Barabinot *et al. Adv. Model. and Simul. in Eng. Sci.(2021)8:22*
https://doi.org/10.1186/s40323-021-00207-3

Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE Open Access

Industrial Digital Twins based on the non-linear LATIN-PGD

Philippe Barabinot[1*], Ronan Scanff[1,2], Pierre Ladevèze[2], David Néron[2] and Bruno Cauville[1]

*Correspondence:
philippe.barabinot@siemens.com
[1]Siemens Industry Software SAS
92320 Châtillon, France
Full list of author information is available at the end of the article

Abstract

Digital Twins, which tend to intervene over the entire life cycle of products from early design phase to predictive maintenance through optimization processes, are increasingly emerging as an essential component in the future of industries. To reduce the computational time reduced-order modeling (ROM) methods can be useful. However, the spread of ROM methods at an industrial level is currently hampered by the difficulty of introducing them into commercial finite element software, due to the strong intrusiveness of the associated algorithms, preventing from getting robust and reliable tools all integrated in a certified product. This work tries to circumvent this issue by introducing a weakly-invasive reformulation of the LATIN-PGD method which is intended to be directly embedded into Simcenter Samcef$^{\mathrm{TM}}$ finite element software. The originality of this approach lies in the remarkably general way of doing, allowing PGD method to deal with not only a particular application but with all facilities already included in such softwares—any non-linearities, any element types, any boundary conditions...—and thus providing a new high-performance all-inclusive non-linear solver.

Keywords: Digital Twin, Reduced-order model, Non-linear, LATIN-PGD, Weakly-invasive, Simcenter Samcef$^{\mathrm{TM}}$

Introduction

The digitization of the manufacturing industry is a topic of global interest known as "Industry 4.0". Companies are prone to generate a digital representation of their entire value chain in order to be more efficient with a more rational use of resources and energy. One talks about "Digital Twins" [1,2] which include two main components: ultra-high fidelity simulations and data collected during the life of the structure [3]. Our approach is more particularly focused on the first component—namely performing ultra-high fidelity simulations very quickly—and hence we do not intend to deal with data in this paper. Despite the current strong interest in data, one of the requirements for the Digital Twin edifice is still based on the simulation part.

Reduced-order modeling (ROM) methods [4] are a key technology for the simulation component requiring a very fast and robust 3D simulation [5]. They have the advantage of drastically reducing CPU time without requiring any compromise on precision where high-fidelity solutions can always be obtained. The richness of the physical phenomena remain purely intact, only the way we are representing them—as separate variables format—differs and allows for interesting time savings. Additionally, ROM methods can lead to a better understanding of certain physical phenomena which were previously difficult or even impossible to simulate due to the curse of dimensionality.

To be used on a large scale and in a reliable way, ROM methods must be integrated as close as possible into the tools that engineers use on a daily basis, which require the joint use with industrial software. Hence, one requirement is to develop non-intrusive ROM methodologies [6]. Some recent papers are interested in this issue [7–12]. Most of the time, it results from a coupling between industrial software used as a black-box and in-house software that drives the ROM methods for parametrized problems on specific applications.

Another point lies in the management of time-dependent non-linear problems with ROM methods. In this context, the LATIN-PGD method [13,14] is a worthwhile one, which has been the subject of many works highlighting its assets: ostensible reduction of computation time [15,16], construction of virtual charts [7], ability to address all types of non-linearities [17,18] among others. Hence, the LATIN-PGD method is attractive to Siemens Digital Industries Software, as a promising tool in the Digital Twins landscape. As a software company, Siemens Digital Industries Software wants to integrate this method into the finite element industrial software Simcenter Samcef$^{\text{TM}}$ part of the Simcenter 3D$^{\text{TM}}$ package. The idea is to get the easiest possible implementation, even if it results in slightly intrusive parts at some places when necessary. This is one of the benefits of having access to the core routines of the software. Three main elements drive the approach: the robustness, the performance and the ease of use by external customers.

This work proposes a strategy to use the Proper Generalized Decomposition (PGD) method [19] directly within an industrial software capable of handling any time-dependent non-linear problem in solid mechanics. We develop a weakly-invasive version of the LATIN-PGD method, which differs from the one most commonly used in recent years based on the internal variables description of materials [13,15,20]. Instead, it can be seen as an extension of the first functional version [21–24]. Numerous numerical treatments have been made to ensure a maximum generality of the presented approach designed to deal with any non-linearity (material, contact, large deformation,...). In addition to performance aspect, robustness remains an essential requirement to be able to deal with any industrial case. While until now the LATIN-PGD method has been used to solve specific applications (see appendix in [25] for related topics), the presented approach shows the possibilities offered by the method with an industrial software where no particular application is targeted.

The novelty is the methodology put in place to use the PGD natively and as seamlessly as possible in an industrial software, thus providing an end-to-end process in one single software environment. The LATIN-PGD method treats the problem simultaneously in space and time. Regarding parameters of the Digital Twin, they are managed externally via a classical non-intrusive *snapshots* approach leveraging that the LATIN-PGD algorithm can be initialized from any previous computation [15,26]. In general, the (extra-)parameters cannot be known in advance for all possible particular applications as they vary from one to another. Hence, by robustness concerns, only a space–time PGD is managed by the Simcenter Samcef$^{\text{TM}}$ software itself because space and time are always present in non-linear computations. Then, if required, a space–time-parameters PGD can be obtained a posteriori following classical non-intrusive technics [8,10] based on *snapshots*.

Simcenter Samcef$^{\text{TM}}$ is organized as different modules to be able to simulate different physics: non-linear mechanical problems, high-speed rotating machines, articulated mechanisms or thermal simulations. As these modules are all part of the same family, it is possible to switch from one analysis to another, combine analyses or conduct co-simulations. In this work, we focus on the module dedicated to non-linear mechanical solver as an example. However, because of its generality, our approach could be used in other modules (thermal one for example) or other finite element software. The only requirement to set up the LATIN-PGD method is to have a non-linear resolution scheme already in place (Newton–Raphson), which notably solves local non-linearities. This is the strength of the proposed version to reuse without any modification the full management of non-linear equations in the local stage.

This paper focuses on the implementation—and the difficulties associated with—of the LATIN-PGD method in an industrial finite element software. Fundamentals as convergence and robustness studies will be presented in companion paper [27]. At the present time, what has been done fits into the framework of moderate displacements, outside instability zones.

In what follows, after a brief reminder of the type of mechanical problems that are considered, the LATIN-PGD strategy and some details on its implementation in Simcenter Samcef$^{\text{TM}}$ software are provided. Finally, results on two industrial test-cases are given to highlight the assets.

Weakly-invasive LATIN-PGD within Simcenter Samcef$^{\text{TM}}$ software

A wide variety of problems in many fields of physics—mechanics, thermics, rotor dynamics—can be handled with the Simcenter Samcef$^{\text{TM}}$ software. In this study, we focus on the mechanical solver capable of solving any non-linear evolutionary problem including potentially materials non-linearities, contacts or large deformations. One of the goals is to combine ROM methods (by PGD in this work) with all possible non-linearities that already exist in the software.

Types of mechanical problems involved

Following a generic manner, any physical system Ω subjected to a whole set of external sollicitations (displacements, efforts, pressures, temperature...) over a time interval $I = [0, T]$ is considered under quasi-static assumption. Using a classical finite element formulation, the problem to be solved can be systematically reduced to the following semi-discretized problem:

$$\text{Find } \boldsymbol{q}(t) \in \boldsymbol{\mathcal{Q}} \text{ such as} \quad \boldsymbol{F}_d(t) = \boldsymbol{F}_i\big(t, \boldsymbol{q}(t)\big) \qquad \forall t \in I \tag{1}$$

$$\text{with} \quad \boldsymbol{F}_i\big(t, \boldsymbol{q}(t)\big) = \boldsymbol{\mathcal{T}}\big(t, \boldsymbol{q}(\tau \leqslant t)\big) \qquad \forall t \in I \tag{2}$$

where $\boldsymbol{\mathcal{Q}}$ defines the space of kinematically admissible fields subject to sufficient regularity, $\boldsymbol{q}$ groups all the displacements unknown, $\boldsymbol{F}_d$ gathers the external loadings, $\boldsymbol{F}_i$ represents the generalized force associated with internal cohesion efforts and $\boldsymbol{\mathcal{T}}$ symbolizes a characteristic operator describing all the material non-linearities. This operator $\boldsymbol{\mathcal{T}}$ is built from the processing of local elements within the considered industrial software. It will act as a black-box in our approach as explained later.

Remark Under the notation $\mathcal{T}$, internal variables can be hidden if some visco-plastic behavioral laws are considered. Let us note in particular that there is an irreversibility in time within the various non-linearities characterized by the operator $\mathcal{T}$ *i.e.* the expression at a given time t depends on the loading history according to a functional formalism.

In case of contacts, additional contact forces $\boldsymbol{F}_c$ have to be taken into account, which requires the introduction of two additional local variables $(\boldsymbol{\nu}, \boldsymbol{\lambda})$ with $\boldsymbol{\nu}$ the trace of the displacements on the parts in contact and $\boldsymbol{\lambda}$ the Lagrange multipliers. Hence, the reference problem becomes:

$$\text{Find } \big(\boldsymbol{q}(t), \boldsymbol{\lambda}(t)\big) \text{ such as} \tag{3}$$

$$\boldsymbol{q}(t) \in \mathcal{Q} \qquad \forall t \in I \tag{4}$$

$$\boldsymbol{F}_d(t) = \boldsymbol{F}_i\big(t, \boldsymbol{q}(t)\big) + \boldsymbol{F}_c\big(t, \boldsymbol{\nu}(t), \boldsymbol{\lambda}(t)\big) \qquad \forall t \in I \tag{5}$$

$$\boldsymbol{F}_i\big(t, \boldsymbol{q}(t)\big) = \mathcal{T}\big(t, \boldsymbol{q}(\tau \leqslant t)\big) \qquad \forall t \in I \tag{6}$$

$$\mathcal{R}\big(t, \boldsymbol{\nu}(t), \boldsymbol{\lambda}(t)\big) = \boldsymbol{0} \qquad \forall t \in I \tag{7}$$

where in the finite element formulation, one can define an operator $\boldsymbol{B}$ such as $\boldsymbol{\nu} = \boldsymbol{B}\boldsymbol{q}$ and the operator $\mathcal{R}$ represents the non-linear contact equations.

This problem is classically solved incrementally in industrial software including Simcenter Samcef$^{\text{TM}}$ software. For quasi-static sollicitations, a time implicit scheme is commonly used which consists in solving for each time step a non-linear problem that can be treated by a Newton or a modified Newton–Raphson algorithm [28–30].

LATIN-PGD methodology

The novelty of our approach is to natively set up a space–time PGD method directly in an industrial finite element software (Simcenter Samcef$^{\text{TM}}$ here) in order to best integrate the reduced-order modeling tools into a unified workflow that can be easily handled by customers. One of the requirements is to integrate a new non-linear solver in the software which has to be non-incremental. We choose the non-linear LATIN solver [13,31] designed to work well with the PGD method. This is a non-incremental iterative solver producing at each iteration a better approximation of the solution over the whole space and time. While standard industrial non-linear solvers generally used are incremental by nature, we propose in this work a gateway to effortlessly implement a non-incremental solver directly within incremental one.

Schematically, this non-incremental iterative algorithm can be represented as shown in Fig. 1 where two sub-spaces $\mathbf{A}_{\boldsymbol{d}}$ and $\boldsymbol{\Gamma}$ are defined. Sub-space $\mathbf{A}_{\boldsymbol{d}}$ groups the linear equations of the problem—equilibrium and kinematic compatibility equations—while sub-space $\boldsymbol{\Gamma}$ groups the local non-linear equations—behavioral laws, contact equations... The solution of the problem (1–2) then results from the fulfillment of this iterative scheme with two search directions $\boldsymbol{\Upsilon}^{+}$ and $\boldsymbol{\Upsilon}^{-}$ representing the parameters of the method. More specifically, Eq. (1) can be identified as $\mathbf{A}_{\boldsymbol{d}}$ while Eq. (2) defines $\boldsymbol{\Gamma}$ sub-space. The unknowns in this iterative scheme are the generalized displacements $\boldsymbol{q}$ and forces $\boldsymbol{F}_i$ which differ from classical variables most recently used in the LATIN framework based on internal variables. This choice is guided by the finite element software.

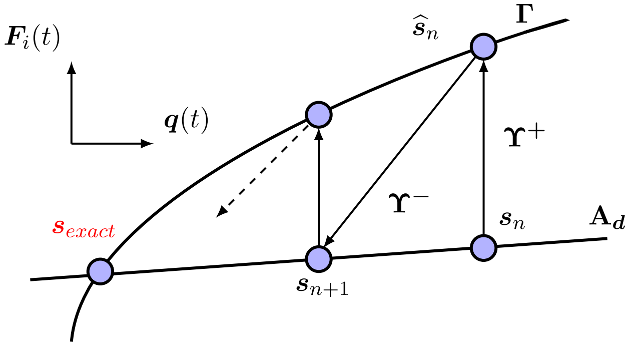

Fig. 1 Convergent iterative scheme for LATIN-PGD algorithm with $\boldsymbol{s}_n = \left(\boldsymbol{q}_n, \boldsymbol{F}_{i,n}\right)$

Usually starting from a linear elastic initialization $\boldsymbol{q}_0(t)$, the algorithm leads us through a succession of *local* and *global* stages by means of the search directions to the terms of a convergent series $\left(\boldsymbol{q}_k(t)\right)_{k\in\mathbb{N}}$. We suppose to be at a given state of the algorithm where $\boldsymbol{q}_n(t)$ is considered to be known.

Local stage: $\boldsymbol{s}_n(t) \longrightarrow \widehat{\boldsymbol{s}}_n(t)$

Given $\boldsymbol{s}_n(t)$, a new solution $\widehat{\boldsymbol{s}}_n(t)$ is sought in $\boldsymbol{\Gamma}$ sub-space by solving the non-linear equations. One has:

Constitutive relations $\boldsymbol{\Gamma}$

$$\widehat{\boldsymbol{F}}_{i,n}\big(t, \widehat{\boldsymbol{q}}_n(t)\big) = \mathcal{T}\big(t, \widehat{\boldsymbol{q}}_n(\tau \leqslant t)\big) \qquad \forall t \in I \tag{8}$$

Ascent search direction $\boldsymbol{\Upsilon}^+$

$$\widehat{\boldsymbol{q}}_n(t) = \boldsymbol{q}_n(t) \qquad \forall t \in I \tag{9}$$

We use a vertical search direction because this is what is used naturally in Newton–Raphson's algorithm. Indeed, at each iteration, one provides the displacement and one recovers the associated internal and external forces to be able to solve the equilibrium afterwards. Kinematics being imposed, this step is local in space and operates at each gauss point of each element. By making this choice for the ascent search direction, we can then reuse exactly the same routines as classically used. The only difference lies in the fact that these routines are no longer used once but several times in order to process all the time steps at each LATIN-PGD iteration.

Global stage: $\widehat{\boldsymbol{s}}_n(t) \longrightarrow \boldsymbol{s}_{n+1}(t)$

Given $\widehat{\boldsymbol{s}}_n(t)$ and an operator $\mathbf{H}$ that defines $\boldsymbol{\Upsilon}^-$ as a parameter, a new solution $\boldsymbol{s}_{n+1}(t)$ is sought in $\mathbf{A}_{\boldsymbol{d}}$ sub-space by solving the equilibrium equations. One has:

Kinematic conditions and equilibrium equation $\mathbf{A}_{\boldsymbol{d}}$

$$\boldsymbol{q}_{n+1}(t) \in \boldsymbol{\mathcal{Q}} \qquad \forall t \in I \tag{10}$$

$$\boldsymbol{F}_d(t) = \boldsymbol{F}_{i,n+1}\big(t, \boldsymbol{q}_{n+1}(t)\big) \qquad \forall t \in I \tag{11}$$

Descent search direction $\boldsymbol{\Upsilon}^-$

$$\boldsymbol{H}\big(\boldsymbol{q}_{n+1}(t)-\widehat{\boldsymbol{q}}_n(t)\big)=\underbrace{\boldsymbol{F}_{i,n+1}\big(t,\boldsymbol{q}_{n+1}(t)\big)}_{\boldsymbol{F}_d(t)}-\widehat{\boldsymbol{F}}_{i,n}\big(t,\widehat{\boldsymbol{q}}_n(t)\big)\qquad \forall t\in I \tag{12}$$

The choice of search direction operator $\boldsymbol{H}$ is the major parameter of the method. The optimal choice in terms of convergence rate would be to consider the tangent operator at each time steps, while a less costly choice could be to consider only the elastic Hooke operator.

In similar fashion, when contacts are involved, one considers $\boldsymbol{s}_n = \big(\boldsymbol{q}_n, \boldsymbol{F}_{i,n}, \boldsymbol{\nu}_n, \boldsymbol{\lambda}_n\big)$ which leads to the following LATIN scheme:

Local stage: $\boldsymbol{s_n(t)} \longrightarrow \widehat{\boldsymbol{s}}_{\boldsymbol{n}}\boldsymbol{(t)}$

Constitutive relations $\boldsymbol{\Gamma}$

$$\widehat{\boldsymbol{F}}_{i,n}\big(t,\widehat{\boldsymbol{q}}_n(t)\big)=\boldsymbol{\mathcal{T}}\big(t,\widehat{\boldsymbol{q}}_n(\tau\leqslant t)\big)\qquad \forall t\in I \tag{13}$$

$$\boldsymbol{\mathcal{R}}\big(t,\widehat{\boldsymbol{\nu}}(t),\widehat{\boldsymbol{\lambda}}(t)\big)=\boldsymbol{0}\qquad \forall t\in I \tag{14}$$

Ascent search direction $\boldsymbol{\Upsilon}^+$

$$\widehat{\boldsymbol{q}}_n(t)=\boldsymbol{q}_n(t)\qquad \forall t\in I \tag{15}$$

Global stage: $\widehat{\boldsymbol{s}}_{\boldsymbol{n}}\boldsymbol{(t)} \longrightarrow \boldsymbol{s_{n+1}(t)}$

Kinematic conditions and equilibrium equation $\boldsymbol{A_d}$

$$\boldsymbol{q}_{n+1}(t)\in\boldsymbol{\mathcal{Q}}\qquad \forall t\in I \tag{16}$$

$$\boldsymbol{F}_d(t)=\boldsymbol{F}_{i,n+1}\big(t,\boldsymbol{q}_{n+1}(t)\big)+\boldsymbol{F}_{c,n+1}\big(t,\widehat{\boldsymbol{\nu}}(t),\widehat{\boldsymbol{\lambda}}(t)\big)\qquad \forall t\in I \tag{17}$$

Descent search direction $\boldsymbol{\Upsilon}^-$

$$\boldsymbol{H}\big(\boldsymbol{q}_{n+1}(t)-\widehat{\boldsymbol{q}}_n(t)\big)=\boldsymbol{F}_{i,n+1}\big(t,\boldsymbol{q}_{n+1}(t)\big)-\widehat{\boldsymbol{F}}_{i,n}\big(t,\widehat{\boldsymbol{q}}_n(t)\big)\qquad \forall t\in I \tag{18}$$

$$\boldsymbol{\lambda}(t)-\widehat{\boldsymbol{\lambda}}(t)=\boldsymbol{h}\big(\boldsymbol{\nu}(t)-\widehat{\boldsymbol{\nu}}(t)\big)\qquad \forall t\in I \tag{19}$$

Both search directions $\boldsymbol{H}$ and $\boldsymbol{h}$ are the parameters of the method. One can interpret $\boldsymbol{h}$ as an interface rigidity [18]. The computation of the $\boldsymbol{\mathcal{R}}$ operator comes directly from the hosting software as a black-box. No particular modification is required for the local stage. In this form, the LATIN-PGD algorithm can be seen as the analogue of the Newton–Raphson algorithm to which convergent loop and time steps loop have been swapped. This is a crucial aspect in terms of ease of implementation as the method could be seamlessly integrated into Simcenter Samcef$^{\mathrm{TM}}$ software without breaking anything and without having to redevelop specific tools at this level. Thus, it is henceforth possible to choose the non-linear solver that will be used for the resolution: either Newton–Raphson or LATIN methods. In the latter case, as only linear equations on the whole space and time are involved in the global stage, one can use PGD technics [4,32].

Focus on the PGD

The PGD method [31,33] consists in approximating the correction calculated in the linear global stage by a finite sum of functions under separate variables:

$$\boldsymbol{q}_{n+1}(t)-\widehat{\boldsymbol{q}}_n(t)\approx\sum_{k=1}^{m}\lambda_k(t)\boldsymbol{\Lambda}_k \tag{20}$$

This low-rank approximation is the reduced-order basis of the problem (1–2), which is obtained using a greedy algorithm. More specifically, a first step consists in carrying out an update phase of the temporal modes already constituted, the spatial ones being kept orthogonal to each other according to the Hooke operator. Then, a second step consists in enhancing the reduced-order basis by adding some new modes if the update phase were proved to be not sufficiently effective. Thus, as the LATIN-PGD iterations progress, one comes to improve the solution until convergence. A criterion based on the energy norm averaged over the time interval enables to determine the state of convergence of the algorithm at each iteration and thus set a stop criterion when a certain threshold is reached.

Some actions are currently underway to bring out these modes directly within the post-processing graphical interface provided by Simcenter 3D$^{\text{TM}}$ platform in order to visualized them as easily as possible. The interest is twofold insofar as this data format under separated variables also allows to significantly compress the information on the outputs and therefore increases even more the execution speed of such post-processing interface.

Integration of the LATIN-PGD method into Simcenter Samcef$^{\text{TM}}$ software

From a software developer's point of view, only a few parsimonious modifications have been necessary to introduced the LATIN-PGD method in pre-existing Simcenter Samcef$^{\text{TM}}$ software notably through the use of a dedicated LATIN-PGD fortran module gathering all the new functionalities in an unique place. Everything that makes up the signature and the richness of Simcenter Samcef$^{\text{TM}}$ software—the way of treating behavioral laws, the way of managing contacts and so on—is fully recycled. In a sense, one of the strength of our approach lies in the privileged place granted to this *local* stage operating at the level of each element, each gauss point of each time steps, thus guaranteeing robustness and generality. A delicate point that has required particular attention concerns data storage since complete spatial-temporal approximations have to be stored at each LATIN-PGD iteration which is quite uncommon within incremental solvers. Nevertheless, the judicious use of pre-existing memory spaces, as well as the use of a few dynamic allocations ensure that the methodology can be carried out smoothly, taking advantage in particular of the fact that the solution can be written in a separable variables format. The result is a controlled memory increase. Seen in a macroscopic way, all this boils down to an interface management between different mathematical spaces grouping together the equations to solve, therefore to establish an efficient communication between the various parts of the code.

Another interesting thing is the notion of having a multi-fidelity solver [34]: typically in design offices, one often only needs to know an idea of the solution, a trend, especially in the upstream phases of the design. This allows a better understanding of the overall functioning of a new system before establishing afterwards which studies should then be carried out in greater depth by calculating high-fidelity solutions, always at low cost.

Management of parametrized aspects

In an industrial environment, it is not uncommon to perform these non-linear space–time computations several times depending on some sets of parameters. The notion of parameter should be taken in a broad sense. By parameter, we mean for instance material data, boundary conditions, number of cycles, geometry variations (morphing),... In what follows, these parameters are managed in a completely external way taking into account an interesting property of the LATIN-PGD method: the algorithm can be initialized with any previous kinematically admissible field. Hence, we do not explicitly refer to the parameters in the notations although our methodology is designed to deal with models involving parameters as illustrated in the numerical results. There is nothing new here, we use the classical techniques as stated in [15,26]. More precisely, if one notes $\boldsymbol{\mu}$ the set of p parameters $\boldsymbol{\mu} = (\mu_1, ..., \mu_p)$, then each LATIN-PGD computation with Simcenter Samcef$^{\text{TM}}$ software provides a space–time PGD approximation $\boldsymbol{q}^{(\boldsymbol{\mu})}$ for one value of the parameters such as:

$$\boldsymbol{q}^{(\boldsymbol{\mu})}(t) = \sum_{k=1}^{m^{(\boldsymbol{\mu})}} \xi_k^{(\boldsymbol{\mu})}(t)\, \boldsymbol{\Xi}_k^{(\boldsymbol{\mu})} \tag{21}$$

Next, if desired, one can then compress the information from these different reduced-order basis to reconstitute a posteriori a space–time-parameters PGD [8,10]:

$$\boldsymbol{q}(t, \boldsymbol{\mu}) = \sum_{k=1}^{m} \xi_k(t)\, \boldsymbol{\Xi}_k \prod_{i=1}^{p} \omega_k(\mu_i) \tag{22}$$

Some industrial illustrations

In the following, two industrial test-cases are presented. They both highlight, on the one hand, the possibilities of the LATIN-PGD method incorporating within Simcenter Samcef$^{\text{TM}}$ finite element software to build efficiently reduced-order basis and, on the other hand, the gain in computation time compared to classical approach.

Combustion engine piston

The first test-case deals with a combustion engine piston submitted to pressure forces on the upper part. Due to the symmetry of the problem, both in terms of loadings and geometry, only a quarter of the piston is modeled and the boundary conditions are set accordingly. In addition, since the piston can only move vertically in its chamber, zero normal boundary conditions are fixed on the outer cylindrical part. Finally, one leads to the resolution of an evolutionary mechanical problem involving 1 million of degrees of freedom, subjected to cyclic pressure loading in order to simulate the back and forth motion of the piston. The material is non-linear Norton law whose equivalent cumulative plastic strain rate $\dot{p}$ follows the formula:

$$\dot{p} = \left(\frac{\langle J_2(\boldsymbol{\sigma}) - k \rangle}{K p^{1/m}} \right)^n \tag{23}$$

where $J_2(\boldsymbol{\sigma})$ represents the equivalent Von Mises stress, k enables to take into account an isotropic hardening and (K, n, m) are three parameters depending on the temperature T of the combustion chamber which is assumed to be uniform for a sake of simplicity. We propose in the following to carry out a parametric study in order to determine the whole space–time solution according to different values of the temperature. In particular, we take out the curve showing the evolution of the cumulative plastic strain during cyclic

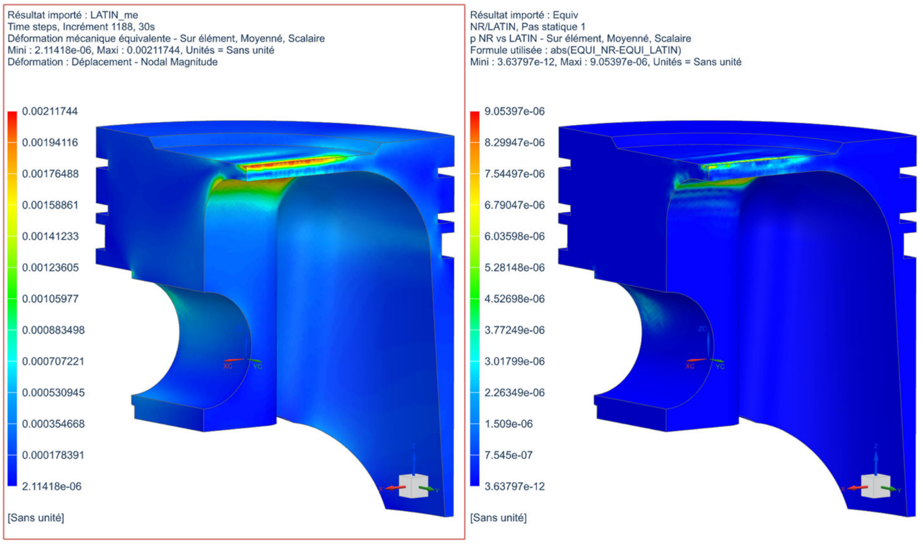
Résultat importé : LATIN_me
Time steps, Incrément 1188, 30s
Déformation mécanique équivalente - Sur élément, Moyenné, Scalaire
Mini : 2.11418e-06, Maxi : 0.00211744, Unités = Sans unité
Déformation : Déplacement - Nodal Magnitude
0.00211744
0.00194116
0.00176488
0.00158861
0.00141233
0.00123605
0.00105977
0.000883498
0.000707221
0.000530945
0.000354668
0.000178391
2.11418e-06
[Sans unité]
Résultat importé : Equiv
NR/LATIN, Pas statique 1
p NR vs LATIN - Sur élément, Moyenné, Scalaire
Formule utilisée : abs(EQUI_NR-EQUI_LATIN)
Mini : 3.63797e-12, Maxi : 9.05397e-06, Unités = Sans unité
9.05397e-06
8.29947e-06
7.54497e-06
6.79047e-06
6.03598e-06
5.28148e-06
4.52698e-06
3.77249e-06
3.01799e-06
2.26349e-06
1.509e-06
7.545e-07
3.63797e-12
[Sans unité]

Fig. 2 Cumulative plastic strain output with LATIN-PGD method (left) and deviation from the reference (right)

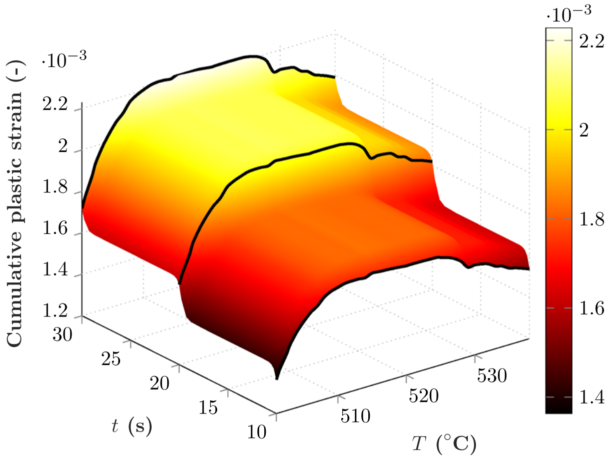

Fig. 3 Temporal evolution of the equivalent cumulative plastic strain according to temperature T for the most loaded element

loading resulting. This curve results from 40 computations. As an example, experiments are achieved on 3 loading cycles but more cycles could have been set.

Figure 2 gives the spatial distribution of the cumulative plastic strain at the end of the third cycle where the first figure on the left shows the output with LATIN-PGD method and the second figure on the right displays the deviation from the reference calculated with classical Newton–Raphson scheme. Thus, with only 9 modes, the low-rank approximation of the LATIN-PGD method is able to perfectly capture the high-fidelity solution as the error does not exceed half a percent on all elements.

On Figs. 3 and 4 evolution of the cumulative plastic strain along the time steps are exhibited. All these calculations have been carried out to achieve a high fidelity solution. While almost 22 h are required to achieve all simulations with standard Simcenter Samcef$^{\text{TM}}$

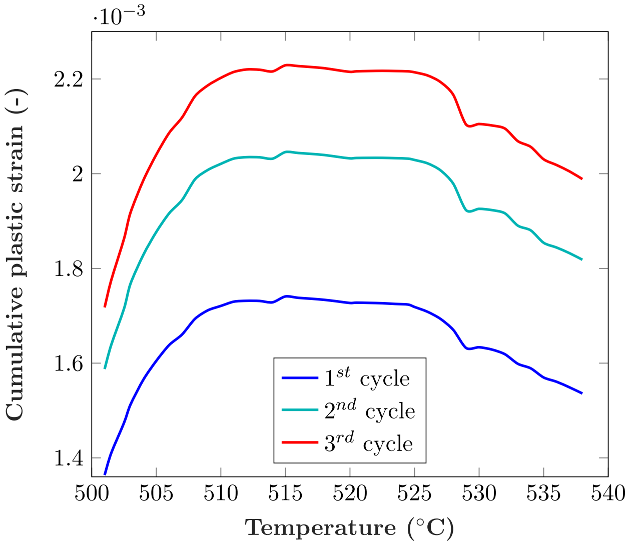

Fig. 4 Equivalent cumulative plastic strain at the end of each cycle according to temperature *T* on the most loaded element of the structure

software, only 7 h are necessary to carry out the entire parametric study with reduced-order model which leads to a speed-up of 3 times faster. Moreover, there is a great disparity in the calculation time: while it remains more or less constant at around 30 min per run with the classical approach, it can vary over a range from 4 to 18 min with new LATIN-PGD method depending on the initialization of the solution. Here, a rather naive path have been pursued consisting in starting from the closest neighbor in the parametric space which proves to be very effective when the gradient evolution remains relatively low (typically the range from 515 to 525 °C) and requires a little more time otherwise. In [26] one can find strategies to follow for a more judicious selection of the path. However, we note that we always outperform the classical resolution except for the first calculation which required 50 min—the time to build a first reduced-order basis through PGD from an elastic-solution. Of course, one could probably gain even more by implementing a first phase of estimation or extrapolation of the reduced-order basis as an initialization of the algorithm for a new calculation. The better the estimation, the shorter the computation time, but the fact remains that a reliable high-fidelity solution will always be obtained by proceeding again a few iterations with the LATIN-PGD algorithm if necessary.

Turbine blade-disk assembly

The second test-case concerns a turbine blade-disk assembly involving material non-linearities—plasticity and creep—and contact non-linearities at the junction between the disk and the blade. The assembly is subjected to centrifugal forces as well as pressure ones on the surface of the blade. Additionally, a temperature gradient is applied according to the radial coordinate. Due to the rotating machine loads and geometry, cyclic symmetry conditions are also prescribed on the lateral parts. This geometry consists of 1 million of degrees of freedom, including 40,000 Lagrange multipliers, resulting from different types of volumic elements—tetrahedrons or hexahedrons—and specific elements at the interfaces between the different parts of the mesh to write kinematic conditions. Because

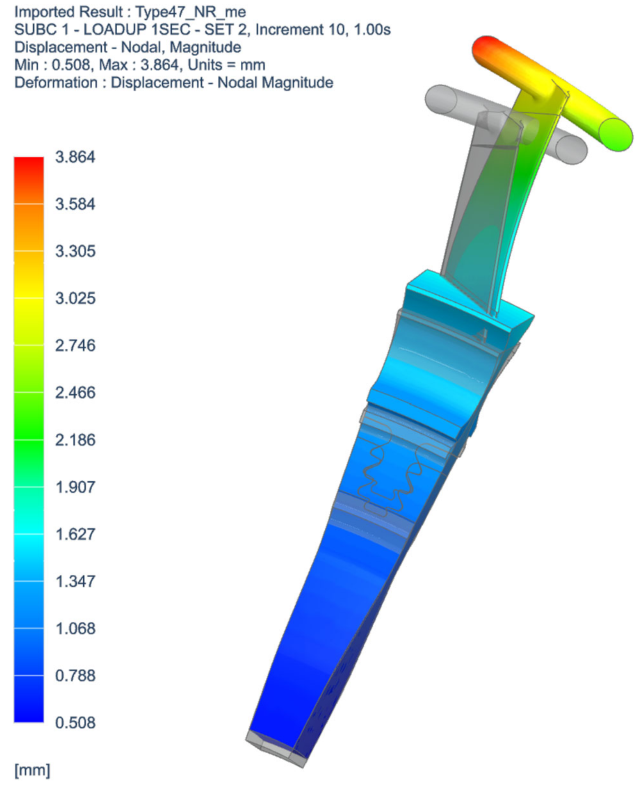
Imported Result : Type47_NR_me
SUBC 1 - LOADUP 1SEC - SET 2, Increment 10, 1.00s
Displacement - Nodal, Magnitude
Min : 0.508, Max : 3.864, Units = mm
Deformation : Displacement - Nodal Magnitude
3.864
3.584
3.305
3.025
2.746
2.466
2.186
1.907
1.627
1.347
1.068
0.788
0.508
[mm]

Fig. 5 Initial configuration (gray) and deformed geometry (color) for the most loading time step

of its relatively large size (about 200 mm) and the loadings, the blade unscrews as illustrated on Figure 5, where one can appreciate the deformed geometry in color with respect to the initial configuration in gray. A maximum displacement of 3.86 mm is observed at the end of the structure.

In what follows, a two parameters study is undertaken where the parameters are the Young modulus E and the thermal expansion coefficient α of the blade. Values of these parameters are assumed to be insufficiently known, mainly related to the manufacturing process. Due to loading, a small variation in the value of these coefficients can affect the maximum Von Mises stress in the contact area where creep occurs over long periods of time, which can damage the structure faster than expected. Although other parameters may be of influence, we focus on these two parameters for a sake of simplicity. Figure 6 presents the stress map obtained for a given point ($E = 220$ MPa and $\alpha = 14 \cdot 10^{-6}$ K^{-1}) in the parametric space. One gives the results of the LATIN-PGD method included in Simcenter Samcef$^{\mathrm{TM}}$ software and the remaining error with respect to the reference. The latter is taken as the classical solution from the commercial software (using Newton–Raphson method). One can note a very good concordance of the solutions as the relative error remains less than one percent in the area of interest for 10 PGD modes.

Figure 7 shows the response surface of the maximum Von Mises stress in the contact area according to the two parameters. This chart highlights a non-linear variation according to the Young modulus E and an almost linear one on the second parameter α. In order to achieve this, 70 computations have been carried out by scanning the parametric

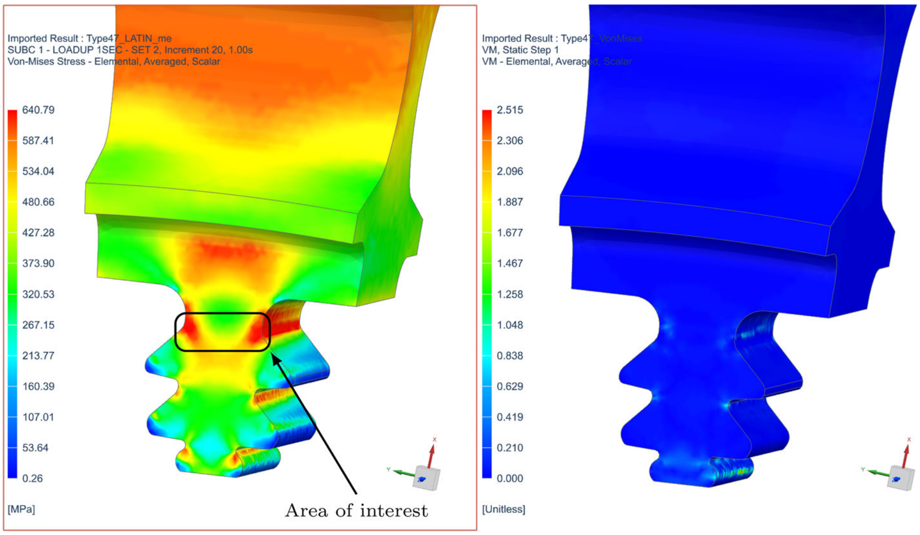

Fig. 6 Von Mises stress map [MPa] with the LATIN-PGD method (left) and deviation from the reference in percent [%] (right)

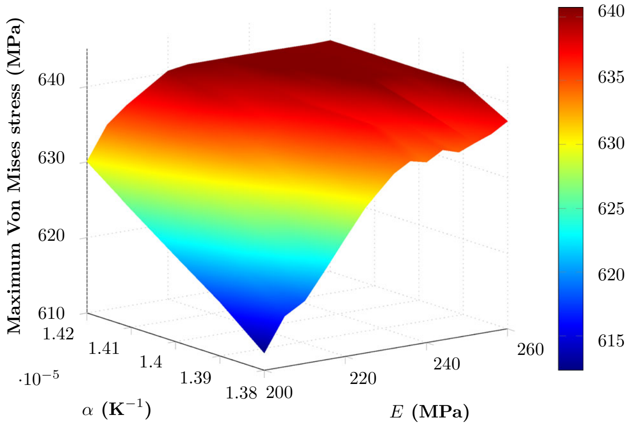

Fig. 7 Maximum Von Mises stress in the contact area according to both E and $\boldsymbol{\alpha}$ parameters resulting from 70 parametric points uniformly pulled across the parametric space

space sequentially in a uniform manner. Thanks to both ROM methods and LATIN-PGD algorithm allowing to restart the computation from a close previous converged point in the parametric space, only 3 h have been necessary to build the virtual chart while more than 21 h would have been necessary using classical Simcenter Samcef$^{\mathrm{TM}}$ software based solely on Newton–Raphson algorithm. This leads to 7 times faster computation.

This test-case highlights one of the advantages having the LATIN-PGD method incorporated natively within an industrial software insofar as one can then exploit all the capabilities of the hosting software at the local stage. As an example, 3 different behavior laws (depending on the temperature) have been considered here, as well as different types of elements, in addition to the contact behavior.

Conclusions

In this work, the LATIN-PGD method have been directly implemented into Simcenter Samcef$^{\mathrm{TM}}$ finite element software, which allows to build and then enrich reduced-order model as desired at the industrial level. Here, a very general way of looking is adopted with PGD by making the best use of all the local behavior pre-existing in industrial softwares. The PGD method becomes in fact integrated as close as possible to the solver, enabling to combine performance, robustness and a wide range of applications that can be carried out. Moreover, a first time saving on parametric studies has been demonstrated but a significantly additional gain could still be made by exploiting hyper-reduction techniques [35–37] and in particular the Reference Point Method [14,38] well suited for LATIN-PGD method. For each quantity of interest, the latter consists in storing the information in an even more condensed manner at a few cleverly chosen points, an analytical formula being subsequently used to reconstruct the field over the whole space and time. These methods are currently being investigated and should enable to go further in the reduction of computation time insofar as up to two thirds of the computation time can be spent in the local stage reflecting the fact that the current algorithm complexity is not independent of the number of degrees of freedom. Furthermore, one also keep in mind that the gain is directly correlated to the complexity of the problem: thus, considering a dozen of parameters and several hundreds or thousands of loading cycles, the performance would be even more better. Examples on material and contact non-linearities have been presented above, but this very general methodology can deal with any other type of non-linearity already present in the industrial Simcenter Samcef$^{\mathrm{TM}}$ software. This is the strength of this approach to be able to treat any type of finite element, any geometry, any boundary condition, any loading or even any non-linearities all using reduced-order modeling methods in order to circumvent prohibitive computation times. However, at the present time, computations are limited to moderate displacements, outside instability zones. One of our next task will be to overcome this limitation.

Acknowledgements

This work is funded by Siemens Industry Software SAS and Association Nationale de la Recherche et de la Technologie (ANRT). Authors also acknowledge the support of the CNRS research group 3587 AMORE.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Siemens Industry Software SAS 92320 Châtillon, France,

[2]Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190 Gif-sur-Yvette, France.

Received: 2 February 2021 Accepted: 24 August 2021
Published online: 04 October 2021

References

1. Chinesta F, Cueto E, Abisset-Chavanne E, Duval JL, Khaldi FE. Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. Arch Comput Methods Eng. 2020;27(1):105–34. https://doi.org/10.1007/s11831-018-9301-4.
2. Hartmann D, van der Auweraer H. Digital twins [cs]. 2020. arXiv:2001.09747.
3. Glaessgen E, Stargel D. The digital twin paradigm for future NASA and US Air Force vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2012-1818.
4. Benner P, Grivet-Talocia S, Quarteroni A, Rozza G, Schilders W, Silveira LM, editors. Model order reduction: snapshot-based methods and algorithms, vol. 2. Berlin: De Gruyter; 2021.
5. Chinesta F, Huerta A, Rozza G, Willcox K. Model reduction methods. In: Encyclopedia of computational mechanics, vol. 3. 2nd ed. New York: American Cancer Society; 2017. p. 1–36. https://doi.org/10.1002/9781119176817.ecm2110.
6. Giraldi L, Litvinenko A, Liu D, Matthies HG, Nouy A. To be or not to be intrusive? The solution of parametric and stochastic equations-the "plain vanilla" Galerkin case. SIAM J Sci Comput. 2014;36(6):2720–44. https://doi.org/10.1137/130942802.
7. Courard A, Néron D, Ladevèze P, Ballere L. Integration of PGD-virtual charts into an engineering design process. Comput Mech. 2016;57(4):637–51. https://doi.org/10.1007/s00466-015-1246-y.
8. Ibáñez R, Abisset-Chavanne E, Ammar A, González D, Cueto E, Huerta A, Duval JL, Chinesta F. A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition. Complexity. 2018. https://doi.org/10.1155/2018/5608286.
9. Zou X, Conti M, Díez P, Auricchio F. A nonintrusive proper generalized decomposition scheme with application in biomechanics. Int J Numer Methods Eng. 2018;113(2):230–51. https://doi.org/10.1002/nme.5610.
10. Borzacchiello D, Aguado JV, Chinesta F. Non-intrusive sparse subspace learning for parametrized problems. Arch Comput Methods Eng. 2019;26(2):303–26. https://doi.org/10.1007/s11831-017-9241-4.
11. Casenave F, Akkari N, Bordeu F, Rey C, Ryckelynck D. A nonintrusive distributed reduced-order modeling framework for nonlinear structural mechanics—application to elastoviscoplastic computations. Int J Numer Methods Eng. 2020;121(1):32–53. https://doi.org/10.1002/nme.6187.
12. Tsiolakis V, Giacomini M, Sevilla R, Othmer C, Huerta A. Nonintrusive proper generalised decomposition for parameterised incompressible flow problems in OpenFOAM. Comput Phys Commun. 2020;249:107013. https://doi.org/10.1016/j.cpc.2019.107013.
13. Ladevèze P. Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation. New York: Springer; 1999.
14. Ladevèze P. On reduced models in nonlinear solid mechanics. Eur J Mech A/Solids. 2016;60:227–37. https://doi.org/10.1016/j.euromechsol.2016.08.005.
15. Relun N, Néron D, Boucard PA. A model reduction technique based on the PGD for elastic-viscoplastic computational analysis. Comput Mech. 2013;51(1):83–92. https://doi.org/10.1007/s00466-012-0706-x.
16. Néron D, Boucard P-A, Relun N. Time-space PGD for the rapid solution of 3D nonlinear parametrized problems in the many-query context. Int J Numer Methods Eng. 2015;103(4):275–92. https://doi.org/10.1002/nme.4893.
17. Abdali A, Benkrid K, Bussy P. Simulation of sheet cutting by the large time increment method. J Mater Process Technol. 1996;60(1):255–60. https://doi.org/10.1016/0924-0136(96)02339-4.
18. Oumaziz P, Gosselet P, Boucard P-A, Guinard S. A parallel non-invasive mixed domain decomposition—implementation and applications to mechanical assemblies. Finite Elem Anal Des. 2019;156:24–33. https://doi.org/10.1016/j.finel.2019.01.004.
19. Chinesta F, Ladevèze P. Proper generalized decomposition. In: Model order reduction: snapshot-based methods and algorithms, vol. 2. Berlin: De Gruyter; 2020. p. 97–138. https://doi.org/10.1515/9783110671490-003.
20. Bhattacharyya M, Fau A, Nackenhorst U, Néron D, Ladevèze P. A LATIN-based model reduction approach for the simulation of cycling damage. Comput Mech. 2018;62(4):725–43. https://doi.org/10.1007/s00466-017-1523-z.
21. Ladevèze P. Sur une famille d'algorithmes en mécanique des structures. Comptes-rendus des séances de l'Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l'univers, sciences de la terre. 1985;300(2):41–4.
22. Boisse P, Bussy P, Ladevèze P. A new approach in non-linear mechanics: the large time increment method. Int J Numer Methods Eng. 1990;29(3):647–63. https://doi.org/10.1002/nme.1620290312.

23. Bussy P, Rougée P, Vauchez P. The large time increment method for numerical simulation of metal forming processes. In: NUMETA. New York: Elsevier Science Ltd; 1990. p. 102–9.
24. Cognard J-Y, Ladevèze P. The large time increment method applied to cyclic loadings. In: Życzkowski M, editor. Creep in structures. Berlin: Springer; 1991. p. 555–62. https://doi.org/10.1007/978-3-642-84455-362.
25. Scanff R, Nachar S, Boucard P-A, Néron D. A study on the LATIN-pgd method: analysis of some variants in the light of the latest developments. Arch Comput Methods Eng. 2020. https://doi.org/10.1007/s11831-020-09514-1.
26. Heyberger C, Boucard P-A, Néron D. A rational strategy for the resolution of parametrized problems in the PGD framework. Comput Methods Appl Mech Eng. 2013;259:40–9. https://doi.org/10.1016/j.cma.2013.03.002.
27. Scanff R, Néron D, Ladevèze P, Barabinot P, Cugnon F, Delsemme J-P. Weakly-invasive LATIN-PGD for solving time-dependent non-linear parametrized problems in solid mechanics; 2021. (**In preparation**).
28. Zienkiewicz OC, Taylor RL. The finite element method for solid and structural mechanics. New York: Elsevier; 2005.
29. de Borst R, Crisfield MA, Remmers JJC, Verhoosel CV. Nonlinear finite element analysis of solids and structures. Chichester: Wiley; 2012.
30. Bonnet M, Frangi A, Rey C. The finite element method in solid mechanics. New York: McGraw Hill Education; 2014.
31. Ladevèze P. PGD in linear and nonlinear computational solid mechanics. In: Chinesta F, Ladevèze P, editors. Separated representations and PGD-based model reduction: fundamentals and applications. Vienna: Springer; 2014. p. 91–152. https://doi.org/10.1007/978-3-7091-1794-1_3.
32. Chinesta F, Ladevèze P, editors. Separated representations and PGD-based model reduction: fundamentals and applications. Courses and Lectures/International Centre for Mechanical Sciences, vol. 554. Wien: Springer; 2014.
33. Chinesta F, Keunings R, Leygue A. The proper generalized decomposition for advanced numerical simulations. SpringerBriefs in applied sciences and technology. Cham: Springer Science & Business Media; 2014. https://doi.org/10.1007/978-3-319-02865-1.
34. Nachar S, Boucard P-A, Néron D, Rey C. Multi-fidelity Bayesian optimization using model-order reduction for viscoplastic structures. Finite Elem Anal Des. 2020;176:103400. https://doi.org/10.1016/j.finel.2020.103400.
35. Barrault M, Maday Y, Nguyen NC, Patera AT. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique. 2004;339(9):667–72. https://doi.org/10.1016/j.crma.2004.08.006.
36. Ryckelynck D. A priori hyperreduction method: an adaptive approach. J Comput Phys. 2005;202(1):346–66. https://doi.org/10.1016/j.jcp.2004.07.015.
37. Maday Y, Mula O. A generalized empirical interpolation method: application of reduced basis techniques to data assimilation. In: Brezzi F, Colli Franzone P, Gianazza U, Gilardi G, editors. Analysis and numerics of partial differential equations. Springer INdAM Series Springer INdAM Series Springer INdAM Series. Milano: Springer Milan; 2013. p. 221–35. https://doi.org/10.1007/978-88-470-2592-9_13.
38. Capaldo M, Guidault P-A, Néron D, Ladevèze P. The reference point method, a "hyperreduction" technique: application to PGD-based nonlinear model reduction. Comput Methods Appl Mech Eng. 2017;322:483–514. https://doi.org/10.1016/j.cma.2017.04.033.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.