



Technique de compression de données pour la réduction de modèle par PGD

Matteo Capaldo, Pierre-Alain Guidault, Pierre Ladevèze, David Néron

► **To cite this version:**

Matteo Capaldo, Pierre-Alain Guidault, Pierre Ladevèze, David Néron. Technique de compression de données pour la réduction de modèle par PGD. 11e colloque national en calcul des structures, May 2013, Giens, France. 11e colloque national en calcul des structures, 2013. <hal-01717028>

HAL Id: hal-01717028

<https://hal.archives-ouvertes.fr/hal-01717028>

Submitted on 25 Feb 2018

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.

Public Domain

Technique de compression de données pour la réduction de modèle par PGD

Matteo CAPALDO, Pierre-Alain GUIDAULT, Pierre LADEVÈZE, David NÉRON

LMT-ENS Cachan/CNRS/UPMC/PRES UniverSud Paris
{capaldo, guidault, ladeveze, neron} @lmt.ens-cachan.fr

Résumé — Ce travail porte sur l’implémentation de la *Proper Generalized Decomposition* (PGD) dans le cadre de la méthode LATIN pour résoudre des problèmes d’évolution paramétrés, éventuellement non linéaires. La PGD est une méthode de réduction de modèle où la base réduite n’est pas donnée *a priori*. La représentation de la solution sur cette base réduite générée automatiquement permet de réduire considérablement les coûts de calcul. L’algorithme utilisé dans ce travail est la méthode LATIN. Il s’agit d’une méthode itérative non incrémentale qui génère, à chaque itération, une approximation de la solution sur tout le domaine espace-temps-paramètres par enrichissements successifs. Les opérations sur ces représentations en variables séparées peuvent s’avérer être la partie de la méthode la plus coûteuse en temps de calcul. Dans ce travail, nous proposons une technique de compression de données permettant de simplifier et de diminuer le coût des opérations algébriques élémentaires.

Mots clés — PGD, LATIN, réduction de modèle, compression de données

1 Introduction

La simulation numérique prend aujourd’hui une place importante dans de nombreuses branches de l’ingénierie. En raison de la complexité croissante des modèles et de l’augmentation du nombre de degrés de liberté, les techniques de résolution classiques conduisent à des coûts de calcul parfois très élevés et, pour de nombreuses applications, des méthodologies innovantes et alternatives sont nécessaires. Le concept de réduction de modèle est une des voies possibles pour résoudre ces problèmes en exploitant le fait que la réponse de modèles complexes (ou d’une famille de modèles) peut souvent être estimée avec une précision satisfaisante par un modèle réduit, qui est obtenu par projection de ces modèles dans une base de dimension de plusieurs ordres de grandeur inférieure à celle de l’espace dans lequel était cherché la solution. Les méthodes de réduction de modèle basées sur la représentation en variables séparées reçoivent un intérêt croissant dans divers domaines du calcul scientifique [1, 2].

Ce travail met l’accent sur la *Proper Generalized Decomposition* (PGD), qui consiste à construire progressivement une approximation à variables séparées de la solution sans imposer une base de représentation au préalable [1, 2]. Elle est utilisée ici pour résoudre les problèmes d’évolution paramétrés et éventuellement non linéaires. La PGD a été appliquée pour résoudre de nombreux types de problèmes dans le cadre de la méthode LATIN et a permis de fortement diminuer les coûts de calcul [3].

Les performances des techniques de réduction de modèles sont d’autant plus élevées que le modèle réduit n’est construit qu’une seule fois, mais ce n’est malheureusement pas possible lorsque le problème à résoudre est non linéaire et que les opérateurs qui interviennent sont mis à jour au cours des itérations de l’algorithme [4]. Pour rendre la manipulation des champs PGD plus efficace, un nouveau cadre algébrique basé sur les notions d’instant, de points et de paramètres *de référence* est proposé ici. Il permet de définir une version *compressée* des données. L’espace des variables compressées montre des propriétés très intéressantes en termes de réalisation des opérations algébriques standards, ce qui devrait apporter des améliorations substantielles en terme de performances de calcul.

2 Description du problème

Dans cette section, une brève description des aspects principaux de la stratégie de calcul est présentée. Plus de détails sont disponibles dans [1, 3].

2.1 Problème de référence et sous structuration

Pour des raisons de simplicité, on considère l'évolution quasi-statique et isotherme d'une structure viscoélastique définie sur le domaine espace-temps $[0, T] \times \Omega$, dans l'hypothèse des petites perturbations (pour un éventail plus large de comportements voir [3]). On suppose que le problème ne dépend pas de paramètres pour cette présentation. La structure considérée est soumise à une force volumique \underline{f}_d , à une force de surface sur la partie $\partial_2\Omega$ de la frontière, et à un déplacement imposé \underline{U}_d sur la partie complémentaire, $\partial_1\Omega$ de la frontière. On note ε la déformation associée au champ du déplacement \underline{U} et σ le tenseur des contraintes. La déformation est divisée en une partie élastique ε_e qui vérifie la loi d'état $\sigma = \mathbf{K}\varepsilon_e$ (\mathbf{K} est l'opérateur de Hooke), et une partie inélastique ε_p liée à la contrainte par la loi d'évolution $\varepsilon_p = \mathbf{B}(\sigma)$ qui est éventuellement non linéaire. Les trois champs, déplacement, déformation et contrainte vérifient des conditions initiales à $t = 0$. Par simplicité, on considèrera nulles les conditions initiales. La structure est considérée comme un assemblage d'entités élémentaires : sous-structures et interfaces. En ce qui concerne le temps, l'intervalle $I = [0, T]$ est divisé en sous-intervalles $I_i^C = [t_i^C, t_{i+1}^C]$. On note \square_E une quantité relative à la sous-structure Ω_E . L'interface entre les sous-structures Ω_E et $\Omega_{E'}$ est désignée par $\Phi_{EE'}$. Chaque sous-structure et interface a ses propres variables et équations (admissibilité cinématique, statique, relation de comportement). L'état de chaque sous-structure est entièrement défini par $\dot{\varepsilon}_{pE}$ et σ_E . L'état de l'interface $\Phi_{EE'}$ est donné par \underline{W}_E et $\underline{W}_{E'}$, la restriction des déplacements \underline{U}_E et $\underline{U}_{E'}$ sur $\Phi_{EE'}$, ainsi que par les forces \underline{F}_E et $\underline{F}_{E'}$, admissibles aux vecteurs contraintes sur $\Phi_{EE'}$.

2.2 Conditions d'admissibilité pour la sous-structure Ω_E

On introduit les espaces suivants avec leurs espaces vectoriels associés (notés \square^*) :

- l'espace \mathcal{E}_E des champs cinématiquement admissibles ($\dot{\varepsilon}_E, \dot{\underline{W}}_E$) :

$$(\dot{\varepsilon}_E, \dot{\underline{W}}_E) \in \mathcal{E}_E \iff \exists \underline{U}_E, \quad \underline{U}_E|_{\partial\Omega_E} = \underline{W}_E, \quad \varepsilon = \nabla_{sym}\underline{U}_E, \quad (1)$$

- l'espace \mathcal{F}_E des champs statiquement admissibles ($\sigma_E, \underline{F}_E$) : $(\sigma_E, \underline{F}_E) \in \mathcal{F}_E \iff$

$$\forall (\dot{\varepsilon}_E^*, \dot{\underline{W}}_E^*) \in \mathcal{E}_E, \quad \int_{I_i \times \Omega_E} \sigma_E : \dot{\varepsilon}_E^* d\Omega dt = \int_{I_i \times \Omega_E} \underline{f}_d \cdot \dot{\underline{U}}_E^* d\Omega dt + \int_{I_i \times \partial\Omega_E} \underline{F}_E \cdot \dot{\underline{W}}_E^* dS dt, \quad (2)$$

- l'espace \mathbf{Ad}_E des variables « E -admissibles » $s_E = (\dot{\varepsilon}_{pE}, \dot{\underline{W}}_E, \sigma_E, \underline{F}_E) \in \mathbf{Ad}_E \iff (\mathbf{K}^{-1}\dot{\sigma}_E + \dot{\varepsilon}_E, \dot{\underline{W}}_E) \in \mathcal{E}_E$ et $(\sigma_E, \underline{F}_E) \in \mathcal{F}_E$.

La loi de comportement d'une interface $\mathbf{b}_{EE'}$ entre les quantités qui décrivent son état est écrite formellement sous la forme :

$$\mathbf{b}_{EE'}(\dot{\underline{W}}_{EE'}, \dot{\underline{W}}_{E'E}, \dot{\underline{F}}_{EE'}, \dot{\underline{F}}_{E'E}) = 0 \quad (3)$$

2.3 Approche multi-échelle

L'approche multi-échelle consiste à introduire une description à deux échelles des inconnues d'interface [6] : l'échelle "macro" (\square^M) et le complément "micro" (\square^m). Les parties macro sont les valeurs moyennes en espace et en temps des inconnues. Pour $\dot{\underline{W}}_E$, la partie macro $\dot{\underline{W}}_E^M$ et sa partie complémentaire $\dot{\underline{W}}_E^m$ sont définies par :

$$\forall \underline{F}^{M*}, \quad \int_{I_i \times \delta\Omega_E} (\dot{\underline{W}}_E^M - \dot{\underline{W}}_E) \cdot \underline{F}^{M*} dS dt = 0 \quad \text{et} \quad \dot{\underline{W}}_E^m = \dot{\underline{W}}_E - \dot{\underline{W}}_E^M \quad (4)$$

Les espaces \mathcal{F}_E^M et \mathcal{W}_E^M sont choisis comme la partie linéaire des forces et déplacements en espace et quadratique par morceaux en temps [6]. La partie importante de la stratégie multi-échelle est le choix de la condition d'admissibilité statique sur les quantités macroscopiques. On impose à l'ensemble des efforts macro $\underline{F}^M = (\underline{F}_E^M)_{\Omega_E \subset \Omega}$ de vérifier systématiquement les équations d'équilibre au niveau des interfaces. L'espace correspondant est noté par \mathcal{F}_{ad}^M .

3 Méthode LATIN multi-échelle

Le moteur de la stratégie est la méthode LATIN. C'est une méthode itérative pour résoudre des problèmes non linéaires et dépendants du temps qui permet, à chaque itération, de générer une approximation de la solution sur le domaine espace-temps. On introduit les espaces suivants :

1. l'espace \mathbf{A}_d des solutions des équations linéaires et globales. $\mathbf{s} = (s_E)_{\Omega_E \subset \Omega} \in \mathbf{A}_d \iff \forall \Omega_E \subset \Omega, \mathbf{s}_E \in \mathbf{A}_{dE}$ (E -admissibilité) et $\underline{\mathbf{F}} \in \mathcal{F}_{ad}^M$ (admissibilité des efforts macro) ;
2. l'espace Γ des équations locales qui éventuellement peuvent être non linéaires. $\hat{\mathbf{s}} = (s_E)_{\Omega_E \subset \Omega} \in \Gamma \iff \forall \Omega_E \subset \Omega, \dot{\mathbf{x}}_{PE} = \mathbf{B}(\sigma_E)$ (loi d'évolution) et $\forall \Phi_{EE'}, \mathbf{b}_{EE'}(\dot{\mathbf{W}}_{EE'}, \dot{\mathbf{W}}_{E'E}, \dot{\mathbf{F}}_{EE'}, \dot{\mathbf{F}}_{E'E}) = 0$.

Les conditions initiales et la loi d'état sont supposées vérifiées dans les deux espaces \mathbf{A}_d et Γ . La solution exacte est définie par : $s_{exact} \in \mathbf{A}_d \cap \Gamma$.

On utilise un schéma itératif en deux étapes (*étape locale* et *étape linéaire*) qui construit alternativement un membre de l'espace Γ , puis un de \mathbf{A}_d par l'introduction de deux directions de recherche \mathbf{E}^+ et \mathbf{E}^- . On exploite le fait que la méthode génère une approximation sur tout l'espace-temps pour introduire une approximation appropriée des inconnues. En particulier, on utilise la *PGD* pour résoudre le problème à l'étape linéaire. Pour plus des détails sur la LATIN multi-échelle, on peut se reporter à [3, 6].

4 Proper Generalized Decomposition

L'approximation *PGD*, introduite par [7] et sous le nom « approximation radiale » en temps-espace permet de réduire de façon drastique le coût de calcul de l'étape linéaire de la LATIN [6]. La *PGD* consiste à approximer une fonction f définie sur le domaine espace-temps $I \times \Omega$ par la somme de produits de fonctions de variables séparées : une fonction du temps $\lambda_i(t)$ et une fonction de l'espace $\Lambda_i(\underline{\mathbf{M}})$: $\forall (t, \underline{\mathbf{M}}) \in I \times \Omega$,

$$f(t, \underline{\mathbf{M}}) \approx f_m(t, \underline{\mathbf{M}}) = \sum_{i=1}^m \lambda_i(t) \Lambda_i(\underline{\mathbf{M}}) \quad (5)$$

où les produits $\lambda_i(t) \Lambda_i(\underline{\mathbf{M}})$ sont à déterminer. La *PGD* peut être prolongée aussi à l'espace des paramètres, en ajoutant une nouvelle coordonnée dans la représentation séparée. Par exemple :

$$f(t, \underline{\mathbf{M}}, \gamma) \approx f_m(t, \underline{\mathbf{M}}, \gamma) = \sum_{i=1}^m \lambda_i(t) \Lambda_i(\underline{\mathbf{M}}) \Gamma_i(\gamma) \quad (6)$$

4.1 Algorithme de construction de la décomposition PGD

Lorsqu'on est à l'itération $n+1$ de la LATIN, une base $\{\Lambda_i(\underline{\mathbf{M}})\}$ provenant de l'itération précédente est disponible. La résolution du problème consiste en deux étapes :

- *étape 1 : réutilisation de la base réduite.* Cette étape consiste à construire une solution du problème en utilisant la base réduite déjà générée. On cherche pour cela la meilleure combinaison de cette base qui minimise une fonctionnelle associée à la vérification de la direction de recherche. Les seules inconnues sont les fonctions du temps λ_i . Ces fonctions du temps vérifient un système d'équations différentielles linéaires dont la résolution est peu coûteuse. Un indicateur d'erreur $E_{CR,E}$ est introduit pour quantifier la précision de cette première prédiction. Si $E_{CR,E}$ est inférieur à une valeur critique, l'étape linéaire à l'itération $n+1$ est considérée terminée, sinon on procède à l'étape 2.
- *étape 2 : enrichissement de la base réduite.* La prédiction de l'étape 1 est connue et une nouvelle approximation est cherchée. Pour cela, un nouveau couple de fonctions $\lambda(t)\Lambda(\underline{\mathbf{M}})$ est calculé en minimisant la fonctionnelle associée à la vérification de la direction de recherche. La génération de la fonction de l'espace $\Lambda(\underline{\mathbf{M}})$ est la partie coûteuse de l'algorithme.

Notons qu'il est possible de partager la base réduite entre toutes les sous-structures (quand elles sont similaires), ce qui permet de limiter encore les phases d'enrichissement nécessaire à l'amélioration de la solution.

4.2 Exemple numérique

On considère une structure homogène en matériau visco-élastique (modèle Kelvin-Voigt : $E = 60 \text{ GPa}$, $\nu = 0.33$, $\eta = 10 \text{ s}$). La **Fig. 1** représente la contrainte de Mises dans la structure et la courbe représentant l'erreur vis-à-vis d'une solution de référence en fonction des itérations de la méthode.

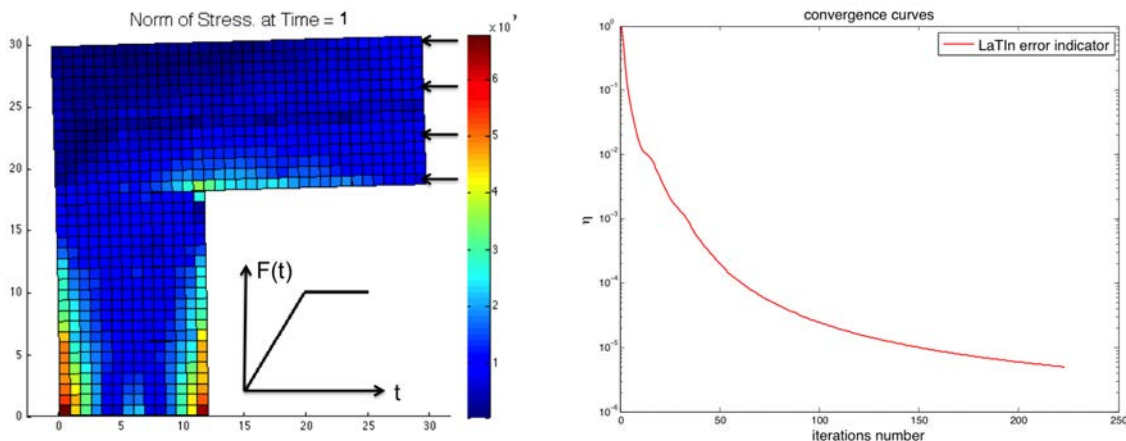


Fig. 1 – Simulation numérique pour une structure visco-élastique

5 Compression de données

Pour simplifier la présentation, on se limitera, dans cet article au développement de la méthode proposée dans le cas de champ dépendants du temps et de l'espace. La démarche peut être prolongée aisément à l'espace des paramètres.

5.1 Représentation dans un cadre à variables séparées

Dans les algorithmes développés dans nos travaux précédents, la PGD était utilisée pour représenter uniquement certaines des quantités dont la résolution nécessitait la résolution de problèmes globaux en espace et définis sur tout l'intervalle de temps. Ces quantités écrites sous forme de variables séparées cohabitaient avec des fonctions décrites sous une forme classique (dite "*full*" dans la suite, i.e. champ spatial pour chaque pas de temps). Pour tirer parti des avantages de la représentation PGD en terme de stockage et de calcul (notamment pour le calcul d'intégrales définies sur l'espace-temps, dans lesquelles la PGD permet de séparer l'intégration sur le temps de l'intégration sur l'espace), on se propose maintenant de généraliser la représentation PGD à l'ensemble des champs du problème. L'objectif est ici de simplifier le calcul des différentes opérations algébriques usuelles (produit, addition, intégration...) sur des champs définis sous forme PGD et d'éviter l'augmentation artificielle du nombre de modes PGD nécessaires à la représentation d'une fonction.

Pour fixer les idées, revenons à l'exemple de la **Fig. 1**. La base réduite nécessaire pour représenter la solution avec une précision suffisante a une dimension de l'ordre de 20. Supposons que nous soyons amenés à effectuer des opérations entre deux fonctions f et f' , représentées sous forme PGD. Celles-ci s'écrivent formellement :

$$f(t, \underline{M}) = \sum_{i=1}^m \lambda_i(t) \Lambda_i(\underline{M}) \quad \text{et} \quad f'(t, \underline{M}) = \sum_{i=1}^{m'} \theta_i(t) \Theta_i(\underline{M}) \quad (7)$$

avec m et m' de l'ordre de 20. On souhaite obtenir le produit de ces deux fonctions sous forme de variables séparées afin de rester dans le cadre PGD :

$$F(t, \underline{M}) = f(t, \underline{M}) f'(t, \underline{M}) = \sum_{i=1}^M \alpha_i(t) \beta_i(\underline{M}) \quad (8)$$

Une première solution est de calculer les produits termes à termes, ce qui conduit à une fonction s'écrivant comme la somme de $M = m \times m'$ produits avec M de l'ordre de 400. Ceci est évidemment beaucoup plus que le nombre nécessaire pour représenter celle-ci et le nombre de couples à manipuler va augmenter extrêmement rapidement avec les différentes opérations qui vont être réalisées.

Une deuxième technique est de revenir aux formes *full* des fonctions f et f' , de calculer le produit F , puis de réaliser une SVD tronquée au niveau de précision souhaité. Cette démarche est elle aussi très coûteuse puisqu'elle a recourt à une SVD sur une fonction qui est éventuellement représentée dans un espace de grandes dimensions.

La technique qui est proposée ici consiste à définir un cadre de représentation qui permet de construire pour un coût modique une première approximation à variables séparées de la fonction qui résulte de l'opération algébrique qu'on vient de réaliser. Pour cela, on introduit les notions d'*instants de référence* et de *points de référence* [5] (cf. **Fig. 2**). On découpe l'intervalle de temps en m_1 sous-intervalles I_i de taille Δt_i . Les centres t_i des sous-intervalles I_i sont appelés instants de référence. Pour l'espace, on introduit m_2 points \underline{M}_j et on divise Ω en sous-domaines Ω_j . Ces points sont appelés points de référence et les tailles des *patches* sont notées ω_j .

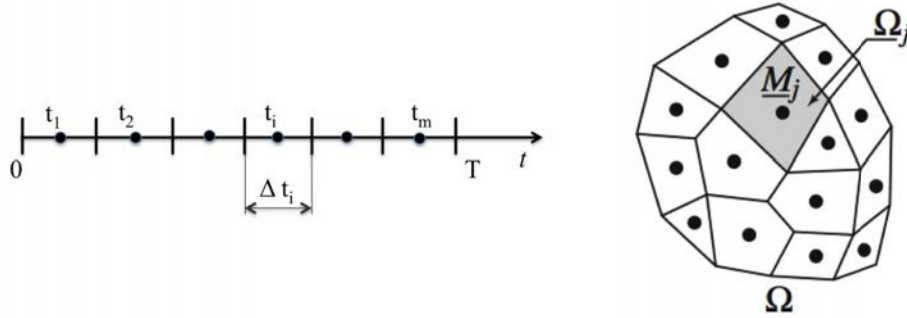


Fig. 2 – Instants et points de référence

Une champ f défini sur le domaine $I \times \Omega$ est *compressé* sous la forme \bar{f} à partir de ce que nous appelons « ses composantes généralisées ». $\bar{f} = \{(\bar{a}_{ij}, \bar{b}_{ij})\}$ définie comme suit :

$$\bar{f} := \left\{ \begin{array}{l} \bar{a}_{ij}(t) = f(t, \underline{M}_j) \quad \text{si } t \in I_i \\ \bar{a}_{ij}(t) = 0 \quad \quad \quad \text{sinon} \end{array} \middle| \begin{array}{l} \bar{b}_{ij}(\underline{M}) = f(t_i, \underline{M}) \quad \text{si } \underline{M} \in \Omega_j \\ \bar{b}_{ij}(\underline{M}) = 0 \quad \quad \quad \text{sinon} \end{array} \right\}_{i=1, \dots, m_1; j=1, \dots, m_2} \quad (9)$$

5.2 Algèbre élémentaire dans le cadre compressé

L'opération de compression des champs revêt un certain nombre de propriétés intéressantes pour réécrire les opérations algébriques classiques. Le **Tab. 1** montre quelques-unes de ces opérations sur deux champs f et f' .

| | |
|----------------|--|
| Addition | $\overline{f + f'} = \bar{f} + \bar{f}'$ |
| Multiplication | $\overline{f f'} = \bar{f} \bar{f}'$ |
| Dérivation | $\overline{\partial f / \partial t} = \partial \bar{f} / \partial t$ |
| Opérateur | $\overline{\mathbf{H} f} = \mathbf{H} \bar{f}$ |

Tableau 1 – Quelques opérations élémentaires dans le cadre compressé

Détaillons par exemple l'opération de multiplication, déjà vue dans l'équation (8), entre f et f' définies sous forme PGD. On procède de la manière suivante :

1. On calcule la version compressée des fonctions f et f' comme définie par (9), $\bar{f} = \{(\bar{a}_{ij}, \bar{b}_{ij})\}$ et $\bar{f}' = \{(\bar{a}'_{ij}, \bar{b}'_{ij})\}$.
2. On calcule la version compressée $\bar{F} = \{(\bar{A}_{ij}, \bar{B}_{ij})\}$ du produit $F = ff'$. Pour cela, on utilise le fait que $\bar{F} = \bar{f}\bar{f}'$, c'est-à-dire on écrit :

$$\bar{A}_{ij} = \bar{a}_{ij}\bar{a}'_{ij} \quad \text{et} \quad \bar{B}_{ij} = \bar{b}_{ij}\bar{b}'_{ij} \quad (10)$$

3. On cherche ensuite à reconstruire F avec une représentation en variables séparées à partir de sa représentation \bar{F} .
4. Si la qualité de la reconstruction obtenue n'est pas suffisante, on enrichit la représentation avec de nouveaux couples PGD.

5.3 Reconstruction à partir de la version compressée

On construit une première approximation à variables séparées F_0 , ne comportant qu'un seul couple par patch temps-espace $I_i \times \Omega_j$. Pour cela, on écrit :

$$F(t, \underline{M}) \approx F_0(t, \underline{M}) = \alpha_{ij}(t)\beta_{ij}(\underline{M}) \quad \forall (t, \underline{M}) \in I_i \times \Omega_j \quad (11)$$

où l'ensemble des $(\alpha_{ij}, \beta_{ij})$ sont définis à partir de $(\bar{A}_{ij}, \bar{B}_{ij})$ comme le résultat d'un problème de minimisation. En pratique, pour les problèmes de mécanique des structures considérés ici, la reconstruction choisie donne plus d'importance à la variable d'espace car les variations spatiales sont plus fortes les variations temporelles. En introduisant les normes suivantes :

$$\|f\|_{I_i}^2 = \int_{I_i} f^2 dt \quad \text{et} \quad \|f\|_{\Omega_j}^2 = \int_{\Omega_j} f^2 d\Omega \quad (12)$$

on définit la fonctionnelle à minimiser pour construire $(\alpha_{ij}, \beta_{ij})$:

$$J(\alpha_{ij}, \beta_{ij}) = \sum_{k=1}^{m_2} [\omega_k \|(\bar{A}_{ik}(t) - \alpha_{ik}(t)\beta_{ik}(\underline{M}_k))\|_{I_i}^2 + \Delta t_i \|\bar{B}_{ik}(\underline{M}) - \alpha_{ik}(t_i)\beta_{ik}(\underline{M})\|_{\Omega_k}^2] \quad (13)$$

où λ_k est une fonction d'influence permettant de donner plus d'importance aux patches voisins du patch considéré ($\lambda_k = 1$ si $k = j$ et décroissant lorsque $|k - j|$ augmente). Cette minimisation conduit aux formules explicites suivantes : $\forall (t, \underline{M}) \in I_i \times \Omega_j$,

$$\alpha_{ij}(t) = \frac{\sum_{k=1}^{m_2} \omega_k \bar{A}_{ik}(t) \bar{B}_{ik}(\underline{M}_k) \lambda_k}{\sum_{k=1}^{m_2} \omega_k \bar{B}_{ik}(\underline{M}_k) \bar{B}_{ik}(\underline{M}_k) \lambda_k} \quad \text{et} \quad \beta_{ij}(\underline{M}) = \bar{B}_{ij}(\underline{M}) \quad (14)$$

La génération de F_0 étant explicite, son coût est très modeste. Pour quantifier la précision de cette première reconstruction, on introduit la norme :

$$\|F\|_{I \times \Omega}^2 = \int_{I \times \Omega} F^2 d\Omega dt = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \int_{I_i \times \Omega_j} F^2 d\Omega dt \quad (15)$$

et on cherche à évaluer :

$$e = \frac{\|F - F_0\|}{\|F_0\|} \quad (16)$$

Le coût de calcul de cette quantité peut être limité en tirant parti du fait que F est le produit des deux fonctions PGD dont les termes peuvent être classés par ordre décroissant. On peut donc se contenter de prendre en compte les premiers termes de f et f' pour avoir une bonne approximation de e . Si e dépasse une tolérance fixée, on peut ajouter à la première approximation F_0 des termes correctifs PGD afin de représenter plus finement la fonction. Cependant, dans bien des cas, la qualité de la première approximation est suffisante.

5.4 Exemple numérique

On considère deux fonctions aléatoires f et f' décrites sous forme PGD sur une base réduite de 10 fonctions ($m = m' = 10$). On souhaite approcher leur produit sous forme compressée selon la démarche précédente afin d'en fournir une représentation PGD avec un seul couple de fonctions espace-temps.

La solution de référence, exacte, (**Fig. 3**, gauche) est obtenue en évaluant le produit des champs *full* à partir des représentations PGD des fonctions f et f' :

$$F(t, \underline{M}) = f(t, \underline{M}) f'(t, \underline{M}) = \left(\sum_{i=1}^{10} \lambda_i(t) \Lambda_i \right) \left(\sum_{i=1}^{10} \theta_i(t) \Theta_i \right)$$

La reconstruction à partir d'une forme compressée à $m_1 = 17$ instants de référence et $m_2 = 17$ points de référence (**Fig. 3**, droite) conduit à une erreur par rapport à la solution exacte de $e = 1\%$ et ceci bien qu'un seul couple PGD de F ne soit reconstruit par patch espace-temps.

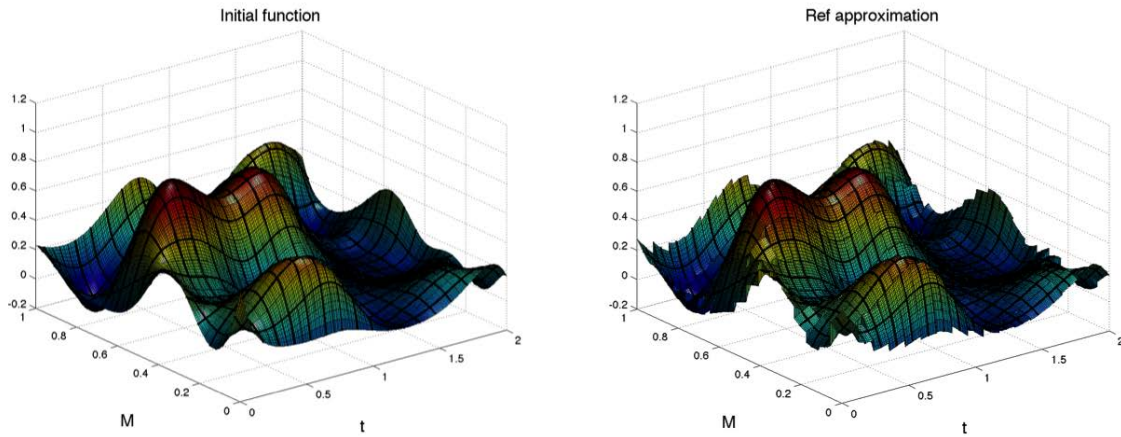


Fig. 3 – Fonction $F = ff'$ exacte (à gauche) et son approximation à 17 points de référence et 17 instants de référence (à droite)

6 Conclusions

Les techniques de réduction de modèles basées sur la PGD peuvent servir de fondement pour l'écriture de codes de calcul hautes performances permettant la simulation de modèles complexes, à très grand nombre de degrés de liberté et éventuellement paramétriques. Cependant, dans les codes développés jusqu'à présent, seule une partie des champs était représentée sous forme PGD, ce qui conduisait à manipuler des champs définis de manières différentes et ne permettait pas de tirer parti de la séparation de variables pour simplifier le calcul et le stockage.

Dans ce travail, nous avons introduit les bases d'une représentation « compressée » qui permet d'envisager une écriture de tous les champs sous forme PGD. Les opérations algébriques élémentaires peuvent être réécrites dans ce cadre, afin de donner une première approximation à variables séparées des résultats de celles-ci. Les travaux d'implémentation de cette algorithmie dans un code de calcul complet sont actuellement en cours et devraient permettre une forte réduction des temps de simulation.

Références

- [1] Ladevèze, P., Néron, D. and Passieux, J.-C., On a multiscale computational mechanics with time-space homogenization, In *Multiscale methods—Bridging the scales in Science and Engineering*, Oxford University Press. Ed. J. Fish, ch. Space Time Scale Bridging methods, 26(1) :247-282, 2009.
- [2] Chinesta, F., Ladevèze, P., and Cueto, E., A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4) :395-404, 2011.
- [3] Ladevèze, P., *Nonlinear Computational Structural Mechanics—New Approaches and Non-Incremental Methods of Calculation.*, Springer Verlag, 1999.

- [4] Carlberg, K., Bou-Mosleh, C., and Farhat, C., Efficient nonlinear model reduction via a least-squares petrov-galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2) :155-181, 2011.
- [5] Ladevèze, P., *A computational technique for the integrals over the time-space domain in connection with the LATIN method*. Technical Report 193 (in French), LMT-Cachan, 1997.
- [6] Ladevèze, P., Nouy, A., On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 192 :3061-3087, 2003.
- [7] Ladevèze, P. *New algorithms : mechanical framework and development*, Technical Report 57 (in French), LMT-Cachan, 1985.