



HAL
open science

Modèles réduits de grandes dimensions compatibles avec les solveurs éléments finis

Charles Paillet, David Néron, Pierre Ladevèze

► To cite this version:

Charles Paillet, David Néron, Pierre Ladevèze. Modèles réduits de grandes dimensions compatibles avec les solveurs éléments finis. 14ème Colloque National en Calcul de Structures (CSMA 2019), CSMA, LEM3, MSME, Université de Lorraine, Arts et Métiers, CNRS, May 2019, Hyères, France. hal-04824568

HAL Id: hal-04824568

<https://hal.science/hal-04824568v1>

Submitted on 7 Dec 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.

Modèles réduits de grandes dimensions compatibles avec les solveurs éléments finis

C. Paillet*, P. Ladevèze, D. Néron

LMT (ENS Paris-Saclay, CNRS, Université Paris-Saclay), 61 avenue du Président Wilson, 94235 Cachan cedex, France,

* paillet@lmt.ens-cachan.fr

Résumé — La PGD multiéchelle en paramètres (PM-PGD) est une méthode de réduction de modèle capable de prendre en compte des centaines de paramètres. Cette méthode a été introduite accompagnée d'une discrétisation spatiale spécifique, mais discontinue. Une adaptation de cet algorithme compatible avec les logiciels industriels classiques tels que les solveurs éléments finis est ici présentée. La procédure est décrite et des exemples numériques associés à des problèmes de formes variés sont traités, incluant jusqu'à mille paramètres.

Mots clés — réduction de modèle, PGD, multiparamétrique

1 Introduction et représentation des données

Les techniques de réduction de modèle apportent une solution au besoin de simulations précises et rapides en ingénierie. Les méthodes majeures telles que la PGD (Proper Generalized Decomposition), la POD (Proper Orthogonal Decomposition) ou les Reduced Basis (RB) offrent des outils permettant de résoudre des problèmes variés et complexes en temps quasi-réel (voir [1] pour un aperçu de ces méthodes). Nos travaux sont basés sur la PGD [2, 3], technique de réduction qui associée au solveur LaTin permet de résoudre efficacement des problèmes mécaniques non linéaires. Elle a été appliquée avec succès dans un grand nombre de situations (voir [4] pour ses développements les plus récents) et est maintenant mature pour des cas industriels.

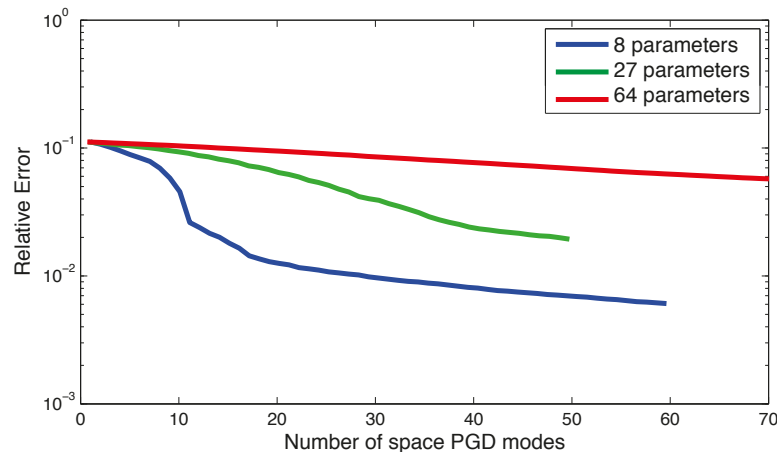


FIGURE 1 – Convergence d'un algorithme de PGD classique pour différents jeux de paramètres

Le calcul de modèles réduits de problèmes paramétriques est un puissant outil pour l'optimisation ou le temps réel. Les paramètres d'entrée peuvent être de natures diverses (matériaux, géométrie, conditions limites ...) et le modèle réduit, calculé hors ligne, donnera la solution du problème mécanique quels que soient les paramètres retenus. Ce vaste espace de solutions peut avoir un grand nombre de dimensions et même sa représentation ou son stockage ne sont pas triviaux, car ils se heurtent à la « malédiction de la dimensionnalité ». Des problèmes d'échelle industrielle avec une vingtaine de paramètres ont été résolus en utilisant la POD [5] ou la PGD [6], mais il s'agit de la limite en terme de nombre de paramètres

impliqués. La Figure 1 issue de [7] montre que pour plus de 20 paramètres, le taux de convergence d'une méthode PGD classique devient extrêmement lent. Au contraire, la Parameter-Multiscale PGD est une procédure dont la précision est quasi-indépendante du nombre de paramètres, ce qui permet de dépasser cette limite.

Plusieurs formats de données permettent de représenter des champs de grande dimension, comme ceux basés sur les formats tensoriels [8]. Cependant, nous avons choisis une représentation basée sur une propriété physique : le principe de Saint-Venant. Celui-ci nous permet de traiter des problèmes pour lesquels les paramètres sont très localisés spatialement. C'est le cas par exemple d'un cas-test dans lequel l'espace physique Ω_E est découpé en sous-domaine Ω_{E_i} , chaque paramètre μ_{E_i} étant associé au coefficient matériau d'un des Ω_{E_i} , tels ceux représentés Figure 2.

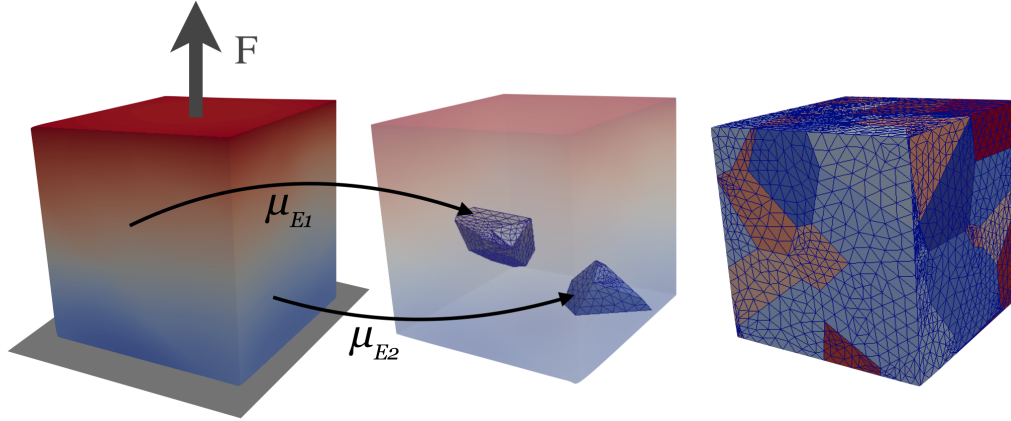


FIGURE 2 –

Cube encastré en traction : problème modèle ; une solution particulière ; mise en valeur de deux sous-domaines Ω_{E_1} et Ω_{E_2} ; maillage et décomposition du cube

Nous avons montré dans [7] qu'une perturbation locale sur un sous-domaine Ω_E possède deux niveaux d'influence distincts :

- un terme est potentiellement non linéaire et localisé dans un voisinage de Ω_E
- l'autre terme est global, mais linéaire.

Ainsi, nous avons introduit deux échelles comme on peut le faire classiquement en temps [9], l'une micro, l'autre macro, pour décrire l'espace paramétrique. Nous proposons d'approximer la solution locale associée au sous-domaine Ω_E par une formulation de la forme :

$$\underline{X}_E(\underline{\mu}) = \sum_{i=1}^m \tilde{X}_E^{(i)} \prod_{E'' \notin \bar{C}_E} \gamma_{E''}^{M(i)}(\mu_{E''}) \prod_{E' \in \bar{C}_E} \gamma_{E'E'}^{m(i)}(\mu_{E'}) \quad (1)$$

où m est le nombre de modes retenus, γ^M et γ^m sont respectivement les fonctions macro et micro et \tilde{X}_E la composante spatiale de cette partie de la solution localisée sur Ω_E . \bar{C}_E est un voisinage du sous-domaine Ω_E qui limite l'impact des fonctions micro du paramètre μ_E . Une représentation linéaire des fonctions macro γ^M permet de stocker intégralement grâce à seulement 2 variables.

L'utilisation de cette représentation peut s'avérer complexe car les champs locaux $\underline{X}_E(\underline{\mu})$ sont *a priori* discontinus d'un sous-domaine à l'autre. Cette difficulté peut être surmontée en utilisant une discrétisation spatiale discontinue, comme par exemple la Weak-Trefftz Discontinuous Galerkin method (WTDG), ce qui a été fait dans [10]. Cependant, ces travaux restaient limités en terme d'applications industrielles car basés sur un solveur développé spécifiquement. La méthode présentée ici permet de s'affranchir de cette limite : nous utiliserons toujours la représentation multiéchelle en paramètres (1), mais elle sera associée à une discrétisation spatiale continue et un algorithme adapté. Un code éléments finis générique peut être utilisé et de nombreuses géométries et répartitions des domaines paramétriques sont devenues accessibles.

2 Algorithme et justification théorique

Problème modèle On s'intéresse à un milieu élastique linéaire occupant un domaine $\Omega \subset \mathbb{R}^3$ avec des conditions limites de Dirichlet appliquées sur $\partial_1\Omega$ et de Neumann sur $\partial_2\Omega$:

$$\begin{cases} \underline{u} \in \mathcal{U} = [\mathcal{H}^1(\Omega)]^3 \\ \operatorname{div}(\underline{\sigma}) + \underline{f}_d = 0 & \text{sur } \Omega \\ \underline{u} = \underline{u}_d & \text{sur } \partial_1\Omega \\ \underline{\sigma}\underline{n} = \underline{F}_d & \text{sur } \partial_2\Omega \\ \underline{\sigma} = \mathbf{H}(\underline{\mu})\underline{\varepsilon} \\ \underline{\varepsilon} = \frac{1}{2}(\nabla\underline{u} + \nabla\underline{u}^T) \end{cases} \quad (2)$$

Le domaine Ω est divisé en sous-domaines $\Omega_E, E \in \mathbf{E}$. Chaque paramètre μ_E est associé à la rigidité d'un Ω_E ce qui permet de définir : $\underline{\mu} = \{\mu_E\}_{E \in \mathbf{E}}$. Ainsi, la relation de comportement choisie dans (2), la loi de Hooke \mathbf{H} , dépend des paramètres $\underline{\mu}$ et est donnée localement par $\mathbf{H}_E(\mu_E)$.

Après discrétisation spatiale, par la méthode des éléments finis par exemple, tous les champs spatiaux peuvent être approximés par N degrés de libertés et le problème à résoudre devient :

$$\begin{aligned} & \text{Trouver } \underline{X}(\underline{\mu}) \in \mathbf{X} \text{ où } \mathbf{X} : \begin{cases} \Sigma_{\mu} & \rightarrow \mathbf{U} = \mathbb{R}^N \\ \underline{\mu} & \rightarrow \underline{X}(\underline{\mu}) \end{cases} \text{ tel que :} \\ \forall \underline{\mu} \in \Sigma_{\mu} & \quad \mathbf{A}(\underline{\mu})\underline{X}(\underline{\mu}) = \underline{F} \end{aligned} \quad (3)$$

où \mathbf{A} est un opérateur linéaire défini positif dépendant de $\underline{\mu}$.

Compensation des erreurs La solution complète du problème (2) est un couple de champs $\varepsilon(\underline{\mu})$ cinématiquement admissible et $\sigma(\underline{\mu})$ statiquement admissible. La solution exacte (σ, ε) doit aussi respecter la relation de comportement : $\sigma = \mathbf{H}\varepsilon$. Considérons deux champs admissibles σ^a et ε^a , le respect de la loi de comportement $\mathcal{E}_{CR} = \sigma^a - \mathbf{H}\varepsilon^a$ est un bon indicateur de la précision de la solution approximée.

Dans notre cas, on va vérifier directement si nos champs élément finis respectent la loi de comportement. On considérera donc les champs discrétisés et issus de la résolution de (3) : $(\sigma^h, \varepsilon^h)$. Ces champs sont cinématiquement et statiquement admissibles au sens des éléments finis (voir [11]). De plus, une erreur au sens des éléments finis peut être définie par $\mathcal{E}_{CR}^h = \sigma^h - \mathbf{H}\varepsilon^h$. Pour un problème de statique classique, cette erreur est triviale et égale à zéro par construction, mais dans notre cas σ^h, ε^h et \mathbf{H} dépendent des paramètres. Cette erreur nous donne donc un bon indicateur de la précision de notre solution sur le domaine paramétrique complet.

\mathcal{E}_{CR}^h sera utilisée à chaque itération de l'algorithme et c'est elle qu'on va chercher à compenser le plus précisément possible à chaque itération. Notons $\mathbf{R} = \sigma - \mathbf{H}\varepsilon$ l'erreur à une itération donnée de l'algorithme. Le problème suivant permet de corriger cette erreur :

$$\begin{cases} \forall \underline{v}^* \in \mathcal{U}_0^{ad}, \quad \int_{\Omega} \operatorname{Tr}((\Delta\sigma - \mathbf{R})\Delta\varepsilon(\underline{v}^*)) d\Omega = 0 \\ \Delta\sigma = \mathbf{H}'\Delta\varepsilon(\underline{u}) \end{cases} \quad (4)$$

où \mathcal{U}_0^{ad} est l'espace des déplacements cinématiquement admissibles à 0 (CA-0) défini par $\mathcal{U}_0^{ad} = \{\underline{v} \in \mathcal{U} | \underline{v} = \underline{0} \text{ sur } \partial_1\Omega\}$ et \mathbf{H}' une approximation de la loi de Hooke. Ainsi, $\Delta\varepsilon$ est CA-0 et $\Delta\sigma$ est statiquement admissible à \mathbf{R} (SA- \mathbf{R}), donc $(\Delta\sigma - \mathbf{R})$ est SA-0. Ainsi, comme l'approximation initiale de notre solution est donnée par ε qui est CA et σ qui est SA, on peut définir une nouvelle solution admissible :

$$\begin{cases} \tilde{\varepsilon} = \varepsilon + \Delta\varepsilon \\ \tilde{\sigma} = \sigma + (\Delta\sigma - \mathbf{R}) \end{cases} \quad (5)$$

La nouvelle erreur associée à ce problème est donnée par : $\tilde{\mathcal{E}} = \tilde{\sigma} - \mathbf{H}\tilde{\varepsilon} = \Delta\sigma - \mathbf{H}\Delta\varepsilon$.

Si le problème est résolu en utilisant la loi de Hooke exacte $\mathbf{H}' = \mathbf{H}$, alors cette nouvelle solution est exacte. Cependant, la dépendance paramétrique de \mathbf{H} est souvent trop complexe et seule une approximation de la loi de comportement est utilisée, donnant une nouvelle erreur de la forme :

$$\tilde{\mathcal{E}} = (\mathbf{H}' - \mathbf{H})\Delta\varepsilon \quad (6)$$

Algorithme et implémentation Chaque paramètre μ_E est associé à un sous-domaine spatial, c'est pourquoi chaque variable sera décrite de manière indépendante sur chaque sous-domaine, en particulier l'erreur \mathbf{R} , car elle dépend de la relation de comportement exacte \mathbf{H} définie localement.

La procédure présentée dans le paragraphe précédent est entièrement basée sur la compensation des erreurs locales. Pour cela, l'algorithme doit être initialisé par une première approximation qui a été choisie comme étant la solution associée à la valeur moyenne des paramètres. Cette solution moyenne peut déjà être considérée comme une approximation et la norme de l'erreur associée permettra de mettre en évidence les corrections ultérieures (voir Table 1).

Après l'initialisation, on peut calculer une première erreur sur chaque sous-domaine : $\mathbf{R}_0 = \{\mathbf{R}_E, E \in \mathcal{E}\}$. La compensation globale de ces erreurs locales peut être réalisée simplement en corrigeant itérativement chacune des \mathbf{R}_E puis en sommant les contributions ainsi obtenues. Pour se faire, on résout le problème (4) pour chaque \mathbf{R}_E . La loi de comportement exacte, qui est de très grande dimension ne peut pas être utilisée. Une version simplifiée doit être choisie : sur le sous-domaine Ω_E dont on veut corriger l'erreur, on garde la loi de comportement exacte et sa dépendance paramétrique en μ_E . Partout ailleurs (domaine noté $\overline{\Omega}_E$), on se contente d'utiliser une valeur moyenne \mathbf{H}_0 qui ne dépend d'aucun paramètre. Ainsi, le problème (4) peut être réécrit :

$$\forall \mathbf{v}^* \in \mathcal{U}_0^{ad}, \quad \int_{\Omega_E} \text{Tr}((\mathbf{H}_E(\Delta\varepsilon) - \mathbf{R}_E)\Delta\varepsilon(\mathbf{v}^*)) d\Omega + \int_{\overline{\Omega}_E} \text{Tr}(\mathbf{H}_0(\Delta\varepsilon)\Delta\varepsilon(\mathbf{v}^*)) d\Omega = 0 \quad (7)$$

Cette formulation ne dépend que d'un seul paramètre, μ_E , les autres paramètres se factorisant aisément. Grâce au principe de Saint-Venant, nous savons que la solution de ce problème sera concentrée sur le sous-domaine Ω_E et son voisinage, d'où le choix de décrire précisément la solution dans cette zone via des fonctions micro. Après résolution de ce problème (7), l'erreur devient :

$$\begin{cases} \mathbf{R}_E^1 = 0 \text{ sur } \Omega_E \\ \mathbf{R}_{E'}^1 = \mathbf{R}_{E'}^0 + (\mathbf{H}_0 - \mathbf{H}_{E'})\Delta\varepsilon, E' \neq E \end{cases} \quad (8)$$

Naturellement, l'erreur \mathbf{R}_E^0 a été compensée intégralement car le champ correcteur respecte exactement la loi de comportement sur Ω_E . Mais ce problème ne corrige pas les erreurs ailleurs que sur Ω_E , au contraire, un nouveau terme lié à l'approximation de \mathbf{H} est créé.

Cette procédure s'appuie sur la formulation faible (7) qui peut être résolue avec un solveur générique de type éléments finis. Aucun solveur discontinu n'est nécessaire car, malgré la représentation multi-échelle potentiellement discontinue (1), toutes les corrections $\Delta\varepsilon$ apportées à la solution sont globales en espace et continues d'un sous-domaine à l'autre. C'est seulement à l'issue de cette opération globale que les discontinuités de la solution complète réapparaissent, quand on applique la représentation (1) à la correction $\Delta\varepsilon$ nouvellement calculée.

Une itération complète de l'algorithme est réalisée lorsque le problème (7) a été résolu pour chaque $E \in \mathcal{E}$. La nouvelle erreur est alors composée d'un bien plus grand nombre de termes qui peuvent à leur tour être compensés par la même procédure lors de l'itération suivante. La structure complète de l'algorithme est résumée Algorithme 1.

Algorithme 1 Principe de construction du modèle réduit

- 1: Initialiser : $X_0, \sigma_0, \varepsilon_0$
 - 2: Calculer $\mathbf{R}_0 = \sigma_0 - \mathbf{H}\varepsilon_0 = \{\mathbf{R}_E, E \in \mathcal{E}\}$
 - 3: **for** $i = 0$ to Iter **do** ▷ Iter, Nombre d'itérations globales
 - 4: **for** $E \in \mathcal{E}$ **do** ▷ Nombre de sous-domaines
 - 5: Calculer $\Delta\sigma, \Delta\varepsilon$, solution du Pb (7) ▷ Correction de \mathbf{R}_E
 - 6: $\tilde{\mathbf{R}}(E) = \Delta\sigma - \mathbf{H}\Delta\varepsilon = \{\tilde{\mathbf{R}}_{E'}(E), E' \in \mathcal{E}\}$
 - 7: **end for**
 - 8: $\mathbf{R}_i = \sum_{E \in \mathcal{E}} \tilde{\mathbf{R}}(E)$ ▷ Nouvelle erreur
 - 9: $\varepsilon_i = \varepsilon_{i-1} + \Delta\varepsilon$; $\sigma_i = \sigma_{i-1} + (\Delta\sigma - \mathbf{R}_{i-1})$ ▷ Incrément de la solution
 - 10: Vérification de la convergence
 - 11: **end for**
-

3 Résultats numériques

Indicateur d'erreur Comme nous l'avons montré précédemment, l'erreur $\tilde{\mathcal{E}} = \tilde{\sigma} - \mathbf{H}\tilde{\varepsilon}$ décrit le respect de la loi de comportement au sens des éléments finis. Cependant, cette erreur est définie continument par rapport aux paramètres. Comme l'espace paramétrique est numériquement discrétisé via l'approximation multiéchelle (1), on utilisera une norme énergétique :

$$|||\bullet|||^2 = \int_{\Sigma_\mu} \int_{\Omega} \mathbf{H}_0 \varepsilon(\bullet) \varepsilon(\bullet) d\Omega d\mu \quad (9)$$

qui permet de calculer un indicateur d'erreur scalaire :

$$\mathcal{E} = \frac{|||\tilde{\mathcal{E}}|||}{|||\sigma_0|||} \quad (10)$$

où σ_0 représente la contrainte moyenne dans le domaine, calculée lors de l'initialisation de l'algorithme.

Cet estimateur a cependant une limite lorsqu'il est calculé numériquement : comme tous les champs sont exprimés sous leur forme multiéchelle (1), il ne permet pas de prendre en compte l'erreur liée à cette approximation, ce qui introduit un biais. On calculera donc en plus une « erreur locale » à partir d'un jeu donné de paramètres :

$$\mathcal{E}_{part} = \frac{||\mathbf{X}_{exact} - \mathbf{X}_{PM-PGD}||}{||\mathbf{X}_{exact}||} \quad (11)$$

où $||\bullet||$ est une simple norme énergétique sans intégration paramétrique : $||\bullet||^2 = \bullet^T \mathbf{A}^{0^{-1}} \bullet$. Cette norme locale est calculée à partir d'un grand nombre de tirages aléatoires de paramètres (50 ici) et seule l'erreur maximale est conservée. Cet indicateur très peu couteux peut donc être calculé pour les problèmes de très grande taille, il est cependant dépendant d'un processus aléatoire.

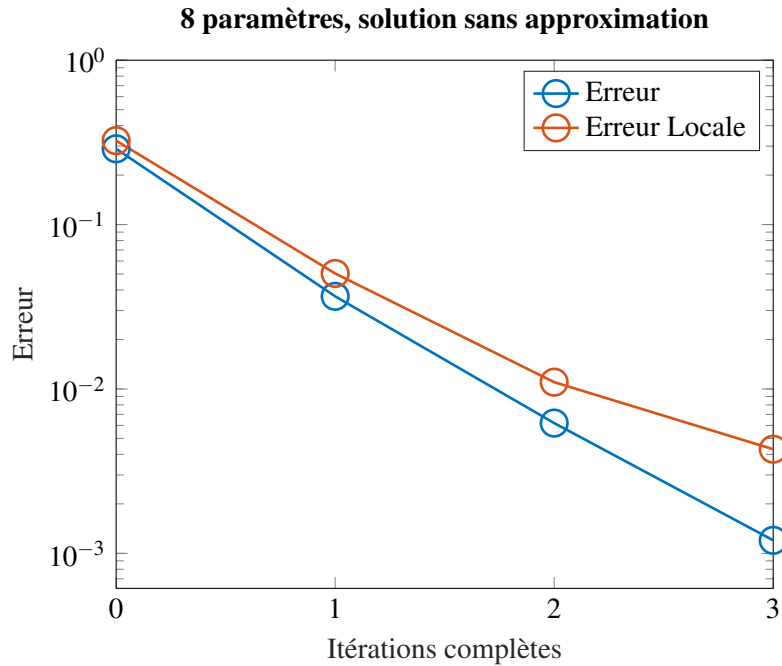


FIGURE 3 –
Erreur et erreur locale pour un problème à 8 paramètres utilisant un algorithme « parfait » sans approximation

Tests numériques En choisissant seulement des fonctions micro dans la formulation (1) l'erreur devrait converger vers zéro au sens des éléments finis, car la compensation de l'erreur est alors complète à

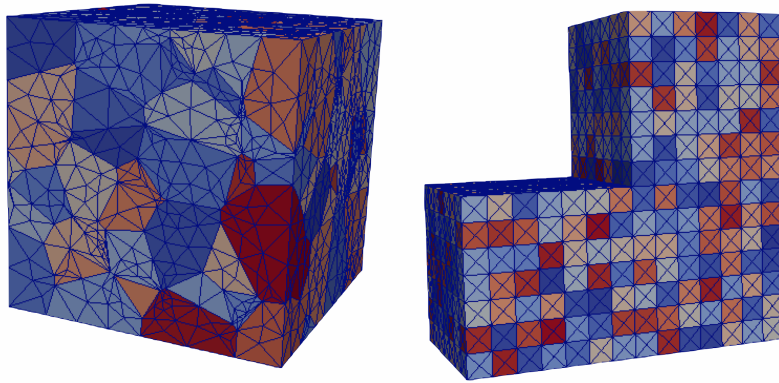


FIGURE 4 –

Sous-domaines paramétriques pour deux cas de grande dimension : décomposition polyédrique d'un cube (300 paramètres), décomposition régulière d'une structure en L (1029 paramètres)

	Erreur locale, solution moyenne	Erreur locale, première itération
Cube, sous-domaines réguliers, 1000 paramètres	27,8%	3,9%
Cube, sous-domaines polyédriques, 300 paramètres	19,5%	3,4%
Structure en L, sous-domaines réguliers, 1029 paramètres	18,0%	4,5%
Structure en L, sous-domaines polyédriques, 120 paramètres	25,9%	4,8%

TABLE 1 –

Erreur locale après la première itération de l'algorithme pour différents problèmes, variation paramétrique de 50% autour de sa valeur moyenne.

chaque itération. C'est ce qui a été fait pour un cas à 8 paramètres sur le problème représenté Figure 2 où chaque sous-domaine est un cube de rigidité indépendante.

La Figure 4 montre deux décompositions différentes en sous-domaines pour deux cas-tests de géométries différentes. Nous avons montré dans [10] que la PM-PGD est quasi-indépendante du nombre de paramètres sur un cas-test donné (ce résultat est même exacte si la dimension spatiale vaut 1, voir [7]). Nous pouvons voir sur ces deux exemples que la procédure décrite précédemment donne des résultats du même ordre de grandeur. Seule la première itération a été calculée pour ces différents cas-tests. En effet, la deuxième est extrêmement coûteuse pour les cas à plus de 100 paramètres et des approximations supplémentaires doivent être introduites, ce qui réduit l'efficacité de la correction.

Conclusion

Cette version continue de la PGD multiéchelle en paramètres peut résoudre une vaste gamme de problèmes paramétriques avec des centaines de paramètres. La possibilité d'utiliser des solveurs éléments finis classiques est un de ses principaux avantages, et des décompositions discontinues spécifiques n'ont pas besoin d'être utilisées. Cependant, cette méthode reste intrusive et ne peut pas se contenter d'utiliser un solveur en « boîte noire » car le problème doit être résolu. Seuls des problèmes à paramètres matériels ont été traités dans des cas linéaires. La PM-PGD ne peut pas s'étendre aisément à des problèmes non linéaires, mais son adaptation à des paramètres géométriques ou liés aux conditions limites par exemple peut être implémentée facilement. De même, des paramètres plus globaux spatialement peuvent toujours être pris en compte, à condition de les décrire comme des combinaisons de variables locales.

Comme elle permet de manier des espaces paramétriques de très grande dimension, la PM-PGD est une technique très prometteuse pour les problèmes à plus d'une douzaine de paramètres. Pour les cas-tests réalisés, le nombre de degrés de liberté de la discrétisation spatiale pouvait dépasser 10^5 , pour un

problème polycristallin à 60 paramètres par exemple. Des utilisations pratiques des modèles réduits ainsi construits sont à l'étude pour des procédures itératives ou des applications d'optimisation.

Références

- [1] Francisco Chinesta and Pierre Ladevèze, editors. *Separated Representations and PGD-Based Model Reduction : Fundamentals and Applications*, volume CISM 554. Springer, springer edition, 2014.
- [2] Pierre Ladevèze. On Algorithm Family in Structural Mechanics. *Comptes rendus des séances de l'Académie des sciences. Série 2*, 300(2), 1985.
- [3] Pierre Ladevèze. The large time increment method for the analyse of structures with nonlinear constitutive relation described by internal variables (in french). *Comptes rendus des séances de l'Académie des sciences. Série 2*, 309(2) :1095–1099, 1989.
- [4] Pierre Ladevèze. On reduced models in nonlinear solid mechanics. *European Journal of Mechanics, A/Solids*, 60 :227–237, 2016.
- [5] Tan Bui-Thanh, Karen Willcox, and Omar Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6) :3270–3288, 2008.
- [6] Basile Marchand, Ludovic Chamoin, and Christian Rey. Real-time updating of structural mechanics models using kalman filtering, modified constitutive relation error, and proper generalized decomposition. *International Journal for Numerical Methods in Engineering*, 107(9) :786–810, 2016.
- [7] Pierre Ladevèze, Charles Paillet, and David Néron. Extended-PGD model reduction for nonlinear solid mechanics problems involving many parameters. *Computational Methods in Applied Sciences*, 46 :201–220, 2018.
- [8] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, series in edition, 2012.
- [9] Pierre Ladevèze, Jean-Charles Passieux, and David Néron. The LATIN multiscale computational method and the Proper Generalized Decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199 :1287–1296, 2010.
- [10] Charles Paillet, David Néron, and Pierre Ladevèze. A door to model reduction in high-dimensional parameter space. *Comptes Rendus Mécanique*, 346(7) :524 – 531, 2018. Model reduction, data-based and advanced discretization in computational mechanics / Réduction de modèles, données et techniques de discrétisation avancées en mécanique numérique.
- [11] Charles Paillet, David Néron, and Pierre Ladevèze. High-dimensional multiparametric reduced order models compatible with finite element solvers. *In preparation*, 2019.