Kirchdoerfer and Ortiz assumes that each experimental result is a pair of strain-stress values (since it is intended for trusses, no tensorial values are considered) that satisfy equilibrium and compatibility. Therefore, their method looks for the closest experimental pair in phase space to satisfy compatibility and equilibrium by minimizing a cost function.

In subsequent works, Kirchdoerfer and Ortiz extend this approach to noisy experimental data sets [66] and also to dynamics [67]. A similar approach is followed in [68] in which the Euclidean distance to experimental points is substituted by the Mahalanobis distance. Other than that, the approach is identical to [14].

More recently, the authors introduced the concept of *constitutive manifold*. By applying manifold learning to pairs of experimental or numerical stress–strain values, the manifold structure of these data can be unveiled so as to ascertain the constitutive behavior of the material or structure [69]. Assume that a set of $\mathtt{n}_{\text{exp}}$ experimental stress–strain couples are stored in our database. These couples are in fact points $\boldsymbol{X}_m \in \mathbb{R}^D$, $m = 1, \ldots, \mathtt{n}_{\text{exp}}$, in a space of dimension $D = 12$ (six stresses and six strains in Voigt notation). If some coherence exists between strains and stresses (and this is no more than a constitutive equation), then, these points could be projected without loss of information onto a manifold of dimension $d \ll D$. Consider, for instance, a set of randomly generated points according to a generalized Hooke's law. By employing Locally Linear Embedding (LLE) techniques, for instance, it is easy to find out that they pertain actually to a flat manifold in which only two parameters are relevant (Young modulus and Poisson's coefficient, for instance, or Lamé coefficients) [43]. The result of embedding coordinates $\boldsymbol{X}_m$ onto the two-dimensional manifold gives the reduced coordinates $\boldsymbol{\xi}_m$. This is represented in Fig. 1 (Fig. 10).

The concept of constitutive manifold not only provides with a very intuitive and visual concept (if the resulting manifold lives a small enough dimension). It allows to compute in a very efficient way by iterating between the equilibrium equation (which is always linear and global) and the non-linear and local constitutive manifold. The intersection between both manifolds will provide precisely with the sought state of the system in the phase space, see Fig. 11. A very simple iterating algorithm can thus be established that closely resembles the Large Time Increment technique by Ladeveze [19, 70, 71].

Thus, the equilibrium manifold $\mathcal{S}$ hosts stress–strain pairs in equilibrium $(\boldsymbol{\sigma}^n, \boldsymbol{\varepsilon}^n)$ at iteration n. To perform an iteration so as to obtain a suitable point on the constitutive manifold, $(\hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\varepsilon}})$ a search direction must be established. The intersection of this search direction with the constitutive manifold provides the sought pair. Note that this iteration is local, since each integration point on the model could be at a different stress–strain state. On the contrary, projection from the constitutive manifold onto the equilibrium manifold so as to obtain a new couple $(\boldsymbol{\sigma}^{n+1}, \boldsymbol{\varepsilon}^{n+1})$ must be done at a global scale.

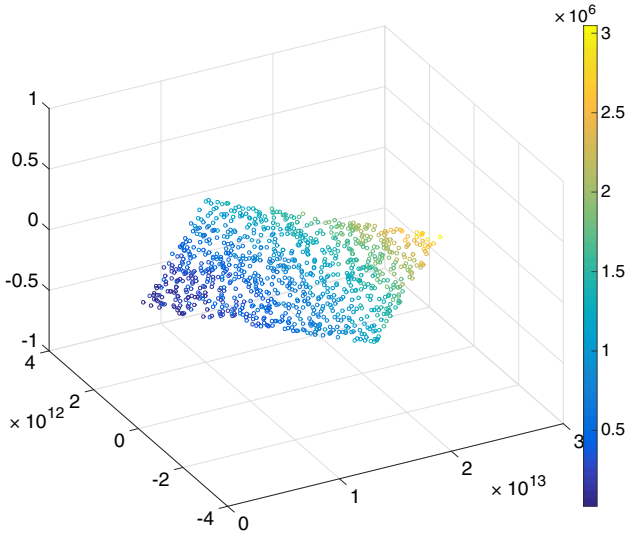


Fig. 10 Reduced coordinates $\boldsymbol{\xi}_m \in \mathbb{R}^2$, $m = 1, \ldots, \mathtt{n}_{\text{exp}}$, on the resulting two-dimensional constitutive manifold. These results correspond to a linear elastic material under small strains. The color map represents the associated elastic energy just to show that the embedding procedure does not hide information

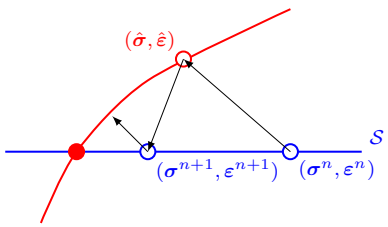


Fig. 11 Sketch of the iterative scheme proposed in [69]. In blue, the linear equilibrium manifold is represented. In red, the constitutive manifold. The technique iterates until finding the intersection of both manifolds, the true state of the system in the phase field

In [72] this technique is extended to materials with rich microstructure in which image techniques can be employed so as to ascertain the details associated with this fine level of detail. For these, the concept of constitutive manifold allows for a proper interpolation among selected sampled RVEs, producing finally a technique that works very much like the ones developed by Yvonnet and coworkers. The extension of the concept of constitutive manifold to problems with elastoplastic behavior was addressed in [73]. In [37], on the contrary, kernel-PCA techniques [41, 42]were employed to ascertain the precise form of the manifolds for different microstructures.

4.2.3 Hyperelasticity

Hyperelasticity deserves maybe a special comment, since it is characterized by the presence of a stored energy (potential) function so as to guarantee energy conservation in

closed cycles. In this framework, data-driven approaches are directed towards the precise determination of the *shape* of this energy functional. While the general procedure is to try to reproduce existing, well-known constitutive laws by means of parameter fitting of experimental data, Montans and coworkers propose to avoid the use of existing laws and to simply interpolate experimental results with the help of splines. This approach is based upon an old technique developed by Sussman and Bathe [74] and is now known as *what you prescribe is what you get* (WYPIWYG) hyperelastcity. It has been applied to transversely isotropic [75] as well as orthotropic materials [76], plasticity [77], compressible elasticity [78] and has been recently applied to living soft tissues [79, 80]. Although initially thought to precisely interpolate data points, when there is considerable noise in the data a new version must be employed [81].

4.2.4 Thermodynamic Consistency

One of the recurrent questions when studying data-driven procedures in the framework of integrated computational materials engineering (ICME) is that of noise in the data. Eventually, this could led to inaccuracies that may have as a consequence the violation of some first principles. For instance, how do we guarantee energy conservation and strict positive entropy generation in the presence of noise in the data?

Recently, the authors have presented a method able to incorporate noisy data and still guarantee the thermodynamic consistency of the resulting simulations [15]. The method is developed by resorting to the GENERIC formalism [82–84]. In a nutshell, the GENERIC ("General Equation for Non-Equilibrium Reversible-Irreversible Coupling") formalism seeks for an expression of the time evolution of the necessary variables to describe the material at hand, $\dot{z}_t$.

Basically, the GENERIC formalism assumes an evolution of the variables of the form

$$\dot{z}_t = \boldsymbol{L}(z_t)\nabla E(z_t) + \boldsymbol{M}(z_t)\nabla S(z_t), \quad z(0) = z_0, \tag{21}$$

where $\boldsymbol{L}$ is the so-called Poisson matrix, which is responsible for the reversible (Hamiltonian) part of the evolution of the system. E represents the energy of the system and $\boldsymbol{M}$ represents the friction matrix, responsible for the irreversible part of the evolution of the system. S represents the entropy of the system for the particular choice of variables z. The choice of these variables is not particularly relevant, since even if they result to be finally related, this will be detected by the method.

Matrices $\boldsymbol{L}$ and $\boldsymbol{M}$ need to satisfy the following relationship:

$$\boldsymbol{L}(z) \cdot \nabla S(z) = \boldsymbol{0}, \tag{22a}$$

$$\boldsymbol{M}(z) \cdot \nabla E(z) = \boldsymbol{0}, \tag{22b}$$

often referred to as degeneracy conditions. This is fulfilled by simply choosing $\boldsymbol{L}$ skew-symmetric and $\boldsymbol{M}$ symmetric, positive semi-definite. Then it is straightforward to verify that

$$\begin{aligned}\dot{E}(z) &= \nabla E(z) \cdot \dot{z} \\ &= \nabla E(z) \cdot \boldsymbol{L}(z)\nabla E(z) + \nabla E(z) \cdot \boldsymbol{M}(z)\nabla S(z) = 0,\end{aligned} \tag{23}$$

which is equivalent to the very basic principle of conservation of energy in closed systems. In turn,

$$\begin{aligned}\dot{S}(z) &= \nabla s(z) \cdot \dot{z} \\ &= \nabla S(z) \cdot \boldsymbol{L}(z)\nabla E(z) + \nabla S(z) \cdot \boldsymbol{M}(z)\nabla S(z) \geq 0,\end{aligned} \tag{24}$$

guarantees the satisfaction of the second principle of thermodynamics.

The method consists, then, in the identification of matrices $\boldsymbol{L}$ and $\boldsymbol{M}$—something straightforward in the vast majority of the cases—and the particular structure of the gradients of energy and entropy (Hamiltonian and dissipative parts of the constitutive equations, respectively).

In [15] this is done by a data fitting procedure that shows very promising characteristics. Not only the particular behavior of the material can be identified. The time discretization of Eq. (21) allows to develop as a byproduct a very efficient time integration scheme with the right properties in terms of conserving and dissipative magnitudes, see [85, 86] for more details.

4.2.5 Hybrid Methodologies

As just emphasized, a growing interest has arose on the development of data-driven techniques to avoid the employ of phenomenological constitutive models. While it is true that, in general, data do not fit perfectly to existing models, and present deviations from the most popular ones, we believe that this does not justify (or, at least, not always) to abandon completely all the acquired knowledge on the constitutive characterization of materials. Instead, what we recently proposed [87], by means of machine learning techniques, to develop correction to those popular models so as to minimize the errors in constitutive modeling.

Plenty of effort has been dedicated throughout history to create very accurate models, however, we also know that no model is perfect: it is always subjected to certain limiting hypotheses. In [87], we provided an alternative route by enhancing or correcting existing, well-known, models with information coming from data, thus performing a sort of *data-driven correction*. In that first work a special effort

was put on the correction of plastic yield functions, while work in progress addresses more complex scenarios involving hardening and damage.

The proposed data driven correction technique is conceptually simple. Imagine that our departure point is a given, well-known parametric model $\mathcal{M}(\boldsymbol{p})$. It is important to keep in mind that we are looking for an enhancement or correction of the previous model based on the available experimental results. Therefore, a *discrepancy* model $\mathcal{D}(\boldsymbol{c})$, which applies to the first model, needs to be defined. So to speak, *reality*, $\mathcal{R}$, is approximated as

$$\mathcal{R} = \mathcal{M}(\boldsymbol{p}) + \mathcal{D}(\boldsymbol{c})\big|_{\boldsymbol{p}}, \tag{25}$$

where $\boldsymbol{p}$ represents the set of parameters governing the model and $\boldsymbol{c}$ represents the set of parameters needed to define the necessary correction.

Since our measurement capabilities will in general be constrained to some experimentally observable quantities, both our objective *reality* and the correction to the model will be restricted to these experimental settings. In other words,

$$\mathcal{R}|_{\boldsymbol{s}} \approx \mathcal{M}(\boldsymbol{p}) + \mathcal{D}(\boldsymbol{c})\big|_{\boldsymbol{p},\boldsymbol{s}}. \tag{26}$$

It is worth to mention that the way we define the observables $\boldsymbol{s}$ could have an important impact over the calibration of the set of correction parameters, $\boldsymbol{c}$ and remains a research field very active as discussed later.

4.3 Model Learners

In the last decades we have seen a tremendous development of artificial intelligence (IA) techniques. Machine learning (ML) and manifold learning, and, notably, deep learning (DL) techniques, have assisted to an unprecedented growth in the wide range of applications they can be envisaged for. With the eruption of data-enabled science and engineering (the so-called fourth paradigm of science), applied science is today a symbiosis of theory, experiments and simulation.

In a changing scenario that goes beyond the industry 4.0 paradigm and moves towards the next generation, 5.0, based on collaborative robotics, connected devices are continuously producing huge amounts of data. These must be stored, curated and processed so as to unveil trends, find out hidden correlations, and, eventually, make decisions in real time. Learning governing equations from data has thus acquired an utmost importance in recent times.

In science, a model is no more than a mathematical expression relating an input and its associated output. If both, input and output, are expressed in a discrete form, i.e., through vectors $\mathbf{I}$ and $\mathbf{O}$ respectively, then the model can be expressed as the matrix $\mathbf{K}$ that allows computing the output $\mathbf{O}$ as soon as the input $\mathbf{I}$ is specified. In other words, the model expresses $\mathbf{KO} = \mathbf{I}$. In this discussion, and without loss of generality, we assume the same number of components of the input and output vectors, but the discussion the discussion remains valid in a more general case.

In mechanics usually input and outputs are loads and displacement (or velocities) and the model is usually known. The model consists of the combination of balance equation (assumed universal) and constitutive equations relating kinematic and mechanical variables (e.g. the Hooke law in elasticity relating strain and stress or the Newton law relating stress and its associated rate of strain).

When the model is assumed known, from the large catalog or dictionary of material behaviors the only needed thing is using experiments to calibrate the model, that is, for identifying the parameters involved in those constitutive equations. As it is well known, the choice is enormous, and the final predictions depend on both the quality of the chosen model and the quality of its calibration.

We could operate differently. We do not assume a model from the so-called models dictionary; we are creating it from the scratch in a different way and using a quite different representation. Both are being described in what follows. Here, the model $\mathbf{K}$ is unknown, but in exchange many input/output couples are available, i.e. $(\mathbf{I}_i, \mathbf{O}_i)$, $i = 1, \ldots, M$. In that case the problem becomes the one of calculating the model $\mathbf{K}$ from the available input/output data, assuming that all input/output couples are related by the unknown model, i.e. $\mathbf{KO}_i = \mathbf{I}_i, \forall i$.

A naive view on the problem, assumed linear, consists of rewriting the problem in the extended matrix form $\mathbb{O}\mathbb{K} = \mathbb{I}$, where $\mathbb{K}$ is the vector form of the unknown matrix $\mathbf{K}$, $\mathbb{I}$ the extended vector that concatenates all the inputs $\mathbf{I}_i$, $\forall i$, and $\mathbb{O}$ an extended matrix constructed from the outputs vectors (known assumed known) $\mathbf{O}_i$, $\forall i$. Now, if sufficient data is available, one could imagine that the model could be extracted by solving the extended linear system, $\mathbb{K} = \mathbb{O}^{-1}\mathbb{I}$ (when $\mathbb{O}$ is not inversible, its pseudo-inverse can be applied, among many other possibilities).

It is worth noting that the choice of inputs and outputs is far from being a trivial task. When Galileo studied falling bodies, he considered the distance travelled by the object after every second, that distance being the difference between its initial and present positions. Thus, by comparing these distances (5 meters travelled during the first second, 15 during the next one, then 25, etc., he observed that data followed the relation $15/5 = 3/1$, then $25/15 = 5/3$, …He thus affirmed that the consecutive traveled distances follow the prime numbers series $(1, 3, 5, \ldots)$. We must remember that at that time differential calculus was not available (it was waiting for the arrival of Newton and Leibniz!). It was without any doubt

an excellent discovery—a predictive model—, but expressed into an alternative form with respect to the nowadays usual model formats. The most important point in this discussion is not the law itself, is the fact that Galileo considered the right variable, the travelled distance and not the position itself. If he had decided to consider the position itself, very probably the deduced law could have violated the principle of Galilean invariance (frame indifference).

Nowadays, after centuries of rigorous and fruitful theoretical and applied scientific accomplishments, for the vast majority of the models employed by engineering and scientists, input and outputs are well pre-defined. However, in many other, less experienced contexts the choice is more involved. This occurs mainly when nonlinearities become history-dependent, such that they involve a number of state variables able to replace time-trajectories.

In the field of data analytics and machine learning, there are many options for constructing a model able to be used to predict outputs for given inputs. The simplest possibility consists in choosing the known output related to the closest known input. Even if it seems a straightforward alternative, it entails a major issue, the choice of the metric. This is particularly delicate when the inputs are of different nature and have very different characteristic values (strongly dependent on the considered units), significantly impacting the notion of neighborhood.

There are many other model constructors. Among them, and without the aim of being exhaustive, we would like to mention:

- Linear and nonlinear regression. Linear regression is considered mainly because of its simplicity. Its main advantage is that when P inputs (parameters) are considered, P data suffice to construct it even if more, or even less, data can also be employed. Non-linear regressions considering higher-order approximation require much more data. For example, the number of monomials involved in quadratic approximations scale with P^2 and, in general, the complexity when considering degree D scales with P^D. Thus, to circumvent the curse of dimensionality P and/or D should be reduced.

  As discussed previously, manifold learning allows considering the strictly minimum number of explicative parameters, $p \leq D$, whereas the use of separated representations (in the context of the rPGD discussed in Sect. 3) limits the effect of D [16].

  Nonlinear regression can be efficiently replaced by locally linear regression, in particular Hierarchical Bayesian Linear Regression seems especially promising [88].

  In a similar way rPGD can be replaced by a multiple local PGD-based nonlinear regression while ensuring continuity thanks to its consideration within the partition of unity (PU) framework [26].
- Decision trees and its random forest counterpart [89] have been traditionally intensively used for classifying and for constructing regressions. The rPGD discussed above aimed at conceiving a sort of fully-combinatorial decision-tree within a variational framework.
- Deep-learning based on the use of neural networks (NN) (see [90] among many others available papers and books) is probably the most powerful and most extensively used regression tool. NN employ a certain number of neuron layers, in order to account for existing couplings, and some ad hoc nonlinear behavior, and then the system is trained with the available data to finally generate a black-box model. Even if such a route is an appealing alternative when nothing a priori is known (e.g., e-commerce, sociology, psychology, marketing, etc.) in the case of engineering such a route makes it difficult to assimilate all the existing scientific acquired knowledge in the form of models. Today, significant efforts are being paid in order to render it more comprehensible from the physical point of view. A deeper understanding of its functioning is crucial to improve its efficiency (reducing the training stage) and addressing more complex phenomena and physically based complex models.

  Physics-informed Deep Learning was considered by Karniadakis and coauthors [91, 92] for data-driven solution of nonlinear PDE as well as for the discovery of nonlinear PDEs.
- Dictionary learning [93] consists in, given many events (vectors), constructing a matrix (called dictionary) so that every event must be written as a sparse linear combination of the columns in the dictionary. More precisely, assume the pairs $(\mathbf{x}_i, \mathbf{b}_i)$ collected into the columns of matrices $\mathbf{X}$ and $\mathbf{B}$ respectively. The goal is to compute $\mathbf{A}$ (the dictionary) and $\mathbf{X}$ from the knowledge of $\mathbf{B}$ in such a way that the columns of $\mathbf{X}$ are sparse. The job is successfully performed by using a variety of techniques: method of optimal directions, K-SVD or the matching pursuit algorithms, including the orthogonal variant. In a more general sense Tensor Learning is offering unexpected possibilities [94].
- Manifold learning, widely described in Sect. 4, the tSNE [95] and other described in [38], complemented with advanced clustering and classification techniques (e.g. K-means [96], Support Vector Machines—SVM—[97, 98] and the incipient powerful techniques based on Topological Data Analysis [99, 100]) are becoming unavoidable.
- Sparse identification [48] consists in assuming the search model from a general form involving many linear and nonlinear contributions (polynomial, cosinus,

exponentials, …and different combinations of them). It is expected that not all these contributions will be required for approximating the available data, and consequently sparsity is invoked.

- Dynamic model decomposition [101] proceeds from a given time series of data, by computing a set of modes each of which is associated with a fixed oscillation frequency and decay/growth rate. For linear systems these modes and frequencies are analogous to the normal modes of the system. Its extended framework by using a data-driven approximation of the Koopman operator [102] is also attracting a growing interest.
- Data-driven operator inference for nonintrusive projection-based model reduction was considered by Peherstorfer and Wilcox [50]. It infers approximations of the reduced operators from the initial conditions, inputs, trajectories of the states, and outputs of the full model, without requiring the full-model operators. Similar procedure was considered in [15] while ensuring a thermodynamic consistency.

4.4 Rationalizing the Need of Data: From Big-Data to Smart-Data

Data-driven engineering requires a huge amount of data. This constitutes one of its main drawbacks and, at the same time, one of newest and powerful characteristics. For many engineering applications, such an amount of data is sometimes not available (as opposed to many other sciences where data is often cheap to acquire but difficult to curate). In the sequel, we assume without loss of generality an elastic behavior. Thus, constructing the constitutive manifold by carrying out a sequence of homogeneous tests with the purpose of activating every possible strain states, seems out of reach for today's capabilities (hopefully it will not be so in a near future).

In our recent works, we considered an alternative approach, widely considered in the community of image correlation [103]. In this field, complex stress states are invoked during experimental campaigns. Thus, for instance, by determining the strain state in a region of the specimen we could, by applying inverse identification, unveil a large region of the constitutive manifold. The concept of constitutive manifold has been established in some of our latest works in the field [69, 72]. In them we analyzed two alternative pathways. In the first one we unveiled gradually the manifold from loading data. Therefore, at each load increment, the elastic tensor for a new strain value is determined. It should be noted, nevertheless, that such an approach revealed to be complex, partly due to the use of the elastic tensor as the main mechanical variable. It also revealed to be complex in the case of nonlinear constitutive equations. The second route consisted of constructing a polynomial approximation of the elastic energy, whose second derivative results in an elastic tensor, and whose identification from collected data seems simpler and more robust.

The establishment of the smart-data paradigm is in progress. All of us will probably agree in that, to describe the filling process of a balloon, for instance, the specification of position and momentum of every molecule is not required. It is enough to specify some macroscopic, thermodynamic variables: volume, temperature, pressure, …to describe the system. In our opinion, the big-data paradigm is analogous to fully characterize every atom. The right approach appears now clear: there is a need to create a multi-scale theory of data, that should work at equilibrium and off equilibrium. The former consists of a sort of thermodynamics of data (knowledge) and the last focuses on its transport mechanisms (information). Some attempts exist on this field [104] and researches should continue progressing.

Thus, one could expect that smart data should inform physics on the type of data to collect, where and when to do it, with the main objective of acquiring maximum information and knowledge. The era of collecting every possible datum to curate only a small percentage of them should be replaced by acquiring the right data, the one of highest quality. Collecting and treating data is expensive and takes time. It compromises real-time feed-back, which is needed for decision-making, and is indeed mandatory in many applications like video-surgery, robotics or autonomous car driving, to cite but a few.

Data rationalization can be efficiently performed by considering smart-sampling strategies. When no prior exist, Latin Hypercube techniques can be used to obtain a reasonable representation of the whole multidimensional space. This technique has been commonly considered in design of experiments—DoE—as well as to construct meta-models (surrogate models).

In the field of a posteriori MOR (POD or RB) the issue of performing better samplings was addressed to correctly drive the greedy constructor. Thus, the so-called "magic points" were proposed in the context of Reduced Based based MOR [17]. In a stochastic framework the issue of better placing the measurement points has also extensively been considered.

When using reduced basis, data assimilation easily allows data-completion. To make it simple, imagine that a given field $u(\boldsymbol{x})$ in a domain Ω, i.e., $\in\Omega$, can be expressed from a linear combination of functions $\phi_i(\boldsymbol{x})$, $i = 1, \ldots, M$, according to

$$u(\boldsymbol{x}) = \sum_{i=1}^{M} \alpha_i \phi_i(\boldsymbol{x}). \tag{27}$$

If this field is known at M particular locations $\boldsymbol{X}_j$, $u_j = u(\boldsymbol{x} = \boldsymbol{X}_j)$, we could compute the M alpha coefficients α_i. The choice of those M points should ensure the invertibility while reducing the numerical errors.Then, with those coefficients already calculated, Eq. (27) allows us to complete the solution, that is, to predict the solution at every point $\boldsymbol{x} \in \Omega$ from the mere knowledge of it at M locations.

Another family of techniques growing rapidly are related to sparse sampling [105], closely connected with compressed sensing that we summarized in what follows.

Most of nonlinear dimensionality reduction techniques consider least-squares fitting of the data, however compressed sensing is based in the use of the L^1 norm instead. As described in [106], there is a subtle link between sparsity and the use of the L^1 norm. When considering curve fitting, the use of standard L^2 norms magnifies the influence of outliers, because of the squared norm. Then the impact of those outliers in the fitted curve can thus be significant.

In the same spirit, the solution of underdetermined algebraic systems is a tricky issue, because they contains an infinite number of solutions. As illustrated in [106], the use of the pseudo inverse produces a fully populated solution vector whereas when considering the Scilab™ or Matlab™ backslash the solution contains a lot of zero entries, and then results sparse. When solving the problem with L^2 and L^1 optimizations (trying to obtain the minimum norm solution), the former becomes much less sparse than the last. In the case of overdetermined systems the same tendencies can be observed.

Thus, from a purely engineering viewpoint, L^1 can be associated to sparsity. For this reason the L^1 norm was considered as an appealing candidate for addressing signal reconstruction problems. This alleviates the Nyquist–Shannon sampling theory, that states that for recovering a signal, one must sample at twice the rate of the highest frequency involved in the signal.

Imagine a vector $\mathbf{f}$ in the usual space or time domains, and its counterpart in a domain in which it should accept a sparse representation, i.e., its vector counterpart $\mathbf{c}$ contains many zeros. Imagine for a while a single-frequency harmonic function in the time domain. Its sampling requires a number of its solution at different time instant as stated by the Nyquist–Shannon sampling theory. However, if we express it in the frequency domain, a single information suffices, the amplitude at the given frequency.

Those appealing spaces of representation, when they exist, remain unknown. Thus, in general, different choices are considered: the ones related to frequency (Fourier or discrete cosines transform) or the ones related to multi-resolution wavelets, among many other possible choices.

If we denote by $\mathbf{T}$ the matrix making the discrete transformation between both representations, the original one and the one in which the representation is expected to be sparse,

$$\mathbf{Tc} = \mathbf{f}, \tag{28}$$

since vector $\mathbf{c}$ is expected to have many zero entries (as soon as it corresponds to a space in which the signal becomes sparse), one could expect that its solution could be computed from some rows of matrix $\mathbf{T}$ and vector $\mathbf{f}$, after solving the resulting underdetermined system by making use of a L^1-norm based optimization.

The choice of such rows can be made in different ways. However, the most usual one consists in a random selection, even if nowadays many works are addressing this issue. From a matrix perspective such extraction simply consists in the definition of a diagonal matrix, with unit entries at the rows to extract. If the set of rows to extract is denoted by $\mathcal{S}$, the extraction matrix $\mathbf{E}$ is defined from

$$\begin{cases} E_{ii} = 1 & \text{if } i \in \mathcal{S} \\ E_{ij} = 0 & \text{otherwise} \end{cases}.$$

The solution of problem defined by Eq. (28) can be approximated by the solution of the underdetermined system

$$\mathbf{ETc} = \mathbf{Ef}, \tag{29}$$

using a L^1-norm based optimization.

Thus, the two main ingredients are: (1) the use of an adequate space in which the solution of the problem at hand is expected to exhibit sparsity, and (2) the solution of the underdetermined problem by using a L^1 norm.

Compressed sensing is at the origin os the so-called "single pixel camera", where instead of acquiring the global image information, i.e., the vector $\mathbf{f}$, to be then compressed, only few entries of it are acquired, i.e., $\mathbf{Ef}$, and as soon as vector $\mathbf{c}$ is calculated by solving Eq. (29), the whole field (image) can be reconstructed from Eq. (28).

5 Conclusions and Prospects

The hybrid twin, that perfectly encompasses the functionalities of its two predecessors, the so-called virtual and digital twins, consists of:

1. the pre-assumed physical contribution, efficiently addressed by using Model Order Reduction techniques;
2. a data-based modeling of the gap between predictions and measurements;

3. external actions to drive the model solution towards the desired target (control and decision making);
4. the unbiased noise filtering;

where sufficient data is required with three main aims: (1) to calibrate the physical model; (2) to construct the data-based model; and (3) to make decisions to keep the system under control and progressing to the wished target.

Control and decision making is efficiently performed by using artificial intelligence and machine learning techniques, as soon as the learning state is successfully accomplished. On the other hand, the data-based model construction can be performed:

- from the use of machine learning techniques (data-mining, regression, deep-learning, manifold learning, …as previously described);
- by expressing the deviation in a parametric form within the PGD framework by using the regression PGD—rPGD—discussed before. In this framework, data-science could be used offline to define the smartest data so be considered, and in particular, what data, and when and where they should be collected, defining the new smart-data paradigm.

It is important to note that in some circumstances the physical model is almost unattainable. Thus, the only possible contribution concerns the data-based model that is constructed from scratch by using any of the available techniques discussed in the present paper, but requiring a larger amount of "smart" data.

From the discussion addressed in the present work, some actions seem urgent to us:

1. In what concerns model order reduction, one of the main challenges is that of constructing consistent interpolations of pre-computed solutions (non-intrusive PGD) on the solution manifold so as to be able to proceed even when solutions exhibit localization. The parametric solutions of models exhibiting bifurcations is another major issue.

   Many engineering problems involve trajectories: processes (incremental forming, additive manufacturing, …), agent trajectories, etc …The issue of parametrizing a trajectory remains an open issue of major interest nowadays. Finally, reduced models of components should be integrated at the system level, and consequently efficient ROM-interfaces defined.
2. Concerning tests, the issue of unbiased and biased noise must be addressed, as well as its collection at different scales. Inverse techniques must be developed in order to have access to non-measurable variables, because of its nature or accessibility.

   In the same way that a single test is able to offer a rich amount of data (e.g., image correlation) one could imagine replacing the test machine by a computer, and expecting that by solving a problem that activates as many parametric values as possible, one could expect having access to the parametric solution from a single (few) numerical simulation(s).
3. Regarding the incipient smart-data paradigm, efforts must be paid to create a multi-scale theory of data, a sort of data-thermodynamics, that should work at equilibrium and off-equilibrium, to offer a response to four key questions: (1) what data should be collected? (2) where? (3) when? and (4) at which scale(s)?
4. For model learners and data-driven modeling, different questions arise. One of them concerns the nature of state variables (able to encapsulate all the history-dependent present state) and the way of identifying them from collected data. Another extremely exciting topic concerns the similarities between deep-learning based on neural networks and more physically based model learners as the ones discussed previously. Finally addressing noise and outliers, and differentiate them from multi-scale physically events remains also an open crucial issue.
5. Finally, concerning data and manifold learning (PCA and its nonlinear counterparts and variants), they are most of the times is based on Euclidean distances. It seems that the extraction of uncorrelated parameters from data needs alternative metrics. Looking at two trees, even a child is able to conclude on their similarity (both are recognized as trees in real-time) even if the Euclidean distance among them could be very large. In this regard, TDA (Topology Data Analysis) is attracting interest because of its appealing properties and spectacular capacity of classifying. Topology persistence, persistent homology, mappers, computational geometry, …are opening a field of unimaginable opportunities.

   Moreover, the use of persistence diagrams allows us to define metrics based on topology (of major interest when addressing shape and topology optimization) and its associated persistent images (eventually combined with sparse sensing) allows defining interpolation, a crucial aspect when addressing reduced order modeling.

   Very often, similarity must be judged and stablished outside a vector space. Imagine establishing similarity between traffic signals or color words (yellow, red, …). Identifying the similarity of words referring to color requires their transcription to a vector in a given vectorial space that allows for applying standard tools. This transcription can be successfully accomplished using Word2Vect techniques [107].

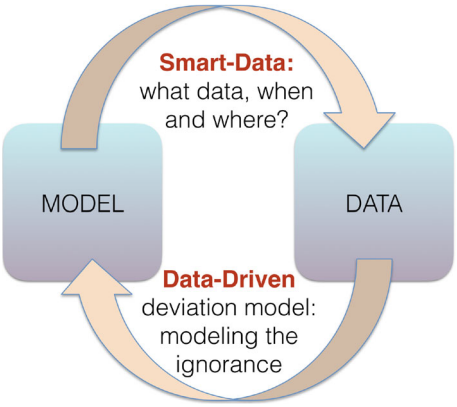


Fig. 12 Closing the models-data circle

It is at this point the dilemma of data versus models totally loses it sense. Both are not concurrent, they should be considered together, one enriching the other and vice-versa. Physics allows determining what observations should be considered when establishing a predictive data-based model while avoiding major risks, as for example, the violation of the frame invariance or thermodynamical consistency (energy conservation and entropy production). On the other hand, data-science could drive physics towards the most pertinent data offering the maximum amount of pertinent information (smart-data versus big-data). The model-data circle is definitively closed as sketched in Fig. 12.

Acknowledgements This project has received funding from the Spanish Ministry of Economy and Competitiveness through Grants number DPI2017-85139-C2-1-R and DPI2015-72365-EXP and by the Regional Government of Aragon and the European Social Fund, research group T24 17R. Special thanks to several industrial groups and companies that contributed to the approches presented in this review article, among them, ESI, Gestamp, Volkswagen, Renault, APT, Vitirover, Airbus, Ariane, Saint Gobain, Michelin, AddUp, …

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Hey T, Tansley S, Tolle K (2009) The fourth paradigm: data-intensive scientific discovery. Microsoft Research, Redmond
2. Darema F (2015) DDDAS: a key driver for large-scale-big-data and large-scale-big-computing. In: Koziel S, Leifsson L, Lees M, Krzhizhanovskaya VV, Dongarra J, Sloot PMA (eds) International conference on computational science, ICCS 2015 computational science at the gates of nature, volume 51 of Procedia computer science. Elsevier, Univ Amsterdam, NTU Singapore, Univ Tennessee, 2015. 15th annual international conference on computational science (ICCS), Reykjavik Univ, Reykjavik, ICELAND, June 01–03, 2015
3. Chinesta F, Huerta A, Rozza G, Willcox K (2017) Model reduction methods. American Cancer Society, Atlanta, pp 1–36
4. 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
5. 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
6. Chinesta F, Ladeveze P (eds) (2014) Separated representations and PGD-based model reduction. Springer, New York
7. Chinesta F, Cueto E (2014) PGD-based modeling of materials, structures and processes. Springer, Dordrecht
8. Cueto E, González D, Alfaro I (2016) Proper generalized decompositions: an introduction to computer implementation with matlab. SpringerBriefs in Applied Sciences and Technology. Springer, New York
9. Chinesta F, Keunings R, Leygue A (2014) The proper generalized decomposition for advanced numerical simulations. Springer, Dordrecht
10. 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
11. Quaranta G, Abisset-Chavanne E, Chinesta F, Duval J-L (2018) A cyber physical system approach for composite part: From smart manufacturing to predictive maintenance. In: AIP conference proceedings, 1960(1):020025
12. González D, Badías A, Alfaro I, Chinesta F, Cueto E (2017) Model order reduction for real-time data assimilation through extended Kalman filters. Comput Methods Appl Mech Eng 326(Supplement C):679–693
13. Soize C, Farhat C (2016) Nonparametric probabilistic approach of model uncertainties introduced by a projection-based nonlinear reduced-order model. In: Papadrakakis M, Papadopoulos V, Stefanou G, Plevris V (eds) ECCOMAS 2016, 7th European congress on computational methods in applied sciences and engineering, Proceedings of ECCOMAS 2016, Island of Crete, Greece, June 2016, Semi-Plenary Lecture, pp 1–26
14. Kirchdoerfer T, Ortiz M (2016) Data-driven computational mechanics. Comput Methods Appl Mech Eng 304:81–101
15. González D, Chinesta F, Cueto E (2018) Thermodynamically consistent data-driven computational mechanics. Contin Mech Thermodyn. https://doi.org/10.1007/s00161-018-0677-z
16. Ibañez R, Abisset-Chavanne E, Ammar A, González D, Cueto E, Huerta A, Duval JL, Chinesta F (2018) A multi-dimensional data-driven sparse identification technique: the sparse proper generalized decomposition. Complexity 2018:5608286. https://doi.org/10.1155/2018/5608286
17. Barrault M, Maday Y, Nguyen NC, Patera AT (2004) An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique 339(9):667–672
18. Aguado JV, Borzacchiello D, Kollepara K, Chinesta F, Huerta A (2018) Tensor representation of non-linear models using cross approximations. J Sci Comput **(in press)**
19. Ladeveze P (1999) Nonlinear computational structural mechanics. Springer, New York
20. 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
21. Bognet B, Leygue A, Chinesta F (2014) Separated representations of 3D elastic solutions in shell geometries. Adv Model Simul Eng Sci 1(1):1–34

22. Bordeu F, Ghnatios C, Boulze D, Carles B, Sireude D, Leygue A, Chinesta F (2015) Parametric 3D elastic solutions of beams involved in frame structures. Adv Aircr Spacecr Sci 2(3):233–248
23. Chinesta F, Leygue A, Bognet B, Ghnatios C, Poulhaon F, Bordeu F, Barasinski A, Poitou A, Chatel S, Maison-Le-Poec S (2014) First steps towards an advanced simulation of composites manufacturing by automated tape placement. Int J Mater Form 7(1):81–92
24. 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
25. Borzacchiello D, Aguado JV, Chinesta F (2017) Non-intrusive *sparse subspace* learning for parametrized problems. Arch Comput Methods Eng. https://doi.org/10.1007/s11831-017-9241-4
26. Ibañez R, Abisset-Chavanne E, Chinesta F, Huerta A, Cueto E (2018) A local, multiple proper generalized decomposition based on the partition of unity. Int J Numer Methods Eng **(submitted)**
27. 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 Part A Appl Sci Manuf 42(9):1169–1178
28. Nouy A (2010) Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. Arch Comput Methods Eng 17(4):403–434
29. Nadal E, Chinesta F, Díez P, Fuenmayor FJ, Denia FD (2015) Real time parameter identification and solution reconstruction from experimental data using the proper generalized decomposition. Comput Methods Appl Mech Eng 296:113–128
30. 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
31. Poulhaon F, Leygue A, Rauch M, Hascoet J-Y, Chinesta F (2014) Simulation-based adaptative toolpath generation in milling processes. Int J Mach Mach Mater 15(3–4):263–284 PMID: 60552
32. Miller RN, Carter EF, Blue ST (1999) Data assimilation into nonlinear stochastic models. Tellus A 51(2):167–194
33. Chinesta F, Abisset-Chavanne E, Aguado JA, Borzacchiello D, Lopez E, Barasinski A, Gonzalez D, Cueto E, Ghnatios C, Duval JL (2018) Big-data, machine learning, data-based models and data-driven simulations, avatars and internet of things. Boarding on the 4th industrial revolution. In: Modeles: Succes et Limites. CNRS Academie des Sciences, France
34. Badías A, Alfaro I, González D, Chinesta F, Cueto E (2018) Reduced order modeling for physically-based augmented reality. Comput Methods Appl Mech Eng 341:53–70
35. Lake BM, Ullman TD, Tenenbaum JB, Gershman SJ (2017) Building machines that learn and think like people. Behav Brain Sci 40:e253
36. González D, Cueto E, Chinesta F (2015) Computational patient avatars for surgery planning. Ann Biomed Eng 44(1):35–45
37. González D, Aguado JV, Cueto E, Abisset-Chavanne E, Chinesta F (2016) KPCA-based parametric solutions within the PGD framework. Arch Comput Methods Eng 25(1):69–86
38. Lee JA, Verleysen M (2007) Nonlinear dimensionality reduction. Springer, New York
39. Kambhatla N, Leen TK (1997) Dimension reduction by local principal component analysis. Neural Comput 9(7):1493–1516
40. Zhang Z, Zha H (2005) Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM J Sci Comput 26(1):313–338
41. Schölkopf B, Smola A, Müller K-R (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
42. Scholkopf B, Smola A, Muller KR (1999) Kernel principal component analysis. In: Advances in kernel methods—support vector learning. MIT Press, Cambridge, pp 327–352
43. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
44. Collins JP (2010) Sailing on an ocean of 0s and 1s. Science 327(5972):1455–1456
45. Holdren JP, Thomas K, Cyrus W, Laurie L (2014) Materials genome initiative strategic plan. Technical report, National Science and Technology Council
46. Mellody M (2014) Big data in materials research and development: summary of a workshop. Technical report, The National Academies Press
47. Rajan K (2005) Materials informatics. Mater Today 8(10):38–45
48. Brunton SL, Proctor JL, Nathan KJ (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. In: Proceedings of the national academy of sciences
49. Peherstorfer B, Willcox K (2015) Dynamic data-driven reduced-order models. Comput Methods Appl Mech Eng 291:21–41
50. Peherstorfer B, Willcox K (2016) Data-driven operator inference for nonintrusive projection-based model reduction. Comput Methods Appl Mech Eng 306:196–215
51. Cooreman S, Lecompte D, Sol H, Vantomme J, Debruyne D (2007) Elasto-plastic material parameter identification by inverse methods: calculation of the sensitivity matrix. Int J Solids Struct 44(13):4329–4341
52. Shi Y, Sol H, Hua H (2006) Material parameter identification of sandwich beams by an inverse method. J Sound Vib 290(3):1234–1255
53. Hartmann S, Gibmeier J, Scholtes B (2006) Experiments and material parameter identification using finite elements. Uniaxial tests and validation using instrumented indentation tests. Exp Mech 46(1):5–18
54. Mahnken R, Stein E (1996) A unified approach for parameter identification of inelastic material models in the frame of the finite element method. Comput Methods Appl Mech Eng 136(3):225–258
55. Mahnken R, Stein E (1994) The identification of parameters for visco-plastic models via finite-element methods and gradient methods. Model Simul Mater Sci Eng 2(3A):597
56. Mahnken R, Stein E (1996) Parameter identification for viscoplastic models based on analytical derivatives of a least-squares functional and stability investigations. Int J Plast 12(4):451–479
57. Mahnken R, Stein E (1997) Parameter identification for finite deformation elasto-plasticity in principal directions. Comput Methods Appl Mech Eng 147(1):17–39
58. Yvonnet J, Gonzalez D, He Q-C (2009) Numerically explicit potentials for the homogenization of nonlinear elastic heterogeneous materials. Comput Methods Appl Mech Eng 198(33):2723–2737
59. Clément A, Soize C, Yvonnet J (2012) Computational nonlinear stochastic homogenization using a nonconcurrent multiscale approach for hyperelastic heterogeneous microstructures analysis. Int J Numer Methods Eng 91(8):799–824
60. Le BA, Yvonnet J, He QC (2015) Computational homogenization of nonlinear elastic materials using neural networks. Int J Numer Methods Eng 104(12):1061–1084

61. Yvonnet J, Monteiro E, He Q-C (2013) Computational homogenization method and reduced database model for hyperelastic heterogeneous structures. Int J Multiscale Comput Eng 11(3):201–225
62. Bessa MA, Bostanabad R, Liu Z, Hu A, Apley DW, Brinson C, Chen W, Liu WK (2017) A framework for data-driven analysis of materials under uncertainty: countering the curse of dimensionality. Comput Methods Appl Mech Eng 320:633–667
63. Liu Z, Bessa MA, Liu WK (2016) Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. Comput Methods Appl Mech Eng 306:319–341
64. Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. J R Stat Soc Ser C (Appl Stat) 28(1):100–108
65. Liu Z, Fleming M, Liu WK (2018) Microstructural material database for self-consistent clustering analysis of elastoplastic strain softening materials. Comput Methods Appl Mech Eng 330:547–577
66. Kirchdoerfer T, Ortiz M (2017) Data driven computing with noisy material data sets. Comput Methods Appl Mech Eng 326:622–641
67. Kirchdoerfer T, Ortiz M (2018) Data driven computing in dynamics. Int J Numer Methods Eng 113(11):1697–1710
68. Ayensa-Jiménez J, Doweidar MH, Sanz-Herrera JA, Doblaré M (2018) A new reliability-based data-driven approach for noisy experimental data with physical constraints. Comput Methods Appl Mech Eng 328:752–774
69. Ibañez R, Abisset-Chavanne E, Aguado JV, Gonzalez D, Cueto E, Chinesta F (2018) A manifold learning approach to data-driven computational elasticity and inelasticity. Arch Comput Methods Eng 25(1):47–57
70. Ladeveze P (1985) On a family of algorithms for structural mechanics (in french). Comptes Rendus Académie des Sciences Paris 300(2):41–44
71. 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 Sciences Paris 309:1095–1099
72. Lopez E, Gonzalez D, Aguado JV, Abisset-Chavanne E, Cueto E, Binetruy C, Chinesta F (2016) A manifold learning approach for integrated computational materials engineering. Arch Comput Methods Eng 25(1):59–68
73. Ibañez 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. Comput Mech 60(5):813–826
74. Sussman T, Bathe K-J (1987) A finite element formulation for nonlinear incompressible elastic and inelastic analysis. Comput Struct 26(1):357–409
75. Latorre M, Montáns FJ (2013) Extension of the Sussman–Bathe spline-based hyperelastic model to incompressible transversely isotropic materials. Comput Struct 122:13–26
76. Latorre M, Montáns FJ (2014) What-you-prescribe-is-what-you-get orthotropic hyperelasticity. Comput Mech 53(6):1279–1298
77. Zhang M, Benítez JM, Montáns FJ (2016) Capturing yield surface evolution with a multilinear anisotropic kinematic hardening model. Int J Solids Struct 81:329–336
78. Crespo J, Latorre M, Montáns FJ (2017) Wypiwyg hyperelasticity for isotropic, compressible materials. Comput Mech 59(1):73–92
79. Latorre M, Peña E, Montáns FJ (2017) Determination and finite element validation of the wypiwyg strain energy of superficial fascia from experimental data. Ann Biomed Eng 45(3):799–810
80. Latorre M, Montáns FJ (2017) Wypiwyg hyperelasticity without inversion formula: application to passive ventricular myocardium. Comput Struct 185:47–58
81. Latorre M, Montáns FJ (2018) Experimental data reduction for hyperelasticity. Comput Struct. https://doi.org/10.1016/j.compstruc.2018.02.011
82. Grmela M, Öttinger HC (1997) Dynamics and thermodynamics of complex fluids. I. Development of a general formalism. Phys. Rev. E 56:6620–6632
83. Oettinger HC (2005) Beyond equilibrium thermodynamics. Wiley, New York
84. Laso M, Öttinger HC (1993) Calculation of viscoelastic flow using molecular models: the connffessit approach. J Non-Newton Fluid Mech 47:1–20
85. Romero I (2009) Thermodynamically consistent time-stepping algorithms for non-linear thermomechanical systems. Int J Numer Methods Eng 79(6):706–732
86. Romero I (2010) Algorithms for coupled problems that preserve symmetries and the laws of thermodynamics: Part I: monolithic integrators and their application to finite strain thermoelasticity. Comput Methods Appl Mech Eng 199(25–28):1841–1858
87. Ibanez R, Abisset-Chavanne E, Gonzalez D, Duval JL, Cueto E, Chinesta F (2018) Hybrid constitutive modeling: data-driven learning of corrections to plasticity models. Int J Mater Forming (accepted for publication),
88. Parsa B, Rajasekaran K, Meier F, Banerjee AG (2018) A hierarchical Bayesian linear regression model with local features for stochastic dynamics approximation. arXiv preprint arXiv:1807.03931
89. Criminisi A, Shotton J, Konukoglu E (2011) Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical report, Microsoft report TR-2011-114
90. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
91. Raissi M, Perdikaris P, Karniadakis GE (2017) Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations. ArXiv preprint arXiv:1711.10561
92. Raissi M, Perdikaris P, Karniadakis GE (2017) Physics informed deep learning (part II): data-driven discovery of nonlinear partial differential equations. ArXiv preprint arXiv:1711.10566
93. Moitra A (2018) Algorithmic aspects of machine learning. SIAM J Sci Comput 26(1):313–338
94. Anandkumar A, Ge R, Hsu D, Kakade SM, Telgarsky M (2014) Tensor decompositions for learning latent variable models. J Mach Learn Res 15(1):2773–2832
95. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(Nov):2579–2605
96. Forgy E (1965) Cluster analysis of multivariate data: efficiency versus interpretability of classification. Biometrics 21(3):768–769
97. Theodoridis S, Koutroumbas K (2009) Pattern recognition. Elsevier, Amsterdam
98. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines: and other kernel-based learning methods. Cambridge University Press, New York
99. Carlsson G (2014) Topological pattern recognition for point cloud data. Acta Numerica 23:289–368
100. Wasserman L (2018) Topological data analysis. Ann Rev Stat Its Appl 5(1):501–532
101. Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. J Fluid Mech 656(August):5–28
102. Williams MO, Kevrekidis IG, Rowley CW (2015) A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. J Nonlinear Sci 25(6):1307–1346
103. Neggers J, Allix O, Hild F, Roux S (2018) Big data in experimental mechanics and model order reduction: today's challenges and tomorrow's opportunities. Arch Comput Methods Eng 25(1):143–164

104. Shaw R (1981) Strange attractors, chaotic begavior, and information flow. Zeitschrift Naturforschung Teil A 36:80–112
105. Guo W, Manohar K, Brunton SL, Banerjee AG (2018) Sparse-TDA: Sparse realization of topological data analysis for multi-way classification. IEEE Trans Knowl Data Eng 30(7):1403–1408
106. Nathan Kutz J (2013) Data-driven modeling & scientific computation: methods for complex systems & big data. Oxford University Press, Oxford
107. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. CoRR arXiv: abs/1301.3781