



HAL
open science

Weakly-invasive LATIN-PGD for solving time-dependent non-linear parametrized problems in solid mechanics

Ronan Scanff, David Néron, Pierre Ladevèze, Philippe Barabinot, Frédéric Cugnon, Jean-Pierre Delsemme

► **To cite this version:**

Ronan Scanff, David Néron, Pierre Ladevèze, Philippe Barabinot, Frédéric Cugnon, et al.. Weakly-invasive LATIN-PGD for solving time-dependent non-linear parametrized problems in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 2022, 396, pp.114999. 10.1016/j.cma.2022.114999 . hal-03676386

HAL Id: hal-03676386

<https://hal.science/hal-03676386>

Submitted on 19 May 2024

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.

Weakly-invasive LATIN-PGD for solving time-dependent non-linear parametrized problems in solid mechanics

Ronan Scanff^{a,b,*}, David Néron^b, Pierre Ladevèze^b, Philippe Barabinot^b, Frédéric Cugnon^c,
Jean-Pierre Delsemme^c

^a*Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190 Gif-sur-Yvette, France*

^b*Siemens Industry Software SAS, 92320 Châtillon, France*

^c*Siemens Industry Software NV, 4031 Liège, Belgique*

Abstract

A weakly-invasive version of the LATIN-PGD method is introduced to compute reduced-order models in solid mechanics for time-dependent non-linear parametrized problems under quasi-static assumption. Its main interest is to be easily implemented in any general-purpose industrial finite element software while taking advantage of all facilities offered, including the ability to handle any kind of non-linearities. This way of proceeding provides unified tools – for the construction of reduced-order models in non-linear context – all integrated in one certified product in consistency with the purpose of having end-to-end processes. Possibilities of our approach are illustrated thanks to the first implementation conducted within Simcenter SamcefTM software developed by Siemens Digital Industries Software without ever redeveloping any non-linear part of the software. Finally, interesting performance gains are highlighted on a non-linear time-dependent test-case involving some parameters.

Keywords: LATIN-PGD, Reduced-order model, Weakly-invasive, Non-linear, Time-dependent, Visco-plasticity

1. Introduction

Driven by the ever-growing expectations of a digitalizing industry in the context of *Industry 4.0*, simulations are becoming more and more widespread. However, despite the advent of gigantic clusters, many non-linear simulations – generally assorted with the exploration of high-dimensional parametric spaces during optimization processes – are still unreachable, requiring the development of new software methods applicable at an industrial level.

*Corresponding author

Email addresses: ronan.scanff@ens-paris-saclay.fr, ronan.scanff@siemens.com (Ronan Scanff), david.neron@ens-paris-saclay.fr (David Néron), pierre.ladeveze@ens-paris-saclay.fr (Pierre Ladevèze), philippe.barabinot@siemens.com (Philippe Barabinot), frederic.cugnon@siemens.com (Frédéric Cugnon), jean-pierre.delsemme@siemens.com (Jean-Pierre Delsemme)

Reduced-Order Modeling (ROM) methods are part of the tools developed to circumvent this issue. Reduced-order models highlight the relevant information and allow robust low cost analyses especially over the parameter space. The book [1] gives a recent and detailed overview of the state-of-the-art about different ROM methods. Given some parametrized PDEs $\mathcal{L}(\mathbf{u}; t, \boldsymbol{\mu})$ defined over $\Omega \times I$ where Ω represents the space domain, I the time interval and $\boldsymbol{\mu}$ is the collection of p parameters $(\mu_1, \mu_2, \dots, \mu_p)$, all these methods start with the following low-rank separated format approximation of the physical field:

$$\mathbf{u}(\mathbf{x}, t; \mu_1, \mu_2, \dots, \mu_p) \approx \sum_{k=1}^m \boldsymbol{\Lambda}_k(\mathbf{x}) \lambda_k(t) \prod_{i=1}^p \lambda_k^i(\mu_i) \quad (1)$$

and therefore only differ in the methodology used to construct the m modes summation. The Proper Orthogonal Decomposition (POD) [2, 3] and the Reduced Basis (RB) [4, 5] are characterized by a preliminary learning stage defining the *snapshots* resulting from high-fidelity solutions. The Proper Generalized Decomposition (PGD) [6, 7] follows a different path in the sense that the m modes are directly computed on-the-fly thanks to a minimization process.

Although ROM methods have been widely studied in research areas, their industrial deployment remains relatively limited. In particular, ROM methods must be accessible to non-expert users which requires their integration as effortlessly as possible into the habits of design offices, involving commercial software. As far as authors knowledge, ROM methods are still not implemented in general-purpose industrial finite element software and therefore do not belong to the everyday tools of the engineers. The main hindrance consists in the non-intrusiveness nature [8, 9] of such methods, which requires unconventional operations and storage format. It can be noticed that most of the non-intrusive ROM methods are generally based on a *a posteriori* POD or RB approaches [10, 11, 12, 13, 14]. The PGD is inherently more intrusive because of its *a priori* formulation. Recently, some efforts have been made to achieve the PGD in a non-intrusive manner where two main approaches can be identified. Based also on *snapshots*, one can first find the non-intrusive Sparse Subspace Learning (SSL) [15] or the sparse PGD (sPGD) [16] to solve parametrized problems as crash analysis. Alternatively, for very specific applications, some dedicated non-intrusive implementation of the PGD can be found which rely on the external use of industrial software: for shape optimization with Simcenter SamcefTM [17], for linear problem in biomechanics with AbaqusTM [18], for applications to the mechanical modeling of 3D woven fabrics or composite forming processes with ESI software [19, 20], for parametric solutions of inertia relief problems with MSC-NastranTM [21], or for the approximation of parametrized laminar incompressible Navier-Stokes equations with OpenFOAMTM [22]. However, in these previous cases, the PGD remains external to the industrial software and usually requires coupling efforts with other software such as MatlabTM or PythonTM.

This work deals with time-dependent non-linear problems in solid mechanics which are classically solved by general-purpose industrial finite element software. We develop a weakly-invasive version of the LATIN-PGD method, whose principle is given in [23], which can be smoothly embedded into any existing general-purpose industrial finite element software. This new version remains still based on the separation over $\Omega \times I$ between equilibrium equations and constitutive relations, but the state at time t is defined by means of generalized quantities (\mathbf{u}, \mathbf{f}) – generalized displacements and forces respectively. The formulation of the constitutive relations is very general, the generalized forces \mathbf{f} at time t being supposed to be a function of the displacement history until time t . This proposed weakly-invasive LATIN-PGD version differs from the today version, which exploits the

internal variable description of the material usually used [24, 25], but can be seen as an extension of the first version of the LATIN-PGD [26, 7]. Some specific numerical treatments have also been introduced to get a robust and efficient ROM method – robustness being the most essential property for use in industrial software. A first implementation has been achieved within Simcenter SamcefTM software enabling to use ROM methods directly at the industrial software level then benefiting from all the tools already present in the code: any sort of non-linearities (behavioral laws, contact, etc.), any type of elements, any kind of external loading and so on. The major positive point arises from there is no further need to redevelop *ad-hoc* or *in-house* external codes and no need to redevelop parts of non-linear codes such as the integration of material laws. Here, we use the industrial software as it is, so we can make a full usage of the underlying optimized procedures which defines a very robust strategy. Its possibilities and performances are illustrated through test-cases involving industrial time-dependent non-linear problems with parameters. For the moment, situations with moderate displacements, outside instability zones, are well managed. Large displacements and instability phenomena will be addressed later.

This work comes as a complement to our former article [27] which illustrated a industrial viewpoint and showed the practical potential of PGD as a reduced-order model tool for engineers who wish to use it in the general-purpose finite element software Simcenter Samcef. This work addresses the more fundamental aspects of the proposed weakly-invasive LATIN-PGD methodology as convergence and robustness concerns. In particular, as one of the major parameter of the method, the optimization of the search direction is being examined. It should also be noted that although the approach remains illustrated within Simcenter SamcefTM software, the general methodology depicts in this paper could be implemented in any general-purpose industrial finite element software.

The remaining part of this paper is organized as follows: Section-2 specifies the reference problem. Section-3 provides the details of the new weakly-invasive LATIN-PGD method. Section-4 is devoted to the presentation of some results based on an industrial test-case highlighting gains in computational time. Finally, some conclusions and perspectives are underlined in Section-5.

2. The reference problem

One considers the standard problem solved by any general-purpose industrial finite element software. Hence, defining any geometry, denoted by the domain $\Omega \subset \mathbb{R}^3$, the studied structure can be subjected to different kind of sollicitations – prescribed displacements, body forces, surface efforts, thermal loadings, among others – over a time interval $I \subset \mathbb{R}_+$. Time-dependent non-linearities may also arise from several sources: contacts, material such as (visco)-plasticity or damage, large transformations, etc. Although our methodology can be applied to any kind of non-linearity, this work is more specifically focused on time-dependent non-linearities under moderate displacements and quasi-static assumption. Instability phenomena will be considered in upcoming papers.

Moreover, in design offices these non-linear problems have to be usually computed several times according to some set of parameters $\boldsymbol{\mu} \in \mathcal{D}$ within a product optimization process. This means that each of the quantities of interest $\bullet^{(\boldsymbol{\mu})}$ depends on $\boldsymbol{\mu}$, a collection of p parameters $\boldsymbol{\mu} = \{\mu_i\}_{i \in [1,p]}$. These parameters can be of different nature: loading, geometry variations, material data, etc. But since these parameters are handled in the following as extra-parameters – by carrying out one

calculation at each point of the parametric space – one skips specifying explicitly $\boldsymbol{\mu}$ in what follows in order to alleviate the notations whilst bearing in mind the parametric dimension of the problem.

The finite element method is followed from the weak form of the equations bringing a spatial discretization of the geometry Ω and thus providing an approximation of the different fields – displacements, forces, etc. – on the finite element space $\mathcal{V} \equiv \mathbb{R}^n$ with n the number of degrees of freedom. For the sake of simplicity, we do not use any specific time discretization: in particular, all quantities are defined over the entire time interval I . Thus, the finite element problem can be expressed in term of generalized quantities: the generalized displacements $\mathbf{u} \in \mathcal{I}_{\mathcal{V}}$ and the associated generalized forces $\mathbf{f}_{\text{int}} \in \mathcal{I}_{\mathcal{V}}$ respectively with $\mathcal{I}_{\mathcal{V}} \equiv L^2(I; \mathcal{V})$. Moreover, for each time $t \in I$, the work $w \in \mathcal{I}_{\mathbb{R}} \equiv L^2(I)$ can be assessed by the canonical scalar product on \mathcal{V} :

$$w(t) = \sum_{j=1}^n u_j(t) \mathbf{f}_{\text{int},j}(t) = \mathbf{u}(t) \cdot \mathbf{f}_{\text{int}}(t) \quad \forall t \in I \quad (2)$$

It follows that the reference problem to solve can be written as:

Find $\mathbf{s} = (\mathbf{u}, \mathbf{f}_{\text{int}}) \in \mathcal{I}_{\mathcal{V}} \times \mathcal{I}_{\mathcal{V}}$ such that:

- the generalized displacements \mathbf{u} comply with the imposed displacements \mathbf{u}_d on the edge of the domain corresponding to *Dirichlet's* conditions – kinematic constraints:

$$\mathcal{C}_u(\mathbf{u}(t)) = \mathbf{u}_d(t) \quad \forall t \in I \quad (3)$$

- the external forces \mathbf{f}_{ext} balance the internal ones \mathbf{f}_{int} – equilibrium equation:

$$\mathbf{f}_{\text{int}}(t, \mathbf{u}(t)) - \mathbf{f}_{\text{ext}}(t) = \mathbf{0} \quad \forall t \in I \quad (4)$$

- the generalized forces \mathbf{f}_{int} result from the non-linear local behavior described by a generic operator \mathcal{A}_u in a functional formalism [28] – constitutive relations:

$$\mathbf{f}_{\text{int}}(t, \mathbf{u}(t)) = \mathcal{A}_u(t, \mathbf{u}(\tau \leq t)) \quad \forall t \in I \quad (5)$$

where in addition zero initial conditions are assumed $\mathbf{u}|_{t=0} = \mathbf{0}$. The operator \mathcal{C}_u is taken linear as usual while \mathbf{u}_d and \mathbf{f}_{ext} are given over the time interval I . It should be noted in particular that the operator \mathcal{A}_u of the generalized constitutive relations refers not only to time but also to the past history of loading insofar as some irreversible processes such as plasticity can be involved. Another way to describe such physics could be with the use of the internal variables [29] which is generally the case for defining and then solving these local equations. **To be more precise, with the internal variables formalism, the writing of the behavioral law at time t depends only on the current state of the material at that time t and a set of internal state variables $\boldsymbol{\chi}$ that keeps track of the past history. Noting \mathcal{A}_{iv} as the operator of the generalized constitutive relations under such internal variables formalism, equation (5) would then be expressed as:**

$$\mathbf{f}_{\text{int}}(t) = \mathcal{A}_{iv}(t, \mathbf{u}(t), \boldsymbol{\chi}(t)) \quad \forall t \in I \quad (6)$$

But we do not intent to interact directly with these possible internal variables which will therefore remain hidden at the overall functional level of generalized quantities. **Thus, only the knowledge of**

\mathbf{u} and its past history is necessary to achieve the integration of the constitutive relations. It should be noted that we have chosen a functional approach in order to remain as close as possible to the industrial software and therefore the least intrusive possible. However, this operator \mathcal{A}_u does not appear naturally in general-purpose industrial finite element software, it must be built from the treatment of local elements carried out in the industrial software considered. This operator implies being able to process the whole time interval each time \mathcal{A}_u is called upon, which is not usually the case with the incremental architectures of finite element industrial software. We will see later how this *black-box* can be managed. Specifically, this will be the only weakly-invasive part of the proposed approach.

Remark for mixed formulation of contact

When contacts are involved, additional constraints occur at the contact surfaces. These constraints induce a new pair of variables $(\mathbf{v}, \mathbf{g}) \in \mathcal{I}_{\mathcal{V}_c} \times \mathcal{I}_{\mathcal{V}_c}$ linking the nodes of the surfaces in contact where $\mathcal{V}_c \equiv \mathbb{R}^{n_c}$, $\mathcal{I}_{\mathcal{V}_c} \equiv L^2(I; \mathcal{V}_c)$ and n_c denotes the number of degrees of freedom associated with contacts. More precisely, \mathbf{v} corresponds to the trace of the generalized displacements over the contacting area and \mathbf{g} refers to the dual quantity, namely additional Lagrange multipliers representing contact forces. Then, the reference problem becomes as follows:

Find $\mathbf{s} = (\mathbf{u}, \mathbf{v}, \mathbf{f}_{\text{int}}, \mathbf{g}) \in [\mathcal{I}_{\mathcal{V}} \times \mathcal{I}_{\mathcal{V}_c}]^2$ such that:

- the kinematic constraints (3) are completed by the ones of the nodes in contact:

$$\forall t \in I, \quad \begin{cases} \mathcal{C}_u(\mathbf{u}(t)) = \mathbf{u}_d(t) \\ \mathcal{C}_c(\mathbf{u}(t)) = \mathbf{v}(t) \end{cases} \quad (7)$$

- the constitutives relations (5) defining the generalized forces \mathbf{f}_{int} also embed the generalized contact forces \mathbf{f}_c :

$$\forall t \in I, \quad \mathbf{f}_{\text{int}}(t, \mathbf{u}(t)) = \mathcal{A}_u(t, \mathbf{u}(\tau \leq t)) + \underbrace{\mathcal{A}_c(t, \mathbf{v}(t), \mathbf{g}(t))}_{\mathbf{f}_c} \quad (8)$$

- the equilibrium equation (4) remains unchanged

and where the operator \mathcal{C}_c is linear as well. Once again, the operators \mathcal{A}_u and \mathcal{A}_c constitute the only weakly-invasive part of what we propose. From an incremental architecture, a new path must be pursued to be able to address the entire time interval several time during the resolution.

The Newton-Raphson method is generally used in computational mechanics to solve such non-linear equations in general-purpose industrial finite element software. In the next section, an other performant method based on a non-incremental approach is presented named LATIN-PGD method [7, 24]. Indeed, in order to take advantage of the strengths of the reduced-order models by PGD, information related to the whole space and the time have to be provided at every stage, an incremental approach is then no longer possible.

3. The weakly-invasive LATIN-PGD

In this section, a way of solving the previous problem is outlined based on a new formulation of the LATIN-PGD method [23] that can be smoothly integrated into any finite element industrial software.

The LATIN-PGD method [7, 24] has already been the subject of several research works [30, 31, 32, 33, 34] – the appendix in [25] may be referred for an exhaustive overview attempt. However, this is the first time that the LATIN-PGD method has been integrated at such a high level into an industrial process. Our approach developed hereafter does not target any particular problem – representing a particular physics like visco-plasticity, contacts, or composite materials for instance – but is aimed at the resolution of any class of non-linear thermo-mechanical problems in general-purpose industrial finite element software.

In the following, the LATIN-PGD method is hereby described through a functional perspective. The cornerstone of the LATIN-PGD strategy is still based on three main principles [24].

3.1. Principles **P1** and **P2**

The first principle **P1** consists in *separation of the difficulties*. Therefore, the set of equations defined over the time interval I are split into two groups: (\mathbf{A}_d) gathers all the global but linear equations of the problem – kinematic constraints (3) and equilibrium equation (4) – while (Γ) includes all the non-linear but local equations – constitutive relations (5) including behavior laws, contacts, etc. Schematically, a geometric representation of the LATIN-PGD algorithm can be provided as shown in Figure 1 where the two sub-spaces (\mathbf{A}_d) and (Γ) are depicted. In this representation, sub-space (\mathbf{A}_d) , which groups the linear equations, is denoted by a straight line while sub-space (Γ) holding the non-linear equations is displayed as a curve. One can note in particular that these sub-spaces embed time intrinsically which means that for instance an element \mathbf{s} belonging to sub-space (\mathbf{A}_d) has to satisfy all the equations gathered in this sub-space over the whole time interval I at each iteration. The intersection \mathbf{s}_χ is the *exact* solution of the reference problem. Of course, we assume that such a solution exists and is unique.

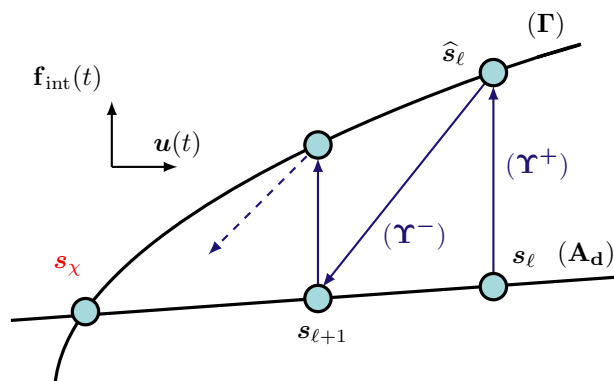


Figure 1: Convergent iterative scheme for LATIN-PGD algorithm where $\mathbf{s}_\ell = (\mathbf{u}_\ell, \mathbf{f}_{\text{int}, \ell})$

The second principle **P2** defines *two search directions*. By defining two additional sub-spaces (Υ^+) and (Υ^-), one can construct alternatively a solution of the first group (\mathbf{A}_d) and then a solution of the second group (Γ) at each iteration (see Figure 1). These search directions are the main parameters of the method. Their choice straightforwardly affects the convergence speed of the algorithm but also the ease of implementation within pre-established industrial software as we discuss later on.

The initialization of the algorithm can be performed with any couple \mathbf{s}_0 respecting static and kinematic admissibility. When any prior information is available, an elastic solution is usually adopted since it is quite easily computed. Then, the algorithm leads us through a succession of *local* and *global* stages by means of two search directions to the terms of a convergent series $(\mathbf{s}_k(t))_{k \in \mathbb{N}}$ of limit $\mathbf{s}_\chi(t)$. At a given state of the algorithm where \mathbf{s}_ℓ is considered to be known, one has to solve:

- **Local stage:** given $\mathbf{s}_\ell(t)$, a new solution $\widehat{\mathbf{u}}_\ell(t)$ is sought at the intersection $(\Gamma) \cap (\Upsilon^+)$ of the sub-spaces by solving the non-linear equations. In particular, a *vertical* search direction is used to close the system of equations:

$$\widehat{\mathbf{u}}_\ell(t) = \mathbf{u}_\ell(t) \quad \forall t \in I \quad (9)$$

One can note that this specific choice of the sub-space (Υ^+) characterizing the first search direction leads to naturally decoupling primal and dual quantities in the *local* stage: knowledge of the primal variable alone is sufficient to solve all the equations at this stage. To get the corresponding forces $\widehat{\mathbf{f}}_{\text{int}, \ell}$, the constitutive relations should be integrated thanks to the general-purpose industrial finite element software. This task is not standard but remains weakly-invasive. This is the cornerstone for the proposed version of the LATIN-PGD where the operator \mathcal{A}_u should be constructed, which involves all the content (constitutive relations, elements, algorithms,...) of the finite element software we are considering;

- **Global stage:** given $\widehat{\mathbf{s}}_\ell(t)$, a new solution $\mathbf{s}_{\ell+1}(t)$ is sought at the intersection $(\mathbf{A}_d) \cap (\Upsilon^-)$ of the sub-spaces by solving the equilibrium equations. Hence, given a descent search direction defined by an operator \mathbf{H} , the equations to be solved are:

$$\forall t \in I, \quad \begin{cases} \mathbf{H} \Delta \mathbf{u}_{\ell+1}(t) = \mathbf{f}_{\text{int}, \ell+1}(t) - \widehat{\mathbf{f}}_{\text{int}, \ell}(t) \\ \mathbf{f}_{\text{int}, \ell+1}(t) - \mathbf{f}_{\text{ext}}(t) = \mathbf{0} \\ \mathbf{u}_{\ell+1}(t) = \widehat{\mathbf{u}}_\ell(t) + \Delta \mathbf{u}_{\ell+1}(t) \end{cases} \quad (10)$$

The choice of the search direction operator \mathbf{H} defining the additional sub-space (Υ^-) is the major parameter of the method. Once again, several choices are possible to define this search direction guided on the one hand by the ease of implementation and on the other hand by the convergence rate. The choice of \mathbf{H} will be detailed later on.

The fulfillment of this iterative algorithm continues as long as the new couple $\mathbf{s}_{\ell+1}$ remains too far from the true solution \mathbf{s}_χ with a certain threshold. Of course, since the true solution can not be known in advance, an error indicator is used to state how confident one can have in the exhibited solution. Several criteria could be considered for this purpose but here a criterion based on the energy norm averaged over the time interval is employed enabling to measure in a certain sense the distance between sub-spaces (\mathbf{A}_d) and (Γ).

3.2. Principle **P3** or PGD approximation

One proposes to use the usual PGD computation technique with the LATIN method which is very robust. For the sake of conciseness, only the main lines are remembered hereafter – see [35] for more details. Typically, at a given iteration, knowing the previous approximation $\{(\mathbf{\Lambda}_i, \lambda_i^\ell)\}_{i \in \llbracket 1, m_\ell \rrbracket} \in \{\mathcal{V} \times \mathcal{I}_{\mathbb{R}}\}^{m_\ell}$ with:

$$\mathbf{u}_\ell(t) = \sum_{i=1}^{m_\ell} \lambda_i^\ell(t) \mathbf{\Lambda}_i \quad (11)$$

one seeks new temporal functions $\{\lambda_i^{up}\}_{i \in \llbracket 1, m_\ell \rrbracket}$ – spatial modes being fixed – giving the best approximation of $\Delta \mathbf{u}_{\ell+1}$ such as:

$$\Delta \mathbf{u}_{\ell+1}(t) = \sum_{i=1}^{m_\ell} \underbrace{(\lambda_i^{up}(t) - \lambda_i^\ell(t))}_{\beta_i(t)} \mathbf{\Lambda}_i \quad \forall t \in I \quad (12)$$

Without going into details, this preliminary stage aims at solving the following minimization problem:

$$\{\beta_i\}_{i \in \llbracket 1, m_\ell \rrbracket} = \arg \min_{\substack{\alpha_i \in \mathcal{I}_{\mathbb{R}} \\ i \in \llbracket 1, m_\ell \rrbracket}} J \left(\sum_{i=1}^{m_\ell} \alpha_i(t) \mathbf{\Lambda}_i \right) \quad (13)$$

where the functional J is given by:

$$J(\mathbf{v}) = \frac{1}{2} \int_I \mathbf{v}^T \left[\mathbf{H} \mathbf{v} - 2(\mathbf{f}_{\text{ext}} - \widehat{\mathbf{f}}_{\text{int}, \ell}) \right] dt \quad (14)$$

Then, if this preliminary stage is proved to be ineffective, a new pair $(\mathbf{\Lambda}, \lambda)$ is sought which leads to solve the following minimization problem:

$$(\mathbf{\Lambda}, \lambda) = \arg \min_{\substack{\alpha \in \mathcal{I}_{\mathbb{R}} \\ \gamma \in \mathcal{V}}} J \left(\alpha \gamma + \sum_{i=1}^{m_\ell} \beta_i \mathbf{\Lambda}_i \right) \quad (15)$$

what finally can be expressed as a classical fixed point problem:

$$\begin{cases} \mathbf{\Lambda} = \xi_{\mathbf{\Lambda}}(\lambda) \\ \lambda = \xi_{\lambda}(\mathbf{\Lambda}) \end{cases} \quad (16)$$

To end, the new approximation $\mathbf{u}_{\ell+1}$ is expressed by:

$$\mathbf{u}_{\ell+1}(t) = \mathbf{u}_\ell(t) + \sum_{i=1}^{m_\ell} \beta_i(t) \mathbf{\Lambda}_i + \lambda(t) \mathbf{\Lambda} \quad \forall t \in I \quad (17)$$

$$= \sum_{i=1}^{m_\ell} \underbrace{(\lambda_i^\ell(t) + \beta_i(t))}_{\lambda_i^{\ell+1}(t)} \mathbf{\Lambda}_i + \lambda(t) \mathbf{\Lambda} \quad (18)$$

3.3. Optimization of the search direction

The search direction operator \mathbf{H} is the main parameter of the method. As a parameter, several choices are then possible, which mainly affect both performance and convergence rate of the method. Two extreme cases can be distinguished. One is to consider the tangent operator $\mathbf{H}(t)$ which should lead to an optimal convergence with respect to the number of iterations. The computation of this search direction operator is relatively easy. This is performed during the local stage at the same time as \mathcal{A}_u directly through the industrial finite element software. Unfortunately, as this operator depends on time t , hence the computation of each iteration could be very high. The other extreme case is the use of the elasticity rigidity \mathbf{K} which has two very interesting properties : this operator is time-independent and does not vary along the iterations. This results in a low cost iteration computation but the total number of iterations to be achieved could be larger. For elasto-viscoplastic problems, with possibly contacts, in the case of moderate displacements, all the numerical tests that have been done give to the elasticity rigidity as the search direction a large superiority. **In addition, as the elasticity rigidity \mathbf{K} is less optimal, a higher number of generated modes is required to reach a similar error level. However, as this operator remains constant over time and iterations, the generation of these few additional PGD modes does not blow up the general computation time.** For example, Table 1 gives the total number of iterations and the total computation time to achieve a same level of accuracy in the case of the turbine blade studied in Section-4.

Table 1: Comparison of computation times, **number of iterations** and **number of PGD modes** according to different search directions

| | Computation time(s) | | | Number of | |
|-----------------|---------------------|---------------------|--------------|------------|--------------|
| | <i>Local stage</i> | <i>Global stage</i> | <i>Total</i> | iterations | modes |
| $\mathbf{H}(t)$ | 2361 | 4499 | 6860 | 18 | 11 |
| \mathbf{K} | 3488 | 811 | 4299 | 31 | 15 |

Let us note that an intermediate option between the tangent operator $\mathbf{H}(t)$ and the elasticity rigidity \mathbf{K} has been introduced for large displacements. It will be detailed in a companion paper.

3.4. Weakly-invasive implementation

A weakly-invasive approach is undertaken in incorporating the LATIN-PGD method into the general-purpose industrial finite element Simcenter SamcefTM software without any disruption to the hosting software.

Constraints

One specificity of our approach is that Siemens Digital Industries Software, as a software editor, provides advanced tools to a broad scientific community, including other industrials, and not only to a specific partner as an aircraft engine manufacturer for example. Therefore, the issue of robustness is fundamental. Not being a simple user of industrial finite element software, one has access to the inside of such software which raises new opportunities but also new constraints. On the one hand,

classical features of any general-purpose industrial finite element software become accessible by calling specifically any different routines of the code. But on the other hand, one wishes to get a unified version with all inclusive tools directly integrated into the industrial software for obvious reasons of ease of the customers' use. Therefore, one wishes not only to be able to build reduced-order models but also manipulate or enhance them directly within the industrial software (or its graphic interface) and it seems then unthinkable to use an external code to perform any additional operations. Additionally, it seems also inconceivable to turn the entire architecture upside down: any new technics must be integrated without any hindrance with regards to the existing tools, hence the notion of *weakly-invasive* implementation. Another wish is to be able to use the same certification steps during the construction of each new version. These certification steps ensure that no adverse effects are produced by the simple addition of the interface connection and that the new LATIN-PGD method is reliable and robust for all the cases considered.

Our approach

Only a few parsimonious modifications have been necessary to introduce the LATIN-PGD method in pre-existing Simcenter SamcefTM software, notably through the definition of a dedicated interface based on a Fortran module connecting all the new LATIN-PGD capabilities to the hosting software. Figure 2 gives the general architecture of the pursued methodology: first, we come back to the global stage incorporating the implementation of the PGD algorithms. This whole part, including all the ingredients for the PGD, which is considered as intrusive and usually handled externally, is being kept independent. This means that there is no chance of breaking the existing software. Hence, all this part could have been outsourced in MatlabTM or PythonTM, but for performance concerns it seemed more appropriate to retain everything compiled in Fortran. Furthermore, external MUMPS sparse solver, SVD routines or any classical linear algebra library can be reemployed, so that only what does not already exist in the original software has to be added. Then, all the information flows through the interface to make the link with the industrial finite element software which performs the local stage. All types of elements, all kinds of loading can be used and no intervention is required at this stage. The only thing we have to do is to provide primal quantities and get in return the dual ones – *i.e.* build the operator \mathcal{A}_u . A tricky point that has required particular attention concerns data storage of such quantities 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 parsimonious dynamic allocations within the interface ensure that the methodology can be carried out smoothly, taking advantage that the solution can be written in a separable variables format. The result is a controlled memory increase.

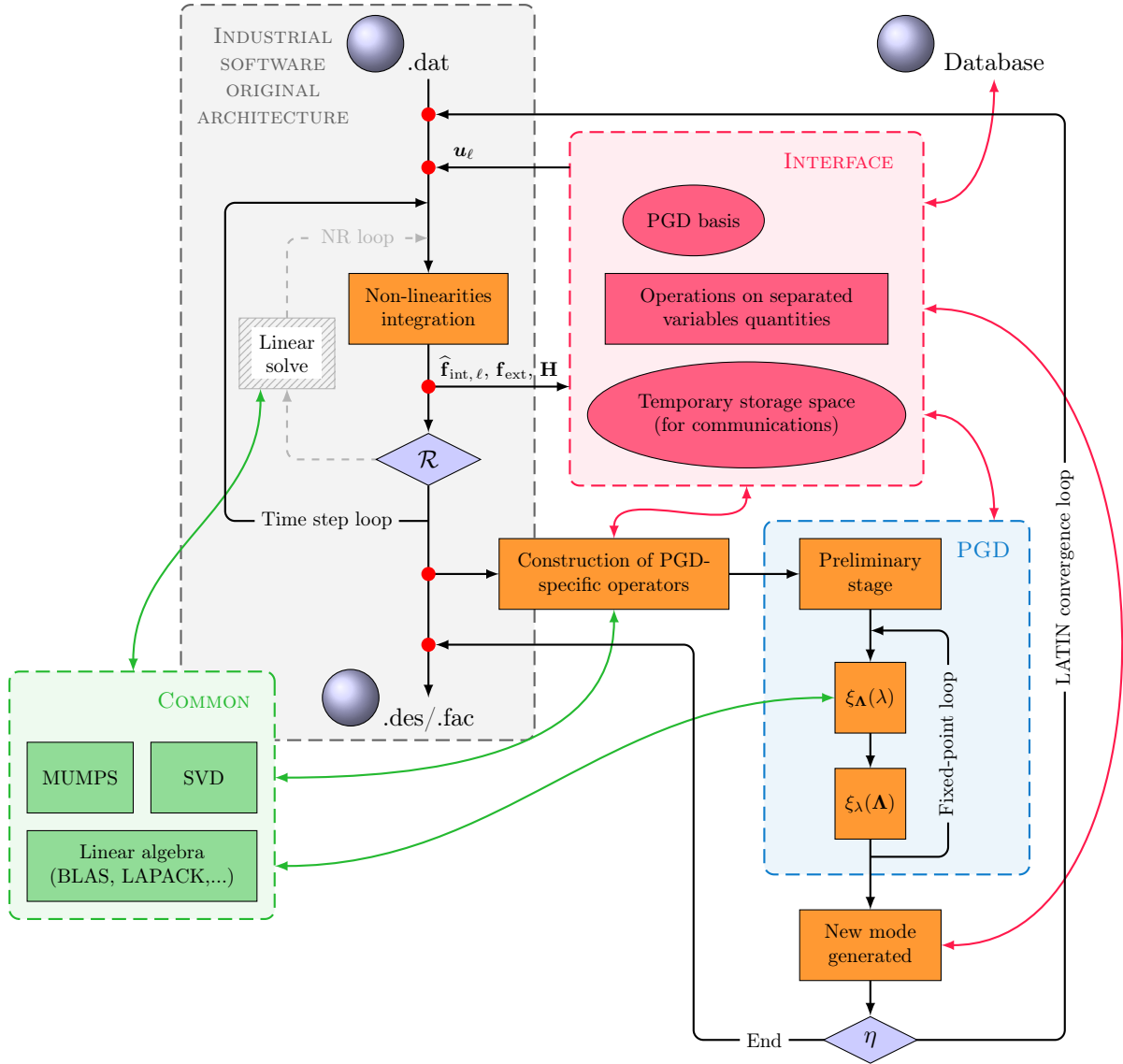


Figure 2: General architecture of the weakly-invasive methodology setting up LATIN-PGD method in general-purpose industrial finite element software

3.5. LATIN-PGD with parameters

Returning to the handling of parametrized problems, one exploits the non-invasive technics developed especially in [36, 33]. Here, we have implemented the simplest version where the response surface is first discretized and then defined thanks to an interpolation process. The snapshots associated with the extra-parameters are computed from one to the next resulting from the applied weakly-invasive LATIN-PGD methodology. This is barely inexpensive insofar as the LATIN-PGD algorithm embeds the special property to be initialized from any previously computed solution. To be as clear as possible, let us consider the computation of a new point $(\boldsymbol{\mu} + \Delta\boldsymbol{\mu}) \in \mathcal{D}$ in the

parametric space. Knowing $\mathbf{s}^{(\mu)}$ over I and the corresponding spatial reduced-order basis:

$$\{\mathbf{\Lambda}_i\}_{i \in \llbracket 1, m^{(\mu)} \rrbracket} \in \mathcal{V}^{m^{(\mu)}} \quad (19)$$

the first iteration of the LATIN-PGD is as follows:

- **Local stage:** given $\mathbf{s}_0^{(\mu+\Delta\mu)}(t)$, one seeks $\widehat{\mathbf{s}}_0^{(\mu+\Delta\mu)}(t)$ with:

$$\widehat{\mathbf{u}}_0^{(\mu+\Delta\mu)}(t) = \mathbf{u}^{(\mu)}(t) \quad \forall t \in I \quad (20)$$

- **Global stage:** given $\widehat{\mathbf{s}}_0^{(\mu+\Delta\mu)}(t)$, one then seeks $\mathbf{s}_1^{(\mu+\Delta\mu)}(t)$ where from the preliminary stage, one gets:

$$\mathbf{u}_1^{(\mu+\Delta\mu)}(t) = \sum_{i=1}^{m^{(\mu)}} \lambda_i^{(\mu+\Delta\mu)}(t) \mathbf{\Lambda}_i + \mathbf{u}_0^{(\mu)}(t) \quad (21)$$

and the algorithm then continues as stated previously with $\ell \geq 1$. For many parameter values, it is not necessary to add a new mode to the working reduced-order basis and the process can be stopped after a few iterations, controlled by some error indicators. In what follows, some results are given to illustrate the advantages of this approach.

Remark around space-time PGD

We have chosen deliberately to use only a space-time PGD for the weakly-invasive LATIN-PGD in order to remain as generic as possible and the least intrusive possible with regard to the industrial software. Indeed, space and time are universal variables in any simulation and classically manage in all general-purpose finite element software, whereas the other (extra-)parameters can vary according to the test case. As these $\boldsymbol{\mu} \in \mathcal{D}$ parameters are currently managed externally in existing industrial software, we must keep this constraint. However, this does not exclude *a posteriori* data compression phases to further limit the memory footprint of these parametric solutions or in order to build multi-parameters reduced-order models.

4. Illustrations

In this section, the assets of the weakly-invasive LATIN-PGD method are highlighted. After having briefly described the test-case, some results are given based on a parametric study involving two extra-parameters. In particular, alongside providing full construction of a reduced-order model in an end-to-end industrial environment, interesting time saving can be appreciated as well.

All computations have been performed on a cluster of linux machines with a MPI multi-nodes architectures involving Intel[®] computing nodes on Intel[®] Xeon[®] Gold 6126 CPU @ 2.60 GHz processors up to 24 cores and 192 GB of RAM per nodes. Among these resources, 12 cores were used in this work, possibly distributed over 2 compute nodes when the memory available on a single node was not sufficient. Moreover, Simcenter Samcef[™] industrial finite element software has been used both for classical Newton-Raphson and weakly-invasive LATIN-PGD methods. Classical commercial version – using Newton-Raphson algorithm – is used as a reference for conducting comparisons. Pre and post-processing phases are done as well though the corporate Simcenter[™] 3D tool developed by Siemens company.

4.1. Test-case description

This test-case deals with a high-pressure blade of an aircraft engine inspired by the M88 engine powering the Rafale fighter aircraft. For confidentiality reasons, a standard geometry is considered. The mesh accounting for 5 millions of degrees of freedom is generated through the SimcenterTM 3D interface, as well as the setting of boundary conditions. Both centrifugal and lateral forces are undertaken, the bottom of the fir stand being clamped. The material follows a 9-parameters Chaboche elasto-visco-plastic law [29] with a high temperature dependence. In what follows, a two parameters study is conducted $\mathcal{D} = \mathcal{D}_\alpha \times \mathcal{D}_T$ where the parameters are the temperature $T \in \mathcal{D}_T = [850, 1010]$ in Celsius and an angle $\alpha \in \mathcal{D}_\alpha = [45, 170]$ in degrees giving the direction of lateral forces at the top, as shown in Figure 3 – these kind of forces can occur when a fairing deformity is involved. It can be noted in particular that these parameters play a role both on (\mathbf{A}_d) and $(\mathbf{\Gamma})$ sub-spaces previously introduced: each new value of the temperature T influences the 9-parameters of the material law so mainly affects $(\mathbf{\Gamma})$ while each new value of the angle α impacts the external forces considered in (\mathbf{A}_d) .

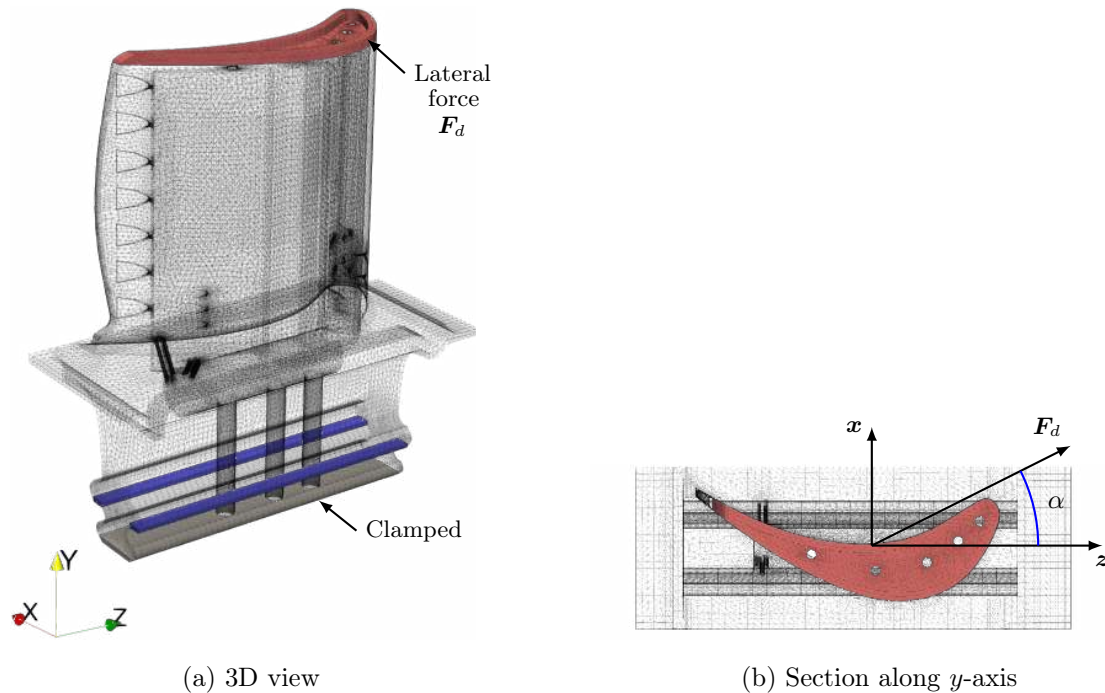


Figure 3: Loading case of the turbine blade

4.2. Results

Figure 4 shows the results obtained for a given point ($\alpha = 155^\circ$ and $T = 890^\circ\text{C}$) in the parametric space. Sub-figures 4a and 4b are respectively devoted to the displacement and the Von Mises stress fields obtained with the weakly-invasive LATIN-PGD method included in Simcenter SamcefTM software. One can see in particular that maximal stresses are mainly located at connections between the upper and the lower part of the blade, as well as around ventilation holes. Sub-figures 4c and 4d plot the error maps corresponding to the difference between the reference – using Simcenter

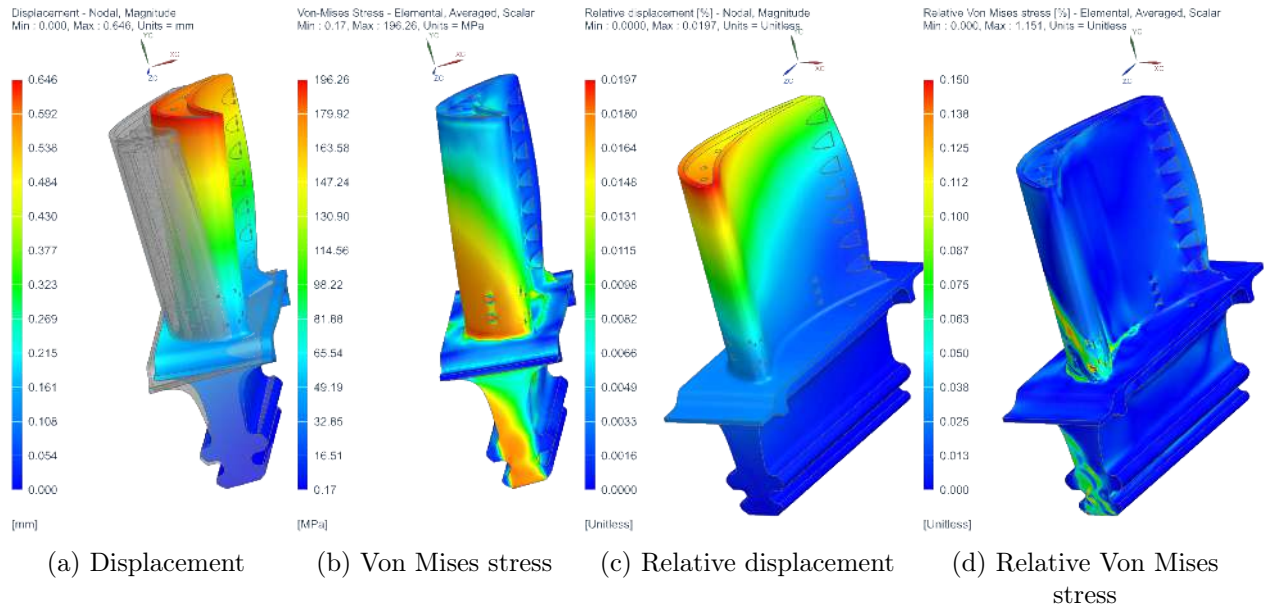


Figure 4: Displacement (a) and Von Mises stress (b) results with LATIN-PGD method included in Simcenter SamcefTM software and resulting error maps with regard to the Newton-Raphson reference : relative displacement (c) and Von Mises stress (d) respectively (in percent [%])

SamcefTM software in its standard version – and our fully integrated LATIN-PGD methodology. In order to better appreciate the possible errors made, error maps are given in relative quantities. In particular, the very good concordance of the high-fidelity solutions can be noted with a relative error of less than one percent with only 8 modes in the reduced-order PGD basis.

Figure 5 gives the virtual chart of the maximum Von Mises stress on the parametric space \mathcal{D} . This chart highlights a strong non-linear variation according to the temperature and a lighter variation but existing one depending on the loading angle α (cf. sub-figure 5a). In order to achieve this, more than 250 high-fidelity computations have been carried out by scanning the parametric space sequentially in a uniform manner. We use a regular discretization of our 2D parametric space with respectively 15 and 17 points per dimension, for α and T . We have decided to put more points on the temperature because of the fact that the solution is clearly more sensitive to the evolution of the temperature. But other choices of sampling of the parametric space are of course possible. Moreover, on sub-figure 5b is depicted the error map in the estimation of the maximum Von Mises stress quantity, elaborated by rigorously performing the same calculation points in the parametric space but using our reference algorithm following the Newton-Raphson method. Thus, we have on this virtual chart an excellent approximation of the quantity of interest since the error appears to be of the order of one percent. One of the major advantages of our approach is exposed here: in addition to generating some virtual charts in an end-to-end industrial process, we can do this faster deploying the weakly-invasive LATIN-PGD methodology (see Figure 6). It can be noticed that here we have completed 255 points in the parametric space. The speed-up would be even more impressive either with more points – one thousand – or with more parameters. Furthermore, there would potentially be an optimal path that would optimize the number of LATIN iterations. Here we have followed a regular path from close to close. We have not investigated this issue insofar

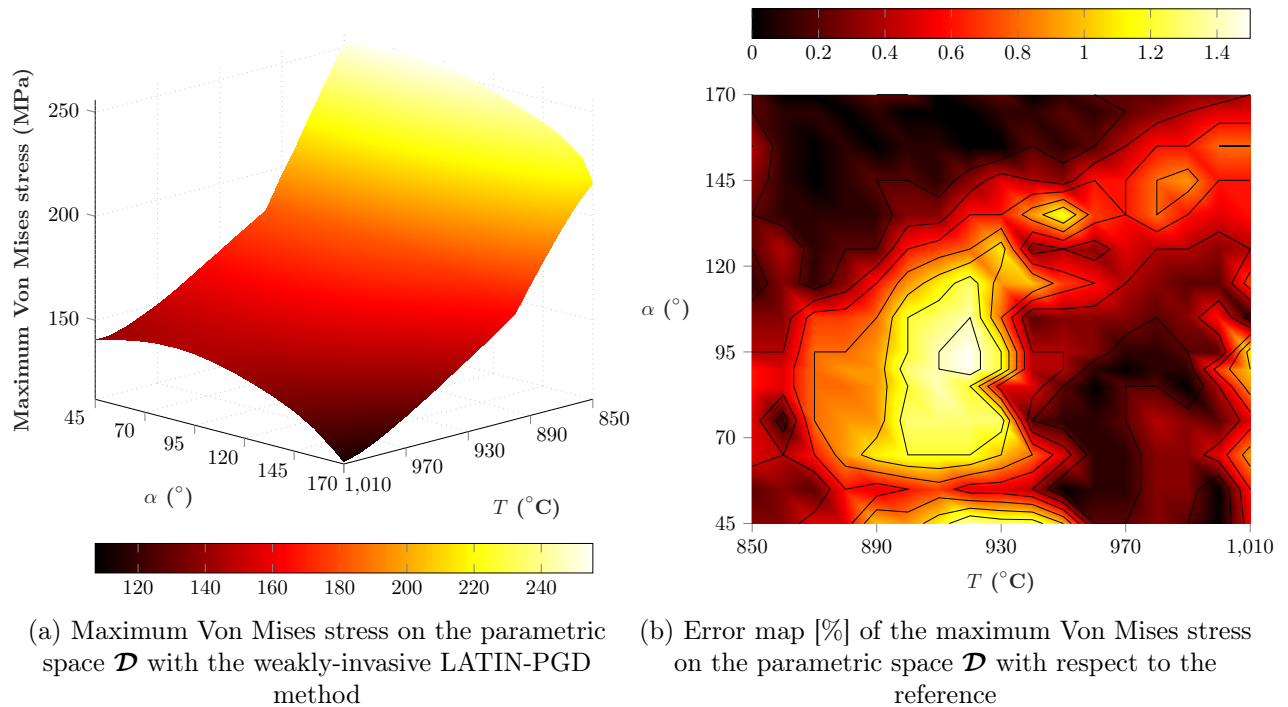


Figure 5: Virtual chart of the maximum Von Mises stress on the parametric space \mathcal{D} resulting from 255 parametric points uniformly pulled across the parametric space

as the focus was main oriented on the analysis of the robustness. But it would undoubtedly be necessary to explore this path in more detail when the number of parameters becomes large. A study on the determination of the optimal path in the design space can be found in [36] for instance.

This good performance is due to both encapsulated weakly-invasive space-time PGD and LATIN properties allowing notably to restart the computation from any close previous converged point in the parametric space [37]. The novelty of our work lies in the end-to-end integration of the LATIN-PGD method into an industrial software, providing a more robust and general tool to deal with any type of non-linear multi-parametric problems. One another future development could be to improve the parameter treatment in relation with design optimization following [38, 39, 40].

Once such virtual charts are available, it is then straightforward to carry out stochastic studies in real time. For instance, if one chooses a Gaussian probability distribution for the temperature $T \sim \mathcal{N}(T_m, \sigma^2)$ (of mean T_m and standard deviation σ) and a uniform density probability law for the second parameter α , more than ten millions Monte-Carlo samples can be made in less than a second. Figure 7a shows the resulting density probability function for the maximum Von Mises stress with $T \sim \mathcal{N}(990, 25)$ and α evenly distributed over the interval [75, 140] in degrees. Uncertainties on the parameters can therefore be easily grasped for structural design or even predictive maintenance – if one imagines that these data come from sensors reflecting the state of the structure in operation. Figure 7b gives the probability bounds for the resulting maximum Von Mises stress where there is a 95% probability of having $136.61 \leq \sigma_M^{95\%} \leq 149.19$ MPa.

An other very interesting property of the LATIN-PGD method consists in describing the main features of the solution from the first iterations. Hence, after each LATIN iteration, one can decide

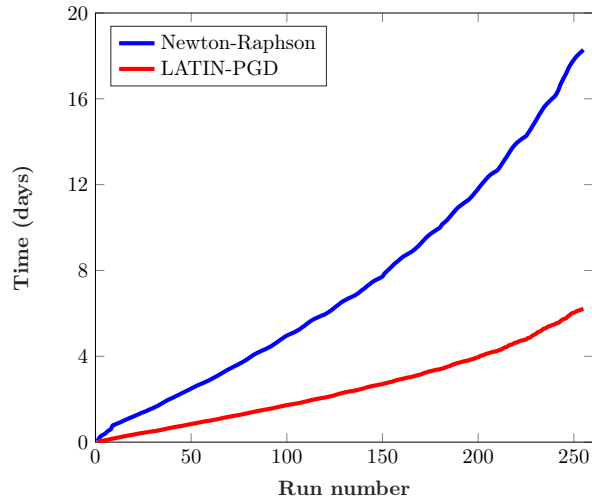


Figure 6: Time needed to perform all the *offline* phase in the parametric space \mathcal{D} according either only external use of industrial finite element software or the proposed weakly-invasive LATIN-PGD methodology

if the accuracy is already sufficient for our needs or if we need to proceed further until achievement of the high-fidelity solution. Figure 8 depicts the evolution of the Von Mises stress error for a given value of the parameters and for the most loading time step according to ζ_ℓ , the inverse speed-up ratio between a classical resolution by Newton-Raphson scheme – needing Δt_r time – and the weakly-invasive LATIN-PGD method along the different iterations – with Δt_ℓ the cumulative time until ℓ iteration:

$$\zeta_\ell = \frac{\Delta t_\ell}{\Delta t_r} \quad (22)$$

It means that the smaller ζ_ℓ , the more efficient the LATIN-PGD method is. Black dashed curve symbolises the goal of having a global error under one percent with respect to the reference, red curve gives the mean error on Ω and the two blue curves shows respectively the bounds below which 95% or 99% of the elements are included. Finally, the small 3D pictures illustrate the spatial distribution of the error according to the right-hand scale. Thus, it can be seen that, on average, a good approximation of the solution is reached very quickly as the red curve crosses the black dotted one around $\zeta_\ell = 0.2$ on the x -axis. This means that on average over the entire space Ω , the error on the Von Mises stress falls below one percent for a corresponding computation time of $\Delta t_\ell = 0.2 \Delta t_r$ which results in a 5 times faster computation compared to standard commercial software. In other words, the first PGD modes capture the main trends of the solution and can be seen as *global* modes. Then, if one needs to have a finer approximation of the solution everywhere else – in the spatial zones still too badly represented –, additional modes can be added to decrease the error. These new modes have a more and more localized effect as the reduced-order basis is enriched, and enable the error to decrease for the few elements not yet satisfying (blue curves). For instance, it can be noted at $\zeta_\ell = 0.4$ that 95% of the elements have an error of less than one percent. Finally, having a multi-fidelity solver can prove to be a real strength for real-time monitoring during computation directly through the SimcenterTM 3D interface and thus not letting a run continue uselessly.

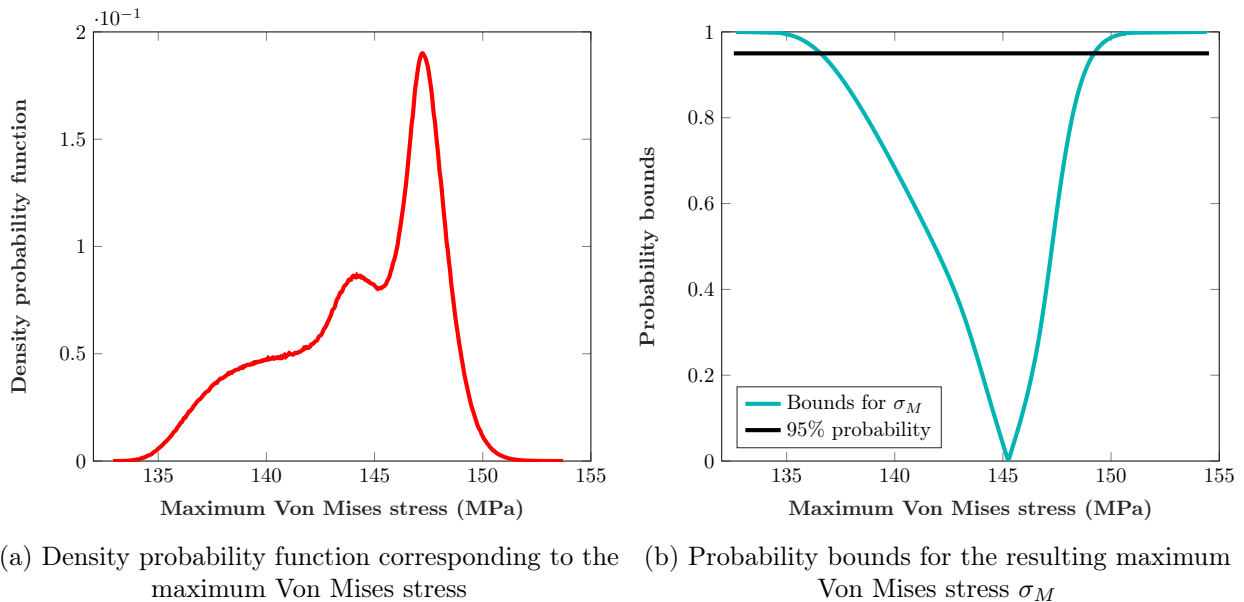


Figure 7: Stochastic study illustration with $T \sim \mathcal{N}(990, 25)$ and α evenly distributed over the interval $[75, 140]$

5. Conclusions and perspectives

In this work, the LATIN-PGD method has been introduced within a general-purpose industrial finite element software in a weakly-invasive manner. All this greatly enhances computation times and in particular allows for fast construction of reduced-order models. Although presented on material non-linearities, this very general approach can also be applied for any kind of other non-linearities natively included in general-purpose industrial finite element software. This is one main strength of our approach, thus avoiding to redevelop in-house external codes or whole parts of the software. Other non-linearities such as contacts or large transformations will be developed in upcoming papers. At the present time, what has been done fits into the general framework of moderate displacements, outside instability zones. Moreover, the high generality of the approach could also allow in the future, with only minor adaptations, to address other software leading the resolution of other physics such as thermal or magneto-static problems. **Some ongoing works are devoted to the solving of the thermal problem alone. A strong thermo-mechanical coupling is for the moment out of our scope but belongs to our futur target. In fact, the simulation of strongly coupled problems (non-linear thermo-poro-elasticity) as been already the topic of previous studies using the LATIN-PGD as solver [41].** From an optimization point of view, hyper-reduction technics [42, 43, 44] could be added to achieve even more interesting speed-up. These technics enable to free oneself from the dimensions of the initial problem in the conduct of the various algebraic operations, which would reduce the non-negligible cost of the local stage along iterations. One another future perspective could be to improve the parameter treatment in relation with design optimization following [38, 39, 40]. To conclude, this is the first time ever that the LATIN-PGD method has been natively integrated into a general-purpose industrial finite element software. The weakly-invasive LATIN-PGD method fully integrated into the industrial finite element Simcenter SamcefTM software enables not only fast building of reduced-order models but also an easy-to-use access to the stored information to achieve further computations smarter and faster.

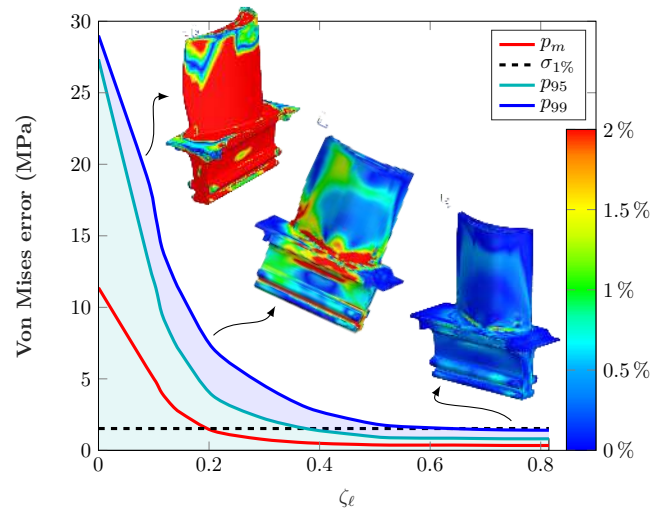


Figure 8: Bundles of curves showing the evolution of the Von Mises stress error according to the inverse speed-up with some 3D global representation at some LATIN-PGD iterations

Acknowledgements

Authors acknowledge the support of the french National Association for Research and Technology (ANRT) and Siemens Digital Industries Software company.

References

- [1] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. M. Silveira (Eds.), Model Order Reduction: Snapshot-Based Methods and Algorithms, volume 2, De Gruyter, Berlin, Boston, 2021.
- [2] A. Chatterjee, An introduction to the proper orthogonal decomposition, *Current Science* 78 (2000) 808–817.
- [3] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1901) 559–572. doi:10.1080/14786440109462720.
- [4] Y. Maday, E. M. Rønquist, A reduced-basis element method, *Comptes Rendus Mathématique* 335 (2002) 195–200. doi:10.1016/S1631-073X(02)02427-5.
- [5] G. Rozza, D. B. P. Huynh, A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Archives of Computational Methods in Engineering* 15 (2007) 229–275. doi:10.1007/BF03024948.
- [6] F. Chinesta, P. Ladevèze (Eds.), Separated Representations and PGD-based Model Reduction: Fundamentals and Applications, number 554 in *Courses and Lectures / International Centre for Mechanical Sciences*, Springer, Wien, 2014.
- [7] P. Ladevèze, 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* 300 (1985) 41–44.
- [8] L. Gendre, O. Allix, P. Gosselet, F. Comte, Non-intrusive and exact global/local techniques for structural problems with local plasticity, *Computational Mechanics* 44 (2009) 233–245. doi:10.1007/s00466-009-0372-9.
- [9] L. Giraldo, D. Liu, H. G. Matthies, A. Nouy, To Be or Not to be Intrusive? The Solution of Parametric and Stochastic Equations—Proper Generalized Decomposition, *SIAM Journal on Scientific Computing* 37 (2015) A347–A368. doi:10.1137/140969063.
- [10] C. Audouze, F. D. Vuyst, P. B. Nair, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, *Numerical Methods for Partial Differential Equations* 29 (2013) 1587–1628. doi:10.1002/num.21768.

- [11] F. Casenave, A. Ern, T. Lelièvre, A nonintrusive reduced basis method applied to aeroacoustic simulations, *Advances in Computational Mathematics* 41 (2014) 961–986. doi:10.1007/s10444-014-9365-0.
- [12] F. Casenave, A. Gariah, C. Rey, F. Feyel, A nonintrusive reduced order model for nonlinear transient thermal problems with nonparametrized variability, *Advanced Modeling and Simulation in Engineering Sciences* 7 (2020) 22. doi:10.1186/s40323-020-00156-3.
- [13] J. K. Hammond, R. Chakir, F. Bourquin, Y. Maday, PBDW: A non-intrusive Reduced Basis Data Assimilation method and its application to an urban dispersion modeling framework, *Applied Mathematical Modelling* 76 (2019) 1–25. doi:10.1016/j.apm.2019.05.012.
- [14] A. Vizzaccaro, A. Givois, P. Longobardi, Y. Shen, J.-F. Deü, L. Salles, C. Touzé, O. Thomas, Non-intrusive reduced order modelling for the dynamics of geometrically nonlinear flat structures using three-dimensional finite elements, *Computational Mechanics* (2020). doi:10.1007/s00466-020-01902-5.
- [15] D. Borzacchiello, J. V. Aguado, F. Chinesta, Non-intrusive Sparse Subspace Learning for Parametrized Problems, *Archives of Computational Methods in Engineering* 26 (2019) 303–326. doi:10.1007/s11831-017-9241-4.
- [16] R. Ibáñez, E. Abisset-Chavanne, A. Ammar, D. González, E. Cueto, A. Huerta, J. L. Duval, F. Chinesta, A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition, *Complexity* (2018). doi:10.1155/2018/5608286.
- [17] A. Courard, D. Néron, P. Ladevèze, L. Ballere, Integration of PGD-virtual charts into an engineering design process, *Computational Mechanics* 57 (2016) 637–651. doi:10.1007/s00466-015-1246-y.
- [18] X. Zou, M. Conti, P. Diez, F. Auricchio, A nonintrusive proper generalized decomposition scheme with application in biomechanics, *International Journal for Numerical Methods in Engineering* 113 (2018) 230–251. doi:10.1002/nme.5610.
- [19] C. Ghnatios, E. Cueto, A. Falco, J.-L. Duval, F. Chinesta, Spurious-free interpolations for non-intrusive PGD-based parametric solutions: Application to composites forming processes, *International Journal of Material Forming* 14 (2021) 83–95. doi:10.1007/s12289-020-01561-0.
- [20] A. Leon, S. Mueller, P. de Luca, R. Said, J.-L. Duval, F. Chinesta, Non-intrusive proper generalized decomposition involving space and parameters: Application to the mechanical modeling of 3D woven fabrics, *Advanced Modeling and Simulation in Engineering Sciences* 6 (2019) 1–20. doi:10.1186/s40323-019-0137-8.
- [21] F. Cavaliere, S. Zlotnik, R. Sevilla, X. Larrayoz, P. Diez, Nonintrusive reduced order model for parametric solutions of inertia relief problems, arXiv:2012.01961 [cs, math] (2020). arXiv:2012.01961.
- [22] V. Tsiolakis, M. Giacomini, R. Sevilla, C. Othmer, A. Huerta, Nonintrusive proper generalised decomposition for parametrised incompressible flow problems in OpenFOAM, *Computer Physics Communications* 249 (2020) 107013. doi:10.1016/j.cpc.2019.107013.
- [23] P. Ladevèze, On a new non-intrusive version of the LATIN method for quasi-static loads (in French), Rapport interne LMT, Ecole Normale Supérieure Paris-Saclay, 2017.
- [24] P. Ladevèze, *Nonlinear Computational Structural Mechanics: New Approaches and Non-Incremental Methods of Calculation.*, Springer, New York, 1999.
- [25] R. Scanff, S. Nachar, P. A. Boucard, D. Néron, A Study on the LATIN-PGD Method: Analysis of Some Variants in the Light of the Latest Developments, *Archives of Computational Methods in Engineering* 28 (2021) 3457–3473. doi:10.1007/s11831-020-09514-1.
- [26] P. Boisse, P. Bussy, P. Ladevèze, A new approach in non-linear mechanics: The large time increment method, *International Journal for Numerical Methods in Engineering* 29 (1990) 647–663. doi:10.1002/nme.1620290312.
- [27] P. Barabinot, R. Scanff, P. Ladevèze, D. Néron, B. Cauville, Industrial Digital Twins based on the non-linear LATIN-PGD, *Advanced Modeling and Simulation in Engineering Sciences* 8 (2021) 22. doi:10.1186/s40323-021-00207-3.
- [28] J. L. Chaboche, Constitutive equations for cyclic plasticity and cyclic viscoplasticity, *International Journal of Plasticity* 5 (1989) 247–302. doi:10.1016/0749-6419(89)90015-6.
- [29] J. Lemaitre, J.-L. Chaboche, A. Benallal, R. Desmorat, *Mécanique des matériaux solides - 3e édition*, Dunod, Paris, France, 2009.
- [30] M. Bhattacharyya, A. Fau, U. Nackenhorst, D. Néron, P. Ladevèze, A LATIN-based model reduction approach for the simulation of cycling damage, *Computational Mechanics* 62 (2018) 725–743. doi:10.1007/s00466-017-1523-z.
- [31] P.-A. Boucard, P. Ladevèze, M. Poss, P. Rougée, A nonincremental approach for large displacement problems, *Computers & Structures* 64 (1997) 499–508. doi:10.1016/S0045-7949(96)00165-4.
- [32] D. Dureisseix, P. Ladevèze, D. Néron, B. Schrefler, A Multi-Time-Scale Strategy for Multiphysics Problems: Application to Poroelasticity, *International Journal for Multiscale Computational Engineering* 1 (2003). doi:10.1615/IntJMCompEng.v1.i4.50.

- [33] N. Relun, D. Néron, P.-A. Boucard, A model reduction technique based on the PGD for elastic-viscoplastic computational analysis, *Computational Mechanics* 51 (2013) 83–92. doi:10.1007/s00466-012-0706-x.
- [34] B. Vandoren, K. De Proft, A. Simone, L. Sluys, A novel constrained LArge Time INcrement method for modelling quasi-brittle failure, *Computer Methods in Applied Mechanics and Engineering* 265 (2013) 148–162. doi:10.1016/j.cma.2013.06.005.
- [35] F. Chinesta, P. Ladevèze, Proper generalized decomposition, in: *Model Order Reduction: Snapshot-Based Methods and Algorithms*, volume 2, De Gruyter, 2020, pp. 97–138. doi:10.1515/9783110671490-003.
- [36] C. Heyberger, P. A. Boucard, D. Néron, A rational strategy for the resolution of parametrized problems in the PGD framework, *Computer Methods in Applied Mechanics and Engineering* 259 (2013) 40–49. doi:10.1016/j.cma.2013.03.002.
- [37] D. Néron, P.-A. Boucard, N. Relun, Time-space PGD for the rapid solution of 3D nonlinear parametrized problems in the many-query context, *International Journal for Numerical Methods in Engineering* 103 (2015) 275–292. doi:10.1002/nme.4893.
- [38] L. Laurent, P. A. Boucard, B. Soulier, A dedicated multiparametric strategy for the fast construction of a cokriging metamodel, *Computers & Structures* 124 (2013) 61–73. doi:10.1016/j.compstruc.2013.03.012.
- [39] N. Courrier, P. A. Boucard, B. Soulier, The use of partially converged simulations in building surrogate models, *Advances in Engineering Software* 67 (2014) 186–197. doi:10.1016/j.advengsoft.2013.09.008.
- [40] S. Nachar, P.-A. Boucard, D. Néron, U. Nackenhorst, A. Fau, Multi-fidelity Metamodels Nourished by Reduced Order Models, in: P. Wriggers, O. Allix, C. Weïßenfels (Eds.), *Virtual Design and Validation, Lecture Notes in Applied and Computational Mechanics*, Springer International Publishing, Cham, 2020, pp. 61–79. doi:10.1007/978-3-030-38156-1_4.
- [41] D. Néron, D. Dureisseix, A computational strategy for thermo-poroelastic structures with a time–space interface coupling, *International Journal for Numerical Methods in Engineering* 75 (2008) 1053–1084. doi:10.1002/nme.2283.
- [42] M. Capaldo, P. A. Guidault, D. Néron, P. Ladevèze, The Reference Point Method, a “hyperreduction” technique: Application to PGD-based nonlinear model reduction, *Computer Methods in Applied Mechanics and Engineering* 322 (2017) 483–514. doi:10.1016/j.cma.2017.04.033.
- [43] S. Chaturantabut, D. C. Sorensen, Nonlinear Model Reduction via Discrete Empirical Interpolation, *SIAM Journal on Scientific Computing* 32 (2010) 2737–2764. doi:10.1137/090766498.
- [44] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, *International Journal for Numerical Methods in Engineering* 98 (2014) 625–662. doi:10.1002/nme.4668.