

Une version faiblement intrusive de la méthode LATIN-PGD pour la résolution de problèmes non-linéaires paramétrés dans les codes industriels

R. Scanff¹, D. Néron², P. Ladevèze², P. Barabinot¹

¹ Siemens Industry Software SAS, 92320 Châtillon, France,
{ronan.scanff, philippe.barabinot}@siemens.com

² Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190 Gif-sur-Yvette, France,
{david.neron, pierre.ladeveze}@ens-paris-saclay.fr

Résumé — Une version faiblement intrusive de la méthode LATIN-PGD est proposée pour construire des modèles réduits au sein de logiciels industriels généralistes par éléments finis, tels que Simcenter SamcefTM développé par Siemens Digital Industries Software. L'originalité de l'approche tient à son caractère générique permettant de gérer tous les types de non-linéarités classiquement rencontrés en mécanique du solide. Adossée à des problèmes paramétrés dépendant du temps, la méthodologie met en évidence un gain de performance.

Mots clés — Faiblement intrusif, LATIN-PGD, Modèles réduits, Non-linéaire

1 Introduction

Dans le cadre de l'industrie 4.0 liée à la digitalisation des entreprises, les simulations numériques occupent une place toujours plus importante. Toutefois, malgré l'avènement ces dernières années de gigantesques clusters possédant plusieurs milliers de cœurs, de nombreux problèmes non-linéaires – généralement assortis à l'exploration d'espaces paramétriques de grandes dimensions – demeurent encore inaccessibles. Pour répondre à ces attentes, de nouvelles méthodes numériques voient le jour comme les méthodes de réduction de modèles [1], reconnues notamment pour circonscire la malédiction de la dimensionnalité. Les méthodes Proper Orthogonal Decomposition (POD) [2] et Reduced Basis (RB) [3] sont caractérisées par une première étape d'apprentissage coûteuse (*offline*) émanant d'une multitude de résolutions hautes fidélités (*snapshots*). La méthode Proper Generalized Decomposition (PGD) [4] suit un chemin différent dans le sens où le modèle réduit est construit simultanément à la résolution du problème grâce à un processus de minimisation.

Bien que les méthodes de réduction de modèles soient le fruit de nombreux travaux de recherche académique, leur déploiement industriel demeure encore relativement limité. Pour devenir accessibles plus largement à l'ensemble de la communauté scientifique – y compris à des utilisateurs non experts – ces méthodes doivent pouvoir s'intégrer aisément dans les habitudes des bureaux d'études, ce qui passe par la volonté de les introduire nativement dans les logiciels industriels. À la connaissance des auteurs, les méthodes de réduction de modèles ne sont toujours pas implémentées dans les logiciels industriels généralistes par éléments finis et ne font donc pas partie des outils des ingénieurs au quotidien. Un des principaux obstacles se situe au niveau du caractère intrusif [5] de ces méthodes qui requièrent des opérations et un format de stockage des données atypique des codes commerciaux.

Des efforts récents ont porté sur le développement de méthodes de réduction de modèles non-intrusives. Orientées majoritairement sur des approches *a posteriori* POD ou RB dans un premier temps [6, 7], ce n'est que plus tardivement que des approches PGD non-intrusives sont apparues [8, 9]. Ces dernières restent basées sur l'utilisation de *snapshots* du fait de la formulation *a priori* de la PGD, intrinsèquement plus intrusive. De manière alternative, et pour des applications spécifiques, on peut trouver certaines implémentations non-intrusives dédiées à la PGD [10, 11, 12, 13] qui reposent systématiquement sur le couplage en externe entre un logiciel industriel et un code maison développé en laboratoire.

Sous hypothèses de quasi-staticité et de déplacements modérés en dehors des zones d'instabilité, les travaux présentés englobent l'ensemble des problèmes non-linéaires dépendant du temps que les logiciels

industriels par éléments finis sont susceptibles de traiter. Définie au moyen de quantités généralisées, une version faiblement intrusive de la méthode LATIN-PGD est élaborée, destinée à être intégrée nativement dans le logiciel industriel Simcenter SamcefTM. On bénéficie en particulier de tous les outils déjà présents dans le code industriel – tous types de non-linéarités, d’éléments, de chargements, etc. Un des principaux atouts réside dans le fait qu’il n’y a plus besoin de redévelopper des codes externes *ad-hoc*, la construction de modèles réduits étant menée de bout en bout dans les outils industriels en exploitant pleinement les procédures sous-jacentes optimisées depuis plusieurs décennies.

2 Méthodologie

On considère un problème standard classiquement résolu par n’importe quel logiciel industriel par éléments finis généraliste : au cours d’un certain intervalle de temps $I \subset \mathbb{R}_+$, une structure $\Omega \subset \mathbb{R}^3$ se déforme sous l’action de diverses sollicitations. Différentes sources de non-linéarités dépendant du temps peuvent intervenir : contacts, matériaux (visco-)plastiques ou endommageables, grandes transformations, etc. Écrit sous un formalisme de grandeurs généralisées sur l’espace éléments finis $\mathcal{V} \equiv \mathbb{R}^n$ avec n le nombre de degrés de liberté, on cherche le champ de déplacements $\underline{u} \in I_{\mathcal{V}} \equiv L^2(I; \mathcal{V})$ ainsi que les éventuels multiplicateurs de Lagrange $\underline{\xi} \in I_{\mathcal{V}_c} \equiv L^2(I; \mathcal{V}_c)$ – caractérisant le cas échéant les conditions de contact additionnelles, avec $\mathcal{V}_c \equiv \mathbb{R}^{n_c}$ et n_c le nombre de degrés de liberté associés aux contacts. En notant \underline{v} la trace des déplacements généralisés sur la zone de contact et $\underline{f}^{\text{ctc}}$ les efforts correspondants, le problème s’écrit :

Trouver $(\underline{u}, \underline{\xi}) \in I_{\mathcal{V}} \times I_{\mathcal{V}_c}$ vérifiant les conditions initiales $\underline{u}|_{t=0} = \underline{0}$ tels que :

- les déplacements généralisés \underline{u} coïncident avec les déplacements imposés \underline{u}_d – contraintes cinématiques ;

$$\forall t \in I, \quad \begin{cases} C_u(\underline{u}(t)) = \underline{u}_d(t) \\ C_v(\underline{u}(t)) = \underline{v}(t) \end{cases} \quad (1)$$

- les efforts externes $\underline{f}^{\text{ext}}$ contrebalancent les efforts internes $\underline{f}^{\text{int}}$ – équations d’équilibre – auxquels s’ajoutent éventuellement les efforts de contact $\underline{f}^{\text{ctc}}$;

$$\forall t \in I, \quad \begin{cases} \underline{f}^{\text{int}}(t, \underline{u}(t)) - \underline{f}^{\text{ext}}(t) + \underline{f}^{\text{ctc}}(t, \underline{\xi}(t)) = \underline{0} \\ \underline{f}^{\text{ctc}}(t, \underline{\xi}(t)) = C_{\xi}(\underline{\xi}(t)) \end{cases} \quad (2)$$

- les efforts internes généralisés $\underline{f}^{\text{int}}$ et les efforts de contact $\underline{f}^{\text{ctc}}$ résultent respectivement de l’intégration des lois matériaux et de contact non-linéaires – relation de comportement et de contact.

$$\forall t \in I, \quad \begin{cases} \underline{f}^{\text{int}}(t, \underline{u}(t)) = \mathcal{A}_u(t, \underline{u}(\tau \leq t)) \\ \mathcal{A}_c(t, \underline{v}(t), \underline{\xi}(t)) = \underline{0} \end{cases} \quad (3)$$

Les opérateurs C_u , C_v et C_{ξ} sont linéaires, tandis que les opérateurs non-linéaires \mathcal{A}_u et \mathcal{A}_c rendent compte respectivement des relations de comportement et des relations de contact. Ce type de problème est couramment résolu à de nombreuses reprises en fonction de certains paramètres $\underline{\mu} \in \mathcal{D}$ lors des processus d’optimisation. Ces paramètres peuvent être de différentes natures : chargements, variations de géométrie, données matériaux, etc. Classiquement, la résolution est orchestrée par un algorithme de Newton-Raphson dans les logiciels industriels. Ces travaux proposent d’utiliser un autre schéma pour linéariser les équations par application du solveur non-linéaire LATIN [14], propice à la mise en œuvre de la technique PGD.

2.1 Version LATIN-PGD faiblement intrusive

Cette version faiblement intrusive de la méthode LATIN-PGD diffère de la version la plus couramment employée ces dernières années, qui exploite la description à variables internes des matériaux [15]. *A contrario*, elle peut être vue comme une extension des premières versions fonctionnelles [16]. Toujours

basée sur la séparation des difficultés, les équations (1 – 3) sont réparties en deux groupes : la variété (A_d) rassemble les équations linéaires, éventuellement globales – Equations (1) et (2) –, tandis que la variété (Γ) concentre les équations non-linéaires mais locales – Equation (3). Puis, par l’intermédiaire de deux directions de recherche, on vient alternativement construire une approximation dans chacune des deux variétés (A_d) et (Γ) jusqu’à atteindre la convergence à l’intersection des deux espaces $s_\chi \in (A_d) \cap (\Gamma)$. Une direction de recherche verticale est mobilisée ($\hat{u}_\ell = \underline{u}_\ell$) au niveau de l’étape locale – (A_d) vers (Γ) – ce qui permet de s’adapter au mieux à l’existant du logiciel industriel. La seconde direction de recherche, intervenant à l’étape globale – (Γ) vers (A_d) –, constitue quant à elle un des paramètres majeurs de la méthode. Son choix influence directement la vitesse de convergence de l’algorithme. Ce point est précisé au niveau des illustrations.

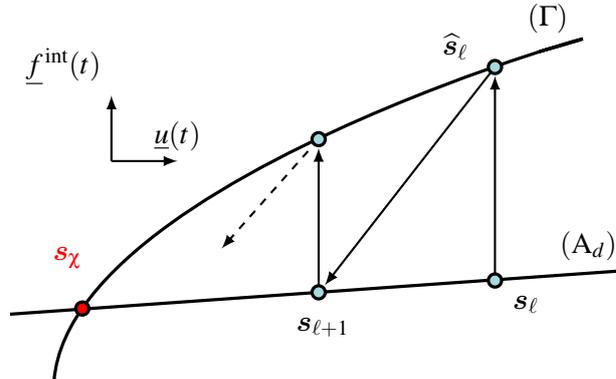


FIGURE 1 – Schéma itératif convergent méthode LATIN-PGD avec $s_\ell = (\underline{u}_\ell, \underline{f}_\ell^{\text{int}}, \underline{\xi}_\ell, \underline{v}_\ell)$

Les opérateurs non-linéaires \mathcal{A}_u et \mathcal{A}_c n’apparaissent pas naturellement dans les codes industriels car ils demandent une connaissance de l’ensemble de l’intervalle de temps à chaque itération ℓ de l’algorithme. Ces opérateurs sont donc construits de manière faiblement intrusive nativement dans le logiciel industriel lors de l’étape locale. Ceci constitue le seul point intrusif de notre approche. Par faiblement intrusif, nous entendons minimiser le nombre de modifications à apporter au logiciel industriel, une des particularités de notre approche étant précisément d’avoir accès au code source – ce qui n’est pas courant. Nous exploitons cet atout pour satisfaire au mieux les attentes en termes de performance et de robustesse, tout en conservant au maximum indépendantes les opérations qui peuvent le rester. La résolution de l’étape globale est ensuite menée par PGD en s’appuyant sur les outils pré-existants.

2.2 Contraintes d’implémentation

Dans un contexte industriel, la robustesse est tout aussi fondamentale que les performances. L’implémentation de la méthode LATIN-PGD dans le logiciel industriel Simcenter SamcefTM doit rester compatible avec l’architecture existante. Un souhait en particulier est de pouvoir utiliser les mêmes étapes de certification lors de la construction de chaque nouvelle version. Ces étapes garantissent qu’aucun effet indésirable n’est introduit par l’ajout de la méthodologie LATIN-PGD. Une attention particulière est accordée au stockage de données sous forme à variables séparées qui n’est pas prévu en standard dans les logiciels commerciaux à architecture incrémentale.

Sous forme d’un module Fortran, une interface dédiée à la méthode est développée permettant de conserver indépendants tous les ingrédients liés à la technique PGD. L’ensemble des informations transitent via cette interface qui réalise le lien entre le code industriel pré-existant et les nouvelles capacités offertes par la méthodologie LATIN-PGD développée. La principale force de cette approche faiblement intrusive réside dans la possibilité de pouvoir s’appuyer sur toute la richesse du logiciel industriel dans la diversité des cas qu’il est susceptible de traiter. Toute l’intelligence emmagasinée depuis de nombreuses années est utilisable sans modification lors de la réalisation de l’étape locale : tous les types d’éléments, tous les types de non-linéarités, tous les types de chargements, etc.

2.3 Gestion des paramètres

Après discrétisation de l'espace paramétrique \mathcal{D} suivant une grille régulière, une solution en chacune des valeurs discrètes des paramètres est exhibée par application de proche en proche de la méthodologie LATIN-PGD faiblement intrusive. On exploite les techniques non-intrusives développées notamment dans [17]. Ceci s'avère peu coûteux en mobilisant une des propriétés remarquables de l'algorithme LATIN-PGD – à savoir pouvoir être initialisé à partir de n'importe quelle solution précédemment calculée. De manière plus précise, supposons que l'on cherche à calculer un nouveau point $(\underline{\mu} + \Delta\underline{\mu}) \in \mathcal{D}$ de l'espace paramétrique. Connaissant la solution spatio-temporelle $s^{(\underline{\mu})}$ pour le jeu de paramètres $\underline{\mu} \in \mathcal{D}$ précédent, ainsi que les modes spatiaux $\{\Lambda_i\}_{1 \leq i \leq m^{(\underline{\mu})}}$ correspondants, la première itération de la méthode LATIN-PGD se déroule comme suit :

- **Étape locale** : étant donnée $s_0^{(\underline{\mu})}(t)$, on cherche $\widehat{s}_0^{(\underline{\mu} + \Delta\underline{\mu})}(t)$ tel que :

$$\underline{u}_0^{(\underline{\mu} + \Delta\underline{\mu})}(t) = \underline{u}_0^{(\underline{\mu})}(t) \quad \forall t \in I \quad (4)$$

- **Étape linéaire** : étant donnée $\widehat{s}_0^{(\underline{\mu} + \Delta\underline{\mu})}(t)$, on cherche $s_1^{(\underline{\mu} + \Delta\underline{\mu})}(t)$ où de l'étape préliminaire de mise à jour des fonctions temporelles $\{\lambda_i^{(\underline{\mu})}\}_{1 \leq i \leq m^{(\underline{\mu})}}$, on détermine la nouvelle approximation :

$$\underline{u}_1^{(\underline{\mu} + \Delta\underline{\mu})}(t) = \sum_{i=1}^{m^{(\underline{\mu})}} \lambda_i^{(\underline{\mu} + \Delta\underline{\mu})}(t) \Lambda_i \quad (5)$$

puis l'algorithme continu classiquement avec $\ell \geq 1$ comme détaillé précédemment. Pour de nombreuses valeurs de paramètres, il n'est pas nécessaire d'ajouter de nouveaux modes à la base réduite. Après quelques itérations, on atteint la convergence – pilotée par certains indicateurs d'erreur – et l'algorithme s'arrête pour entamer la gestion du point suivant dans l'espace paramétrique \mathcal{D} , ce qui représente un gage de performance.

3 Résultats sur la base d'une illustration industrielle

Le cas-test concerne une aube haute pression d'un réacteur d'avion. Le maillage constitué de 5 millions de degrés de liberté a été réalisé par l'intermédiaire du logiciel Simcenter 3DTM permettant de réaliser facilement toutes les étapes de pré et post-traitements des calculs. Des efforts latéraux en tête d'aube viennent compléter les efforts centrifuges ; le pied d'aube étant considéré comme fixe. Le matériau utilisé suit une loi de Chaboche élasto-visco-plastique à 9 paramètres qui dépendent de la température.

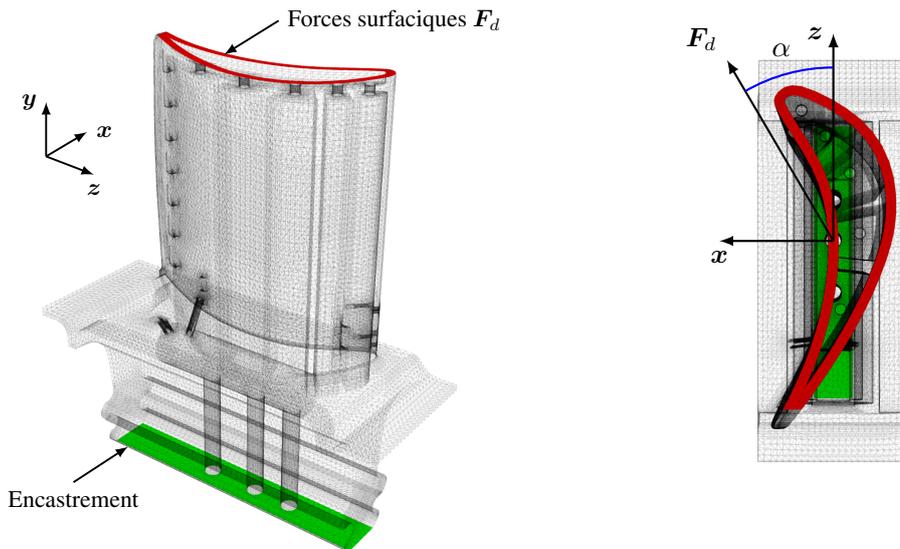


FIGURE 2 – Géométrie et chargements appliqués sur l'aube de réacteur

On mène une étude paramétrique sur l'espace à deux dimensions $\mathcal{D} = \mathcal{D}_\alpha \times \mathcal{D}_T$ où l'angle de chargement $\alpha \in \mathcal{D}_\alpha = [45, 170]$ en degrés (cf. Figure 2) et la température $T \in \mathcal{D}_T = [850, 1010]$ en degrés

Celsius représentent les deux paramètres du modèle. Ces paramètres possèdent une influence sur les deux variétés (A_d) et (Γ) : chaque nouvelle valeur de la température T affecte les 9 paramètres matériaux et donc principalement (Γ) tandis que chaque nouvelle valeur de l'angle de chargement α impacte les efforts extérieurs intervenant dans (A_d).

En tant que paramètres de la méthode LATIN-PGD, plusieurs directions de recherche au niveau de l'étape globale ont été étudiées : soit l'utilisation d'un opérateur tangent cohérent $H_\ell(t)$ dépendant du temps, soit la conservation de la rigidité élastique K – constant et indépendant du temps. Le premier présente une convergence optimale en terme de nombres d'itération ℓ ; le second s'avère plus performant du fait de la mutualisation possible de la factorisation de l'opérateur au fil des itérations LATIN-PGD. Dans le cadre des déplacements modérés, pour les problèmes élasto-visco-plastiques, avec contacts éventuels, tous nos tests réalisés donnent une large supériorité à l'utilisation d'une rigidité élastique K dès que le problème considéré devient grand. Le Tableau 1 fournit la comparaison des temps de calcul pour la réalisation d'un point tiré au hasard dans l'espace paramétrique. Par le processus de réutilisation des modes préalablement établi, le temps total de chacun des points diminue globalement à mesure que la base réduite s'enrichit. La Figure 3 détaille plus précisément la répartition des temps de calcul en utilisant l'opérateur K constant où on constate le poids particulièrement important de l'étape locale.

TABLE 1 – Comparaison des temps de calcul et du nombre d'itérations selon deux directions de recherche : direction tangente $H_\ell(t)$ et direction élastique K (pour le calcul d'un point dans l'espace paramétrique \mathcal{D})

		Temps de calcul [s]				Nombre d'itération	
	Étape locale	Étape globale	Compléments	Total			
$H_\ell(t)$	1324 (32.9%)	2373 (59.0%)	326 (8.1%)	4023 (100%)	-		
K	2206 (73.9%)	354 (11.9%)	425 (14.2%)	2985 (100%)	26		

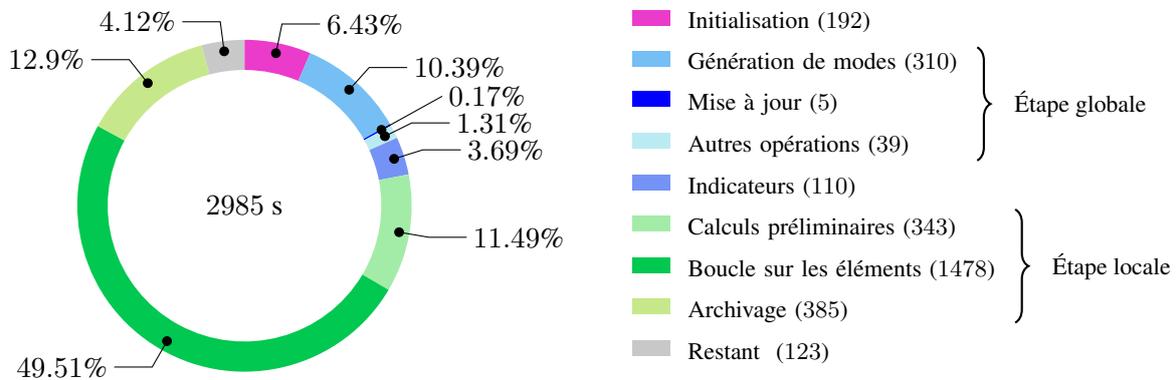


FIGURE 3 – Répartition des temps de calcul pour l'opérateur K constant (en secondes)

La construction du modèle réduit passe par le calcul de 255 points uniformément répartis dans l'espace paramétrique \mathcal{D} . L'application de proche en proche de la stratégie multi-paramétrique explicitée précédemment permet d'atteindre un facteur d'accélération de plus de 3 fois comparé à l'utilisation classique du logiciel commercial (18 jours avec la version commerciale suivant un algorithme de Newton-Raphson contre 6 jours seulement avec la stratégie LATIN-PGD faiblement intrusive déployée nativement dans le logiciel industriel Simcenter SamcefTM). Ce gain serait sans doute plus important avec plus de points ou plus de paramètres.

4 Conclusion

Le cœur de ces travaux porte sur l'introduction de la méthodologie LATIN-PGD dans le logiciel industriel généraliste Simcenter SamcefTM de manière faiblement intrusive. C'est la première fois que la méthode LATIN-PGD est implémentée à un aussi haut niveau dans un logiciel industriel généraliste, ce

qui permet d’atteindre des gains intéressants en terme de performance, tout en permettant la construction native et efficace de modèles réduits. Des stratégies d’hyper-réductions sont actuellement regardées pour réduire le poids de l’étape locale, ce qui permettrait d’atteindre des gains encore plus impressionnants. Une des principales forces de l’approche développée réside dans sa robustesse : l’ensemble des capacités pré-existantes du logiciel industriel, optimisées depuis de longues années, peuvent être réutilisées sans avoir besoin de recoder le traitement des diverses non-linéarités déjà prises en charge. De bons résultats ont déjà été obtenus avec des problèmes variés comportant des non-linéarités matériaux, des phénomènes de contact ou encore des grandes rotations. Les avantages reconnus des modèles réduits peuvent alors être mobilisés pour la résolution de problèmes mécaniques non-linéaires suivant un processus de bout en bout au sein d’une chaîne de simulation industrielle.

Remerciements

Nous remercions Siemens Digital Industries Software ainsi que l’Association Nationale de la Recherche et de la Technologie (ANRT) pour le financement de ces travaux.

Références

- [1] Peter Benner, Stephano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, and Luis M. Silveira, editors. *Model Order Reduction : Snapshot-Based Methods and Algorithms*, volume 2. De Gruyter, Berlin, Boston, 2021.
- [2] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7) :808–817, 2000.
- [3] Yvon Maday and Einar M. Rønquist. A reduced-basis element method. *Comptes Rendus Mathématique*, 335(2) :195–200, January 2002.
- [4] Francisco Chinesta and Pierre Ladevèze, editors. *Separated Representations and PGD-based Model Reduction : Fundamentals and Applications*. Number 554 in Courses and Lectures / International Centre for Mechanical Sciences. Springer, Wien, 2014.
- [5] Loïc Giraldi, Dishu Liu, Hermann G. Matthies, and Anthony Nouy. To Be or Not to be Intrusive? The Solution of Parametric and Stochastic Equations—Proper Generalized Decomposition. *SIAM Journal on Scientific Computing*, 37(1) :A347–A368, January 2015.
- [6] Christophe Audouze, Florian De Vuyst, and Prasanth B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5) :1587–1628, 2013.
- [7] Fabien Casenave, Asven Gariah, Christian Rey, and Frederic Feyel. A nonintrusive reduced order model for nonlinear transient thermal problems with nonparametrized variability. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1) :22, May 2020.
- [8] Rubén Ibáñez, Emmanuelle Abisset-Chavanne, Amine Ammar, David González, Elías Cueto, Antonio Huerta, Jean Louis Duval, and Francisco Chinesta. A Multidimensional Data-Driven Sparse Identification Technique : The Sparse Proper Generalized Decomposition. *Complexity*, 2018.
- [9] Domenico Borzacchiello, José V. Aguado, and Francisco Chinesta. Non-intrusive Sparse Subspace Learning for Parametrized Problems. *Archives of Computational Methods in Engineering*, 26(2) :303–326, April 2019.
- [10] Amaury Courard. *PGD-Abaques virtuels pour l’optimisation géométrique des structures*. PhD thesis, Ecole Normale Supérieure de Cachan, Université Paris-Saclay, January 2016.
- [11] X. Zou, M. Conti, P. Díez, and F. Auricchio. A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113(2) :230–251, 2018.
- [12] Angel Leon, Sebastien Mueller, Patrick de Luca, Rajab Said, Jean-Louis Duval, and Francisco 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(1) :1–20, December 2019.
- [13] Vasileios Tsiolakis, Matteo Giacomini, Ruben Sevilla, Carsten Othmer, and Antonio Huerta. Nonintrusive proper generalised decomposition for parametrised incompressible flow problems in OpenFOAM. *Computer Physics Communications*, 249 :107013, April 2020.
- [14] Pierre 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(2) :41–44, 1985.

- [15] Pierre Ladevèze. *Nonlinear Computational Structural Mechanics : New Approaches and Non-Incremental Methods of Calculation*. Springer, New York, 1999.
- [16] Philippe Boisse, Pascal Bussy, and Pierre Ladevèze. A new approach in non-linear mechanics : The large time increment method. *International Journal for Numerical Methods in Engineering*, 29(3) :647–663, 1990.
- [17] Nicolas Relun, David Néron, and Pierre-Alain Boucard. A model reduction technique based on the PGD for elastic-viscoplastic computational analysis. *Computational Mechanics*, 51(1) :83–92, January 2013.