

## Stratégies avancées d'optimisation et analyse inverse basées sur l'utilisation de la PGD.

C. Ghnatios<sup>1</sup>, F. Chinesta<sup>2</sup>, A. Leygue<sup>1</sup>,  
P. Villon<sup>3</sup>, P. Breïtkopf<sup>3</sup>, A. Poitou<sup>1</sup>

<sup>1</sup> Ecole Centrale de Nantes, France, {Chady.Ghnatios,Adrien.Leygue,Arnaud.Poitou}@ec-nantes.fr

<sup>2</sup> EADS Corporate Foundation International Chair-Ecole Centrale de Nantes, France, Francisco.Chinesta@ec-nantes.fr

<sup>3</sup> Université de Technologie de Compiègne, France, {Pierre.Villon,Piotr.Breïtkopf}@utc.fr

---

**Résumé** — L'optimisation des procédés de fabrication nécessite souvent le choix optimal des paramètres des procédés. La plupart des méthodes classiques calculent la solution pour un certain nombre de choix de ces paramètres. Les méthodes de descente itérative sont utilisées pour atteindre l'optimum, avec la nécessité de résoudre à chaque itération le problème direct. Dans ce travail, on abordera une nouvelle méthode, qui consiste à calculer off-line la solution du modèle paramétré, où toutes les sources de variabilité ont été introduites comme des extra-coordonnées, et ensuite résolu par la PGD qui est capable d'éviter la malédiction de la multi-dimensionnalité [1], pour ainsi retrouver l'optimum. Le couplage entre une solution d'un modèle enrichi multidimensionnel calculée off-line par la PGD et une technique d'optimisation on-line faisant appel aux moindres carrés mobiles est un choix pertinent pour s'affranchir de la résolution efficace des problèmes d'optimisation et d'analyse inverse.

---

## 1 Construction du modèle paramétrique dans la PGD

La PGD ou « Proper Generalized Decomposition » est une nouvelle méthode de calcul numérique qui consiste à décrire les variables du modèle en forme séparée. Ainsi la complexité résultante évolue linéairement avec le nombre de dimensions du modèle [2], au contraire des méthodes de discrétisation basées sur la construction d'un maillage de l'espace où la difficulté évolue exponentiellement avec le nombre de dimensions du problème. Avec la PGD, si on a un problème 3D, par exemple, on pourra séparer la solution comme une suite de solutions de problèmes de dimension plus réduite ( $3 \times 1D$  ou  $1D+2D$  par exemple). Pour cela, la PGD combine représentation séparée avec un algorithme de construction des fonctions intervenant dans la forme séparée de la solution.

Dans un problème d'optimisation on souhaite calculer un certain nombre de paramètres pour minimiser une certaine fonction cout. Classiquement, on fait un choix de ces paramètres et on évalue la fonction cout. Si elle n'est pas admissible, on corrige le choix de paramètres, avec une stratégie adéquate, et on recalcule la solution du problème. Ainsi, pour chaque choix de paramètres testé il faut résoudre le problème pour ensuite évaluer la fonction cout. Optimisation et discrétisation vont ensemble. Une possibilité intéressante consiste à introduire tous les paramètres du modèle comme extra-coordonnées dans le modèle qui devient ainsi multidimensionnel, mais comme nous venons de voir cela n'entraîne aucune difficulté majeure quand on procède à sa résolution avec la PGD [1]. Ainsi, une fois résolu, nous aurons accès à la solution du problème dans tout point, instant de temps et pour tout choix possible des paramètres. Il faut signaler que l'obtention des sensibilités, dérivées de la fonction cout par rapport aux paramètres est explicite. Maintenant, l'optimisation ne nécessite aucune nouvelle résolution puisque la solution est connue pour tout choix des paramètres. On pourrait imaginer une résolution off-line et une optimisation on-line, et en tout cas on évite la nécessité de résoudre un problème directe dans chaque itération de l'algorithme de optimisation.

## 2 Introduction des MLS

Une fois le solution paramétrique construite par la PGD, on a un problème d'optimisation à très haute dimension. La formulation d'une fonction coût ainsi que la résolution du problème d'optimisation par

les méthodes classiques s'avère très coûteuse. Donc, on propose d'utiliser une méthode type surface de réponse combinée aux MLS (« Moving Least Squares »).

La MLS est une méthode d'approximation sans maillage qui procède à partir d'un ensemble de points d'échantillonnage, via les moindres carrées pondérées au voisinage d'un point de départ [3]. Une reconstruction avec des fonctions basées sur le développement de Taylor révèle plusieurs avantages dans l'optimisation [4]. On illustrera la construction de la MLS dans notre modèle, tout en utilisant les hyper-cubes LATIN, qui réduisent aussi le temps de reconstruction considérablement, tout en gardant un nombre de points d'échantillonnage nécessaire pour la bonne représentation de la solution.

### 3 Combinaison de la PGD à la MLS dans un algorithme d'optimisation

Une fois le modèle paramétrique construit par la PGD, on commence par définir une fonction coût. Cette fonction coût a comme dimension le nombre de paramètres du modèle, qui peut atteindre des valeurs où le calcul classique devient prohibitif. Puis, on agira en partant d'un point arbitraire, à la reconstruction de la solution par les MLS. On verra donc l'intérêt de l'utilisation des fonctions basées sur le développement de Taylor dans la reconstruction. Ensuite, on agira par une stratégie de descente du type Quasi-Newton pour déterminer le point suivant, autour duquel on répètera la même démarche.

#### Références

- [1] F. Chinesta, A. Ammar, E. Cueto. *Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models*, Archives of Computational Methods in Engineering, in press.
- [2] F. Chinesta, A. Ammar, E. Cueto. *Proper generalized decomposition of multiscale models*, International Journal of Numerical Methods in Engineering, in press.
- [3] P. Breitkopf, H. Naceur, A. Racineux, P. Villon. *Moving least squares response surface approximation formulation and metal forming applications*, Computers And Structures, 1411-1428, 2005.
- [4] P. Breitkopf, A. Racineux, P. Villon. *An introduction to moving least squares meshfree methods*, Meshfree Computational Mechanics, 825-867, 2002.