

Sur une vision non intrusive de la méthode LaTIn-PGD en non linéaire

R. Scanff^{a,b}, D. Néron^a, P. Ladevèze^a

a. LMT, ENS Paris-Saclay, CNRS, Université Paris-Saclay -
{ronan.scanff, david.neron, pierre.ladeveze}@lmt.ens-cachan.fr

b. SIEMENS Industry Software SAS - ronan.scanff@siemens.com

Résumé :

Une reformulation de la méthode LaTIn-PGD sous forme de grandeurs généralisées est proposée dans l'optique de coupler méthodes de réduction de modèles et codes éléments finis industriels. Ces travaux proposent d'intervenir directement au niveau du solveur non linéaire afin de pouvoir manipuler des modèles réduits de manière intrinsèque. Le point clé de la méthode réside dans le fait que l'introduction de la PGD en natif dans les codes de calculs commerciaux tels que le logiciel Samcef, développé par SIEMENS, nécessite de connaître une approximation de la solution sur l'ensemble espace-temps-paramètres à chaque étape de la résolution, ce que fournit naturellement la méthode LaTIn. Un cas test sur une loi de comportement élasto-visco-plastique permet d'appréhender un premier gain de temps CPU.

Abstract :

A reformulation of the LaTIn-PGD method in the form of generalized quantities is proposed in order to couple model order reduction methods and industrial codes using finite elements. This work proposes to operate directly at the level of the non-linear solver in order to be able to manipulate reduced models in an intrinsic way. The key point of the method is that the introduction of the PGD natively in commercial calculation codes, such as Samcef software developed by SIEMENS, requires an approximation of the solution on the space-time-parameter set at each step of the resolution, which is naturally provided by the LaTIn method. A test case on an elasto-visco-plastic behavioural law is performed with a reduction of calculation time.

Mots clefs : LaTIn-PGD, modèles réduits, non linéaire, non intrusif, Samcef

1 Introduction

La simulation numérique s'est immiscée ces dernières années au coeur des méthodologies de conception des entreprises. Toutefois, la volonté d'obtenir des résultats dans des délais courts, compatibles avec les besoins industriels, se heurte à la volonté d'avoir des simulations au plus proche de la réalité, donc nécessairement gourmandes en temps CPU. Pour lever ce verrou scientifique, nous nous intéressons aux modèles réduits qui possèdent par essence la capacité de réduire la complexité apparente d'un problème tout en préservant sa qualité. Ainsi, les modèles réduits représentent un atout de taille pour la simulation numérique, tant pour la réalisation de calculs très fins que pour la réalisation d'une multitude de calculs multi-paramétriques dans une démarche d'optimisation, en vue de construire des abaques virtuels par exemple. Cependant, bien que très démocratisés dans le milieu académique, ces outils le sont beaucoup moins dans le secteur industriel ; une des principales causes étant que, du fait de la forte intrusivité des algorithmes associés, les modèles réduits ne sont encore que très peu implémentés dans les codes de calculs commerciaux. Le but de ces travaux est de proposer une méthode permettant de remédier à ce constat. Plusieurs méthodes de réduction de modèles existent dans la littérature : parmi les plus répandues se trouvent les méthodes Proper Orthogonal Decomposition (POD), Reduced Basis (RB) ou encore Proper Generalized Decomposition (PGD) [3, 4]. Pour un problème dépendant du temps, les deux premières possèdent une phase d'apprentissage (Offline) au cours de laquelle le problème initial est résolu pour certains pas de temps, choisis judicieusement, suivie d'une seconde phase (Online) permettant alors de déceler la meilleure approximation de la solution dans la base précédemment exhibée. Nous nous plaçons dans le cadre de la troisième méthode et plus spécifiquement de la méthode LaTIn-PGD [10]. Cette méthode suit une philosophie différente puisqu'elle permet de calculer directement la solution du problème sans nécessiter la résolution de problèmes connexes, la base réduite étant générée à la volée en même temps que la solution du problème. Couplée au solveur non-linéaire LaTIn, la méthode LaTIn-PGD permet de traiter efficacement les problèmes complexes comme les problèmes multi-échelles et multi-physiques [13] ou encore visco-plastiques [14]. Les travaux présentés portent sur une reformulation généralisée de l'approche LaTIn-PGD qui soit implémentable de manière peu intrusive dans les codes commerciaux. Cette intervention directement au niveau du solveur a pour vocation de réduire considérablement le temps de calcul, en tirant partie de la puissance des modèles réduits, tout en conservant la généralité et la richesse du code commercial dans la variété des cas qu'il peut traiter. Dans le cadre de ces travaux, cette approche est réalisée dans le code de calcul non linéaire Samcef développée par SIEMENS.

2 Reformulation fonctionnelle de la méthode LaTIn-PGD

2.1 Hypothèses et notations

Le cadre de cette étude se limite aux problèmes d'évolutions quasi-statiques (accélérations négligées) isothermes sous l'hypothèse de petites perturbations. De plus, on se munit d'un cadre fonctionnel adéquat pour l'utilisation de la méthode des éléments finis. On travaille donc dans l'ensemble des espaces de Hilbert (fonctions de carré intégrable $L^2(\Omega)$, espace de Sobolev $H^1(\Omega)$) sur le domaine d'étude Ω , un ouvert borné de \mathbb{R}^3 .

2.2 Position du problème de référence

Nous considérons un système, représenté par son domaine $\Omega \subset \mathbb{R}^3$, soumis à des forces volumiques \mathbf{f}_d . Des efforts surfaciques \mathbf{F}_d , sur une partie $\partial_2\Omega$ de Ω , ainsi que des déplacements \mathbf{u}_d , sur la partie complémentaire $\partial_1\Omega$, sont également imposés (cf. Figure 1). Nous étudions l'évolution de ce système sur un intervalle de temps $I = [0, T]$. L'ensemble des différentes quantités sont définies sur $\Omega \times I$ et sont, par hypothèse, suffisamment régulières. De plus, on suppose toutes les conditions initiales nulles.

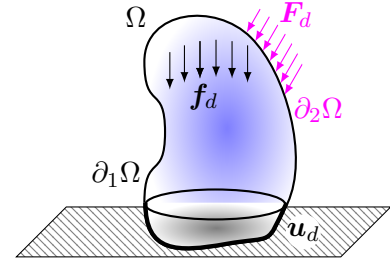


FIGURE 1 – Problème de référence

L'évolution du système est gouvernée par un ensemble d'équations dont on introduit les différents espaces d'admissibilités :

□ \mathcal{U} l'espace des champs cinématiquement admissibles :

$$\mathbf{u} \in \mathcal{U} \equiv \left\{ \mathbf{u}(\mathbf{x}, t) \mid \tilde{\mathbf{u}} : I \longrightarrow H^1(\Omega), \tilde{\mathbf{u}}(t)(\mathbf{x}) = \mathbf{u}(\mathbf{x}, t) \text{ et } \mathbf{u} = \mathbf{u}_d \text{ sur } \partial_1\Omega \right\}$$

□ \mathcal{U}_0 l'espace des champs cinématiquement admissibles à zéro :

$$\mathbf{u} \in \mathcal{U}_0 \equiv \left\{ \mathbf{u}(\mathbf{x}, t) \mid \tilde{\mathbf{u}} : I \longrightarrow H_0^1(\Omega), \tilde{\mathbf{u}}(t)(\mathbf{x}) = \mathbf{u}(\mathbf{x}, t) \text{ et } \mathbf{u} = \mathbf{0} \text{ sur } \partial_1\Omega \right\}$$

□ \mathcal{S} l'espace des champs statiquement admissibles :

$$\boldsymbol{\sigma} \in \mathcal{S} \equiv \left\{ \boldsymbol{\sigma}(\mathbf{x}, t) \mid \forall \mathbf{u}^* \in \mathcal{U}_0, \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{u}^*) \, d\Omega = \int_{\Omega} \mathbf{f}_d \cdot \mathbf{u}^* \, d\Omega + \int_{\partial_2\Omega} \mathbf{F}_d \cdot \mathbf{u}^* \, d\Gamma \right\}$$

Le problème à résoudre s'exprime alors suivant l'équation (1) où l'expression $\boldsymbol{\sigma} = \mathcal{H}(\boldsymbol{\varepsilon}(\mathbf{u}))$ symbolise la loi de comportement du matériau avec $\boldsymbol{\varepsilon}$ le gradient symétrique du champ de déplacement \mathbf{u} :

$$\text{Trouver } (\mathbf{u}, \boldsymbol{\sigma}) \in \mathcal{U} \times \mathcal{S} \text{ tels que : } \boldsymbol{\sigma} = \mathcal{H}(\boldsymbol{\varepsilon}(\mathbf{u})) \quad (1)$$

2.3 Mise en place du solveur non-linéaire LaTIn

L'ensemble de ces équations est résolu via la méthode LaTIn [10], une méthode itérative non incrémentale qui, à chaque itération, produit une meilleure approximation de la solution sur l'ensemble de l'espace et du temps. Deux grandes visions de la méthode LaTIn coexistent : la vision à variables internes [8, 9] qui est la vision la plus répandue de la méthode et qui a connu de multiples développements ces 30 dernières années ([2, 13, 14] à guise d'exemples, pour ne citer que quelques travaux), et la vision fonctionnelle [1, 5, 6, 7] bien moins répandue car jugée légèrement moins performante vis-à-vis de la première. Cependant, cette seconde vision possède un atout considérable : elle se présente comme bien moins intrusive du point de vue algorithmique, et est donc à la base des travaux présentés ici [11].

Par souci de commodité et pour alléger les écritures, nous réécrivons le problème (1) en terme de déplacement et d'effort généralisés selon le formalisme éléments finis. Ainsi, en notant avec un exposant \bullet^h les grandeurs généralisées, le problème (1) est équivalent au problème (2) où l'expression $\mathbf{F}_{int}^h = \mathcal{T}(t, \mathbf{u}^h(\tau \leq t))$ symbolise la loi de comportement du matériau, expression dépendant de l'histoire du chargement sous un formalisme fonctionnel :

$$\text{Trouver } (\mathbf{u}^h, \mathbf{F}_{int}^h) \in \mathcal{U}^h \times \mathcal{F}^h \text{ tels que : } \mathbf{F}_{int}^h = \mathcal{T}(t, \mathbf{u}^h(\tau \leq t)) \quad (2)$$

Ce problème peut être classiquement résolu de manière incrémentale par un schéma de type Newton-Raphson. Nous choisissons d'effectuer la résolution au moyen de la méthode LaTIn.

Schématiquement, on représente cet algorithme itératif non incrémental comme présenté sur la Figure 2 où deux variétés \mathbf{A}_d et Γ sont définies. La variété \mathbf{A}_d regroupe les équations linéaires, éventuellement globales, du problème tandis que la variété Γ regroupe les équations locales, éventuellement non linéaires de celui-ci. La solution du problème (2) résulte alors de la résolution de ce schéma itératif alterné à deux directions de recherche représentant des paramètres de la méthode.

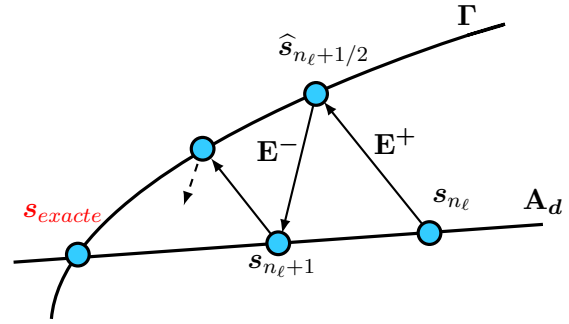


FIGURE 2 – Schéma itératif convergent

Nous définissons donc nos deux variétés \mathbf{A}_d et Γ telles que le problème (2) s'écrive :

$$\text{Trouver } (\mathbf{u}^h, \mathbf{F}_{int}^h) \text{ tels que : } \begin{cases} (\mathbf{u}^h, \mathbf{F}_{int}^h) \in \mathbf{U}^h \times \mathcal{F}^h & \forall t \in I \quad (\mathbf{A}_d) \\ \mathbf{F}_{int}^h = \mathcal{T}(t, \mathbf{u}^h(\tau \leq t)) & \forall t \in I \quad (\Gamma) \end{cases} \quad (3)$$

L'algorithme LaTin nous conduit à opérer une succession d'étapes locales et d'étapes linéaires au moyen de deux directions de recherche : connaissant à l'itération n_ℓ la quantité $\mathbf{s}_{n_\ell} = (\mathbf{u}_{n_\ell}^h, \mathbf{F}_{int, n_\ell}^h)$, nous cherchons la quantité $\mathbf{s}_{n_\ell+1} = (\mathbf{u}_{n_\ell+1}^h, \mathbf{F}_{int, n_\ell+1}^h)$ en deux temps. Tout d'abord, au moyen de l'étape locale, nous cherchons la quantité $\hat{\mathbf{s}} = (\hat{\mathbf{u}}^h, \hat{\mathbf{F}}_{int}^h)$ étant donné la quantité $\mathbf{s}_{n_\ell} = (\mathbf{u}_{n_\ell}^h, \mathbf{F}_{int, n_\ell}^h)$. Nous choisissons pour ce faire la direction de recherche verticale, c'est-à-dire :

$$\hat{\mathbf{u}}^h = \mathbf{u}_{n_\ell}^h \quad \forall t \in I \quad (4)$$

La connaissance de $\hat{\mathbf{F}}_{int}^h$ découle directement de la relation de comportement $\hat{\mathbf{F}}_{int}^h = \mathcal{T}(t, \hat{\mathbf{u}}^h(\tau \leq t))$. Cette étape est locale dans la mesure où la relation de comportement est locale en espace, la cinématique étant imposée. L'intégration du comportement peut donc s'effectuer en chaque points de Gauss de chaque éléments à chaque pas de temps. Cette étape est une étape classique effectuée dans les codes commerciaux, elle a donc l'avantage sous cette forme d'être non-intrusive dans le sens où on va pouvoir allègrement réutiliser les routines développées dans les codes éléments finis. L'étape suivante est l'étape linéaire qui consiste à trouver $\mathbf{s}_{n_\ell+1} = (\mathbf{u}_{n_\ell+1}^h, \mathbf{F}_{int, n_\ell+1}^h)$ connaissant $\hat{\mathbf{s}} = (\hat{\mathbf{u}}^h, \hat{\mathbf{F}}_{int}^h)$. Nous introduisons, en tant que paramètre de la méthode, l'opérateur \mathbf{H} associé à la direction de recherche de montée qui vérifie l'équation (5) écrite en terme de déplacements et d'efforts généralisés :

$$\mathbf{H}(\mathbf{u}_{n_\ell+1}^h - \hat{\mathbf{u}}^h) = \mathbf{F}_{int, n_\ell+1}^h - \hat{\mathbf{F}}_{int}^h \quad \forall t \in I \quad (5)$$

Le choix de la direction de recherche est le paramètre majeur de la méthode. Afin d'assurer la convergence de l'algorithme, l'opérateur \mathbf{H} est choisi symétrique défini positif [10]. Le choix optimal en terme de vitesse de convergence serait alors de considérer l'opérateur tangent, tandis qu'un choix moins coûteux à calculer pourrait être de ne considérer que l'opérateur de Hooke élastique.

Ainsi, sous cette forme, l'algorithme LaTin peut être vu comme l'analogue de l'algorithme de Newton-Raphson auquel on a permuté les boucles de résolutions (boucle sur les itérés et boucle sur les itérations). Toutefois, le fait de raisonner sur l'ensemble de l'espace et du temps nous amène à manipuler des quantités qui sont très coûteuses en espace de stockage et en temps de calculs. L'astuce consiste alors à utiliser une représentation astucieuse sous forme à variables séparées en mettant en oeuvre la PGD.

2.4 Résolution par approche de réduction de modèles PGD

La résolution des systèmes linéaires se déroulant lors de l'étape *linéaire*, c'est lors de celle-ci que nous faisons intervenir la PGD. L'approximation PGD consiste alors à chercher la solution $\mathbf{u}_{n_\ell+1}^h$ sous la forme d'une somme finie de fonctions à variables séparées du type (6) :

$$\mathbf{u}_{n_\ell+1}^h(t) \approx \mathbf{u}_{n_\ell+1, n_p}^h(t) = \sum_{i=1}^{n_p} \lambda_i(t) \mathbf{\Lambda}_i^h + \mathbf{u}_0^h(t) \quad (6)$$

où les conditions aux limites non homogènes sont prises en compte au moyen d'un relèvement à partir de la solution élastique, et avec $\{\lambda_i\}_{i=1..n_p} \in \mathcal{I} \equiv L^2(I)$ et $\{\mathbf{\Lambda}_i\}_{i=1..n_p} \in \mathcal{V} \equiv L^2(\Omega)$. Ces dernières constituent les bases réduites en temps et en espace respectivement. L'algorithme glouton mis en oeuvre pour la résolution de l'étape *linéaire* comprend deux phases : une première consistant à effectuer une phase d'update des modes temporels déjà constitués, les modes spatiaux étant conservés, et une seconde consistant à générer éventuellement un nouveau mode, en fonction du résultat de l'étape d'update précédente. Plus précisément, soit l'étape d'update est efficace et auquel cas nous considérons qu'il n'est pas nécessaire de produire un mode supplémentaire dans le sens où la base spatio-temporelle déjà établie est capable de représenter convenablement la solution de l'équation (5), soit l'étape d'update n'est pas suffisamment efficace et auquel cas nous décidons d'ajouter un mode supplémentaire. Au fur et à mesure des itérations LaTIn, nous venons ainsi apporter une correction à la solution précédemment exhibée à l'itération précédente jusqu'à convergence de l'algorithme. Nous utilisons η comme critère d'arrêt, la norme énergétique de la différence entre $\mathbf{u}_{n_\ell+1}^h$ et $\hat{\mathbf{u}}^h$ moyennée sur le temps.

3 Exemple numérique

Sans perte de généralités, nous présentons le cas pour des matériaux présentant une loi de comportement élasto-visco-plastique avec écrouissage cinématique non-linéaire et écrouissage isotrope de type Chaboche [12]. Dans ce qui suit, \mathbf{s}_\bullet désigne la partie déviatorique de la quantité \bullet .

Pour ce type de loi, l'expression générique $\boldsymbol{\sigma} = \mathcal{H}(\boldsymbol{\varepsilon}(\mathbf{u}))$, sous conditions isothermes, se décline de la manière suivante :

$$\begin{aligned} & \square \text{ Partie élastique du comportement (équation d'état) : } \boldsymbol{\sigma} = \mathcal{K} : \boldsymbol{\varepsilon}^e \\ & \square \text{ Partie plastique du comportement (équations d'évolution) : } \dot{\boldsymbol{\varepsilon}}^p = \frac{3}{2} \dot{p} \frac{\mathbf{s}_\sigma - \mathbf{s}_x}{J_2(\mathbf{s}_\sigma - \mathbf{s}_x)} \\ & \text{avec } \begin{cases} \dot{p} = \left\langle \frac{J_2(\boldsymbol{\sigma} - \mathbf{X}) - R(p) - \sigma_0}{K(p)} \right\rangle^n & p \text{ déformation plastique cumulée} \\ \dot{\mathbf{X}} = \frac{2}{3} C \dot{\boldsymbol{\varepsilon}}^p - \gamma(p) \mathbf{X} \dot{p} & \mathbf{X} \text{ centre actuel du domaine d'élasticité} \end{cases} \end{aligned}$$

où \mathcal{K} symbolise le tenseur de Hooke, $R(p) = R_\infty(1 - e^{-bp})$ représente la force thermodynamique associée à la variable interne indiquant l'état actuel d'écrouissage du matériau, J_2 représente le deuxième invariant (correspondant à la contrainte équivalente de Von Mises), et \mathbf{X} se compose d'un terme d'écrouissage cinématique (linéaire) et d'un terme de rappel fournissant un effet mémoire évanescence du trajet de déformation. Pour simplifier, on suppose de plus $K(p)$ et $\gamma(p)$ constant. Finalement, cette loi de comportement possède sept paramètres matériaux : σ_0 , n , K , R_∞ , b , C et γ avec σ_0 la limite d'élasticité.

Les résultats issus du calcul pour une structure 3D correspondant à une barre entaillée sollicitée en traction cyclique suivant l'axe z et encastree à l'autre extrémité (cf. Figure 3) sont présentés. Cette structure possède 300 000 degrés de liberté spatiaux.

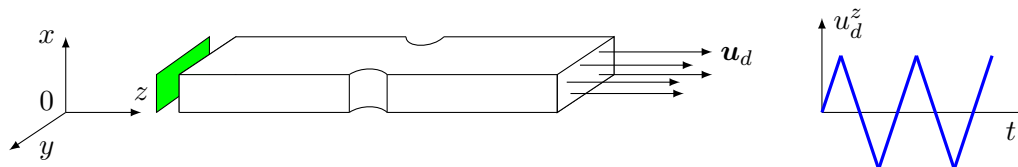


FIGURE 3 – Barre entaillée 3D sous sollicitation cyclique

Nous comparons le résultat obtenu avec la méthode LaTIn-PGD codée sous MATLAB à une solution de référence obtenue avec le logiciel Samcef. Nous définissons le gain de temps ζ comme le rapport entre le temps de calculs Samcef et le temps de calculs de la méthode LaTIn-PGD. Ainsi, notre objectif consiste à obtenir un ζ le plus élevé possible. Nous définissons également δ l'erreur par rapport la référence comme la différence des champs de déplacements au sens de la norme énergétique :

$$\delta = \frac{\|\mathbf{u}_L - \mathbf{u}_S\|}{\|\mathbf{u}_S\|} \quad \text{avec} \quad \|\bullet\| = \int_{\Omega \times I} \varepsilon(\bullet) : \mathcal{K} : \varepsilon(\bullet) \, d\Omega dt \quad (7)$$

L'un des principaux avantages de la méthode LaTIn-PGD réside dans l'obtention de solutions partiellement convergées. En effet, en fonction de l'étude menée, il n'est pas forcément nécessaire d'avoir une solution parfaitement convergée, une approximation de celle-ci peut amplement suffire. Le Tableau 1 présente le gain en temps et l'erreur par rapport à la solution de référence évaluée avec Samcef (critère d'arrêt TESF par défaut égal à $1e^{-3}$) pour différents niveau de fidélité η .

	Temps (s)	Speed-up ζ	δ (%)	Von Mises (MPa)
Samcef (Newton-Raphson)	14 773	-	-	215.26
LaTIn-PGD ($\eta = 1e^{-2}$)	537	27.51	3.56	[211.39 - 218.70]
LaTIn-PGD ($\eta = 1e^{-3}$)	1 062	13.91	1.14	[215.08 - 215.56]
LaTIn-PGD ($\eta = 2e^{-4}$)	1 738	8.5	0.67	[215.11 - 215.77]

TABLE 1 – Comparaison des temps de calcul des différentes méthodes

Plus précisément, les Figures 4 et 5 présentent respectivement la décroissance de l'erreur δ en fonction de l'inverse du gain $\frac{1}{\zeta}$ et l'évolution de la contrainte de Von Mises pour l'élément le plus chargé en fonction de l'inverse du gain $\frac{1}{\zeta}$.

4 Conclusions

Ces travaux ont permis de mettre en évidence un premier gain de temps en faveur de la méthode LaTIn-PGD comparée à un code industriel d'une part, tout en prenant en compte la contrainte d'être le moins intrusif possible d'autre part, c'est-à-dire en réutilisant, autant que faire se peut, les différentes structures algorithmiques et opérations usuellement utilisées en standard dans les codes éléments finis. Ainsi, ce premier démonstrateur de performance ne figure être qu'une étape d'un projet plus vaste, dont la finalité se trouve dans l'intégration complète d'une méthode de réduction de modèle, en natif, dans un code commercial. L'implémentation de la méthode LaTIn-PGD dans le code industriel Samcef, ainsi que son optimisation, figurent parmi les travaux en cours.

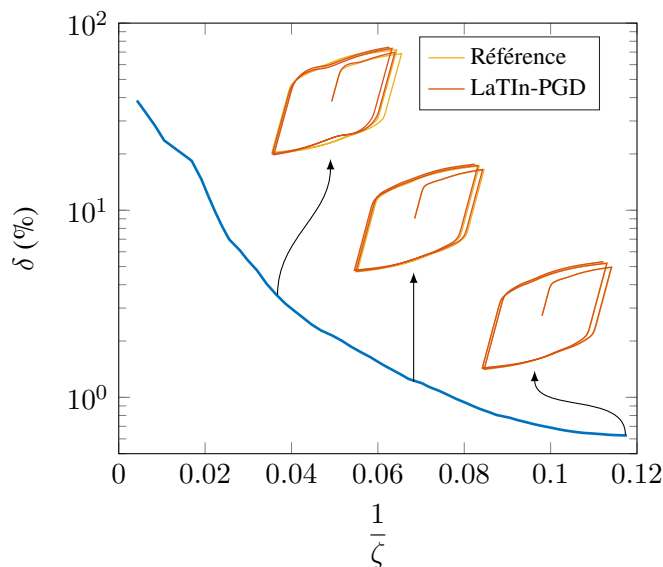


FIGURE 4 – Décroissance de l'erreur δ en fonction de l'inverse du gain $\frac{1}{\epsilon}$ avec illustration sur l'élément le plus chargé de la différence des cycles contrainte fonction de la déformation entre la référence et la méthode LaTin-PGD

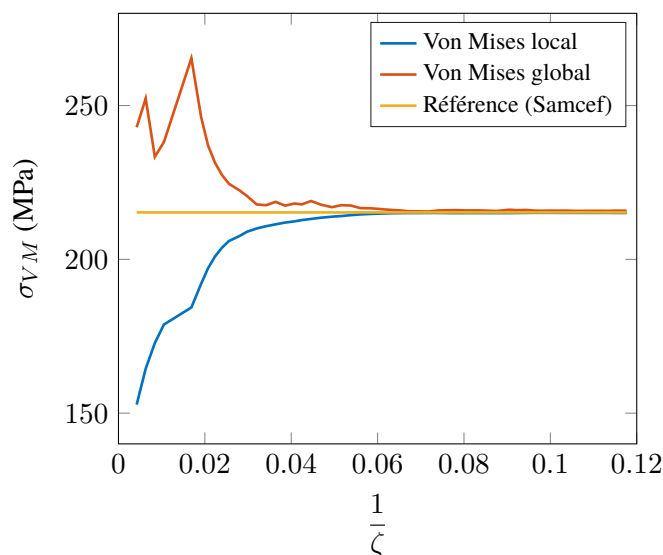


FIGURE 5 – Évolution de la contrainte de Von Mises en fonction de $\frac{1}{\epsilon}$ pour l'élément le plus chargé évaluée respectivement à l'issue de l'étape locale, de l'étape linéaire, et comparée à la valeur de référence

5 Remerciements

Nous remercions SIEMENS pour le financement de ces travaux ainsi que le CNRS et la région Île-de-France pour leur soutien au Mésocentre de calculs Fusion de Centrale-Supélec et de l'École normale supérieure Paris-Saclay.

Références

- [1] Boisse, P., Bussy, P., and Ladeveze, P. (1990), "A new approach in non-linear mechanics : the large time increment method," Int. J. Num. Meth. Engr. 29, 647-663.
- [2] Boucard, P.-A., Ladevèze, P., Poss, M., and Rougée, P. (1997). A non incremental approach for large displacement problems. Computers & structures, 64(1-4) : 499 ?508.
- [3] Chinesta, F. & Ladevèze, P. (2014). Separated Representations and PGD-Based Model Reduction. Fundamentals and Applications, International Centre for Mechanical Sciences, Courses and Lectures, 554.

- [4] Chinesta, F., Huerta, A., Rozza, G., and Willcox, K. (2016). Model order reduction : a survey. Wiley.
- [5] Cognard, J.-Y. (1990), "Le traitement des problèmes non linéaires à grand nombre de degrés de liberté par la méthode à grand incrément de temps," *Calcul des Structures et Intelligence Artificielle* (J. M. Fouet et al. eds), Pluralis, pp. 211-222.
- [6] Ladeveze, P. (1985a), "Nouveaux algorithmes : cadre mécanique et développements," Report No. 57, LMT Cachan.
- [7] Ladeveze, P. (1985b), "Sur une famille d'algorithmes en Mécanique des Structures," *Comptes Rend. Acad. Sci. Paris*, 300 II, 41-44.
- [8] Ladeveze, P. (1989a), "La méthode à grand incrément de temps (M.A.G.I.T.). Une nouvelle classe d'algorithmes pour les modèles de comportement à variables internes," Report No. 95, LMT Cachan.
- [9] Ladeveze, P. (1989b), "La méthode à grand incrément de temps pour l'analyse de structures à comportement non linéaire décrit par variables internes," *Comptes Rend. Acad. Sci. Paris* 309 11, 1095-1099.
- [10] Ladevèze, P. (1999). *Nonlinear computational structural mechanics : new approaches and non-incremental methods of calculation*. Springer Science & Business Media.
- [11] Ladevèze, P. (2016). Sur une nouvelle version non-intrusive de la méthode LATIN en quasi-statique. Rapport interne LMT.
- [12] Lemaitre, J., Chaboche, J.-L., Benallal, A., and Desmorat, R. (2009). *Mécanique des matériaux solides - 3ème édition*. Dunod.
- [13] Néron, D. & Ladevèze, P. (2010). Proper generalized decomposition for multi-scale and multi-physics problems. *Archives of Computational Methods in Engineering*, 17(4) : 351-372.
- [14] Relun, N., Néron, D. & Boucard, P. A. (2013). A model reduction technique bases on the PGD for elastic-viscoplastic computational analysis. *Computational Mechanics*, 51(1) : 83-92.