Optimisation de structures par couplage métamodèles multi-fidélité et modèles réduits

Stéphane Nachar[1], Pierre-Alain Boucard[1], David Néron[1]

1. LMT/ENS-Cachan/CNRS/Université Paris-Saclay
61 avenue du président Wilson, 94235 Cachan, FRANCE
{stephane.nachar,pierre-alain.boucard,david.neron}@lmt.ens-cachan.fr

Résumé :

Un des principaux points de blocage de l'optimisation des systèmes mécaniques est le temps de simulation prohibitif auquel elle peut conduire, notamment dans le cas non linéaire. L'utilisation de métamodèles multi-fidélités alimentés par des simulations numériques de précision variable permet de réduire le temps de calcul dans les zones éloignées de l'optimum. Mais pour exploiter au mieux ces métamodèles, il est nécessaire de disposer d'un simulateur efficace, permettant de générer ces simulations avec un niveau de précision adapté. L'idée développée ici est d'utiliser la technique de réduction de modèles PGD. Son intérêt est qu'elle produit naturellement des solutions de précision variable en fonction du nombre de modes utilisés. Dans le cas non linéaire, la PGD utilisée ici est une décomposition "espace-temps" couplée avec la méthode LATIN pour le traitement des non linéarités.

Abstract :

Computation time is the main limitation to use optimisation toolbox on mechanical models, especially in the nonlinear case. The use of multi-fidelity metamodels fed by numerical simulations of variable precision makes it possible to reduce time computation in areas which are far from the optimum. Hence to use as the best these metamodels, it is necessary to have an efficient solver, allowing to generate simulations with a given level of precision. The idea developed here is to use PGD models reduction technique. They product naturally solutions whose precision levels varies according to the number of modes used. In the nonlinear case, the PGD used here is a "space-time" decomposition coupled with the LATIN method for the treatment of material nonlinearities.

Mots clefs : Optimisation, métamodèle multi-fidélité, modèles réduits

1 Introduction

Les outils de simulation ont révolutionné le processus de conception et de dimensionnement dans les bureaux d'études en permettant des calculs toujours plus fidèles à la physique. Cependant, l'optimisation globale de systèmes complexes se heurte encore à des difficultés d'ordre méthodologique, organisationnelle, informatique et numérique. Un verrou clairement identifié concerne les coûts inhérents à l'appel systématique aux modèles numériques qui rendent l'optimisation directe des systèmes difficilement envisageable. À titre d'exemple, pour l'étude d'un joint de transmission [1] comportant 12 boulons, 60 zones de contact et 150 000 ddls, une simulation demande environ 1h20. Si on se fixe comme objectif de maximiser le couple transmissible en jouant sur le serrage des 12 boulons, on est facilement amené à réaliser un millier de simulations (pour différents jeux de paramètres de serrage), ce qui conduirait à plus de 6 semaines de calcul. Il apparait donc clairement qu'il faut mettre en place des stratégies adaptées pour réduire ces coûts de simulation. L'approche directe étant impossible avec la grande majorité des algorithmes d'optimisation, plusieurs méthodes ont été développées afin de réduire le coût numérique de l'optimisation.

2 Génération d'un modèle de substitution pour l'optimisation

Pour diminuer les coûts de calcul, deux grandes classes de méthodes peuvent être utilisées : les méthodes utilisant des modèles de substitution et les méthodes utilisant des modèles réduits.

Les méthodes utilisant des modèles de substitution consistent à remplacer la fonction objectif coûteuse à obtenir (souvent issue d'un modèle EF) par ce qu'on appelle un métamodèle. Celui-ci s'apparente au résultat d'une régression réalisée à partir de valeurs de la fonction objectif issu d'une simulation numérique pour quelques jeux de paramètres. Il existe plusieurs types de métamodèles comme les fonctions à bases radiales [8], les machines à support de vecteur [12], ou encore le krigeage [4]. Le krigeage est particulièrement intéressant dans le cadre de l'optimisation car il a permis la mise en place d'une stratégie adaptative d'amélioration du métamodèle (en le rendant ainsi plus précis pour un coût supplémentaire faible) [7].

2.1 Krigeage

Le krigeage est une méthode de régression par processus gaussien conditionné par les observations. Considérons la fonction objectif $Y(\boldsymbol{\mu})$ calculée uniquement en quelques points $\boldsymbol{\mu_i}$ avec $Y(\boldsymbol{\mu_i}) = y_i$, l'idée du krigeage est de ne plus considérer les valeurs calculées comme caractérisant le modèle de manière purement déterministe, mais en considérant l'approximation par une composante déterministe et une composante aléatoire. On définit ainsi un métamodèle associé $\hat{Y}(\boldsymbol{\mu})$:

$$\hat{Y}(\boldsymbol{\mu}) = \langle f(\boldsymbol{\mu_i}), \beta \rangle + Z(\boldsymbol{\mu_i}),\ (\boldsymbol{\mu_i}) \in \mathcal{D} \tag{1}$$

$f(\boldsymbol{\mu})$: fonctions déterministes (**choisies**) de tendance
β : coefficients (inconnus) de pondération des fonctions de tendance
$Z(\boldsymbol{\mu})$: processus gaussien stationnaire de covariance $C(\boldsymbol{\mu}, \boldsymbol{\mu}')$ (inconnu)
$\mathcal{D}$: espace de conception

Par construction, le métamodèle $\left\{\hat{Y}(\boldsymbol{\mu_i}), (\boldsymbol{\mu_i}) \in \mathcal{D}\right\}$ devient un champ aléatoire gaussien de moyenne $\boldsymbol{\mu} \longrightarrow \langle f(\boldsymbol{\mu}), \beta \rangle$ et de covariance choisie $(\boldsymbol{\mu}, \boldsymbol{\mu}') \longrightarrow C(\boldsymbol{\mu}, \boldsymbol{\mu}')$.

En utilisant les propriétés de conditionnement des processus gaussiens,ainsi que le Best Linear Unbiased Predictor [5], on définit la variable aléatoire gaussienne $\hat{Y}(\boldsymbol{\mu})$, de moyenne $\bar{x}(\boldsymbol{\mu})$ et de variance $\sigma^2(\boldsymbol{\mu}, \boldsymbol{\mu}')$ telle que :

$$\left\{ \begin{array}{lcl} \bar{x}(\boldsymbol{\mu}) & = & \langle f(\boldsymbol{\mu}), \hat{\beta} \rangle + r(\boldsymbol{\mu})^T [C]^{-1} (\boldsymbol{y} - [F]\ \hat{\beta}) \\ \sigma^2(\boldsymbol{\mu}, \boldsymbol{\mu}') & = & [C](\boldsymbol{\mu}, \boldsymbol{\mu}) - r(\boldsymbol{\mu})^T [C]^{-1} r(\boldsymbol{\mu}) + \boldsymbol{u}(\boldsymbol{\mu})^T ([F]^T [C]^{-1} [F])^{-1} \boldsymbol{u}(\boldsymbol{\mu}) \end{array} \right. \tag{2}$$

$$\text{avec} \left\{ \begin{array}{lcl} \hat{\beta} & = & ([F]^T [C]^{-1} [F])^{-1} [F]^T [C]^{-1} \boldsymbol{y} \\ \boldsymbol{u}(\boldsymbol{\mu}) & = & [F]^T [C]^{-1} r(\boldsymbol{\mu}) - f(\boldsymbol{\mu}) \end{array} \right.$$

$[C]$: matrice de covariance
$[F]$: matrice de regression
$\boldsymbol{y}$: vecteur des points échantillonnés y_i

Par construction, le métamodèle créé $\hat{Y}(\boldsymbol{\mu})$ est ainsi :

— non-biaisé : $\mathbb{E}([\hat{Y}(\boldsymbol{\mu})]) = \mathbb{E}([Y(\boldsymbol{\mu})])$
— optimal dans l'ensemble des prédictions linéaires non-biaisés , au sens de la minimisation de l'erreur de prédiction L^2 :

$$\hat{Y}(\boldsymbol{\mu}) = \arg\min_g \mathbb{E}\left[(Y(\boldsymbol{\mu}) - g\boldsymbol{\mu})^2\right] \tag{3}$$

— interpolant vis-à-vis des observations : $\mathcal{P}(\hat{Y}(\boldsymbol{\mu}_i) = y_i) = 1$.

Cette méthode de génération d'un modèle de substitution pour la fonction objectif est intéressante dans le cadre d'une optimisation car la variance générée permet d'enrichir le modèle : celle-ci peut donner une indication sur la zone à enrichir pour diminuer au maximum l'erreur au moindre carré (Mean Square Error criterion), ou sur la zone où la probabilité d'avoir le minimum est la plus forte (Expected Improvement criterion [3]).

Ainsi, après chaque calcul ou batch de calculs, la génération du métamodèle donne à la fois une image de la fonction objectif, une estimation de l'erreur, et les jeux de paramètres à utiliser dans le solveur mécanique pour générer de nouvelles valeurs de la fonction objectif propres à améliorer le métamodèle, ou à s'approcher du minimum.

2.2 Krigeage multi-fidélité

Il ne s'agit pas des seules propriétés du krigeage. L'aspect stochastique du métamodèle généré permet la prise en compte de données de qualité diverses. Considérons deux types de données d'entrée :

— Des données basse fidélité comme un champ de déplacement approximé $\tilde{u}_{BF}(\boldsymbol{\mu})$. La valeur de la fonction objective associée serait $y_{BF,i} = Y(\tilde{u}_{BF}(\boldsymbol{\mu_i}))$.
— Des données haute fidélité comme un champ de déplacement approximé $u(\boldsymbol{\mu})$. La valeur de la fonction objective associée serait $y_{HF,i} = Y(\tilde{u}_{HF}(\boldsymbol{\mu_i}))$.

Il est possible via une méthode de krigeage multi-fidélité de prendre en compte ces deux types de données.

Citons deux familles de méthodes de krigeage multi-fidélité : une première utilise le cokrigeage et nécessite en particulier de définir la covariance associée aux données BF/HF [14], une seconde utilise de multiples métamodèles, que ce soit des métamodèles récursifs comme dans le cadre du krigeage hiérarchique [6], ou de métamodèles d'erreur comme dans le cadre de l'Evofusion [7].

Des études ont déjà été réalisées dans le cadre de données issues de calculs analytiques et de simulation, ou de données partiellement et totalement convergées [2]. Dans le cadre de cette étude, l'usage de modèles réduits comme données multi-fidélité va être présenté.

3 Une nouvelle approche du krigeage multi-fidélité par l'usage de modèles réduits

Dans le cadre de la génération de données multi-fidélité pour la création d'un modèle de substitution, nous avons choisi d'utiliser la Proper Generalized Decomposition [9]. Il existe d'autres méthodes telles que la Proper Orthogonalized Decomposition (POD) [13], ou encore les Reduced-Basis [8] que nous ne développerons pas ici.

3.1 Approximation par la méthode PGD

La PGD a été développée initialement dans le cadre du framework LATIN pour une décomposition espace/temps. L'objectif de cette méthode est de générer la meilleure approximation possible d'une solution avec une représentation séparée des variables. Dans le cadre de la PGD, on n'impose plus la forme de la base comme dans la POD et on cherche la solution du problème en même temps que la meilleure base pour sa représentation.

On donne ici les principaux éléments permettant de construire une représentation PGD solution du problème suivant [15] :

$$\text{Trouver } \boldsymbol{u}^{(\boldsymbol{\mu_i})} \in \mathcal{V} / \forall \boldsymbol{v} \in \mathcal{V}_0, a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu_i}) = \ell(\boldsymbol{v}; \boldsymbol{\mu_i}), \quad \forall t \in I = [0, T] \tag{4}$$

avec $\boldsymbol{u}$ fonction de $(\boldsymbol{x}, t) \in \Omega \times I$, $\mathcal{W} = L^2(I, \mathcal{V})$, $\mathcal{V} = \left\{ \boldsymbol{v} \in \mathcal{H}^1(\Omega), \text{cinématiquement admissible} \right\}$. a est un opérateur bilinéaire coercif continu selon $\mathcal{H}_0^1$ et ℓ est un opérateur linéaire continu selon $\mathcal{H}_0^1$. On définit également $\mathcal{I} = L^2(I, \mathbb{R})$ l'espace des functions à carré sommable dans I.

Une approximation de la solution $\boldsymbol{u}$ est définie en se plaçant dans l'approximation de $\mathcal{W}$, $\tilde{\mathcal{W}} = \mathcal{V} \otimes \mathcal{I}$. On définit ainsi approximation de rang réduit $\tilde{\boldsymbol{u}}$ sous la forme d'une somme de fonctions de l'espace et du temps :

$$\tilde{\boldsymbol{u}}_m^{(\boldsymbol{\mu_i})}(\boldsymbol{x}, t) = \sum_{k=1}^{m} \lambda_k^{(\boldsymbol{\mu_i})}(t) \, \boldsymbol{\Lambda}_k^{(\boldsymbol{\mu_i})}(\boldsymbol{x}) \tag{5}$$

avec $(\boldsymbol{\Lambda}_k, \lambda_k) \in \mathcal{V} \times \mathcal{I}$

De multiples choix sont possibles pour construire les modes PGD (Méthode d'orthogonalisation de Galerkin, Minimisation du résidu...) [10]. La méthode de Galerkin est rappellée ici et s'appuit sur la formulation faible temps-espace de (4) :

$$\text{Trouver } \boldsymbol{u}^{(\boldsymbol{\mu_i})} \in \tilde{\mathcal{W}} / \forall \boldsymbol{v} \in T(\tilde{\mathcal{W}}), A(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu_i}) = L(\boldsymbol{v}; \boldsymbol{\mu_i}) \tag{6}$$

avec $A(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu_i}) = \int_I a(\boldsymbol{u}, \boldsymbol{v}; \boldsymbol{\mu_i}) dt$, $L(\boldsymbol{v}; \boldsymbol{\mu_i}) = \int_I \ell(\boldsymbol{v}; \boldsymbol{\mu_i}) dt$
$T(\tilde{\mathcal{W}})$ l'espace tangent linéaire à $\tilde{\mathcal{W}}$.

La construction des modes est réalisée par un algorithme glouton. Supposons qu'une approximation d'ordre $(m-1)$ est définie :

$$\tilde{\boldsymbol{u}}_{m-1}^{(\boldsymbol{\mu_i})}(\boldsymbol{x},t) = \sum_{k=1}^{m-1} \lambda_k^{(\boldsymbol{\mu_i})}(t)\ \boldsymbol{\Lambda}_k^{(\boldsymbol{\mu_i})}(\boldsymbol{x}) \tag{7}$$

On améliore l'approximation de la solution de (6) usuellement par deux étapes. Une première étape de correction des fonctions temporelles $\lambda_k^{(\boldsymbol{\mu_i})}$ pour améliorer l'approximation de la solution sans regénerer de modes. Si besoin, une seconde étape de génération d'un nouveau mode est exécuté. On cherche une nouvelle approximation $\tilde{\boldsymbol{u}}_m^{(\boldsymbol{\mu_i})} = \tilde{\boldsymbol{u}}_{m-1}^{(\boldsymbol{\mu_i})} + \delta u = \tilde{\boldsymbol{u}}_{m-1}^{(\boldsymbol{\mu_i})} + \boldsymbol{\Lambda}\lambda$. Cette nouvelle paire $(\boldsymbol{\Lambda}, \lambda)$ est issue de la résolution suivante :

$$\begin{aligned}\text{Trouver } (\boldsymbol{\Lambda},\lambda) \in \mathcal{V}\times\mathcal{I} \ /\ \forall(\boldsymbol{\Lambda}^*,\lambda^*)\in\mathcal{V}\times\mathcal{I},\\ A(\tilde{\boldsymbol{u}}_{m-1}^{(\boldsymbol{\mu_i})}+\boldsymbol{\Lambda}\lambda, \boldsymbol{\Lambda}^*\lambda+\boldsymbol{\Lambda}\lambda*) = L(\boldsymbol{\Lambda}^*\lambda+\boldsymbol{\Lambda}\lambda*)\end{aligned}$$

L'équation est divisé en deux sous-équations :

$$\begin{aligned}\text{Trouver } (\boldsymbol{\Lambda},\lambda) \in \mathcal{V}\times\mathcal{I} \ /\\ \forall(\boldsymbol{\Lambda}^*)\in\mathcal{V}, A(\boldsymbol{\Lambda}\lambda,\boldsymbol{\Lambda}^*\lambda) = L(\boldsymbol{\Lambda}^*\lambda) - A(\tilde{\boldsymbol{u}}_{m-1}^{(\boldsymbol{\mu_i})},\boldsymbol{\Lambda}^*\lambda)\\ \forall(\lambda^*)\in\mathcal{I}, A(\boldsymbol{\Lambda}\lambda,\boldsymbol{\Lambda}\lambda^*) = L(\boldsymbol{\Lambda}\lambda^*) - A(\tilde{\boldsymbol{u}}_{m-1}^{(\boldsymbol{\mu_i})},\boldsymbol{\Lambda}\lambda^*)\end{aligned}$$

Le détail d'un cas de résolution peut être trouvé dans [11]. Usuellement, un algorithme de point fixe est mis en oeuvre. Cette méthode permet générer à la volée des modes de la solution, en choisissant l'approximation du résultat réalisé par le nombre de modes générés m. Ainsi, on définit un nombre de modes pour la solution BF m_{BF}, et un nombre de modes pour la solution HF m_{HF}.

Ici, la problématique est de déterminer le nombre de modes m_{BF} nécessaire pour que la valeur basse fidélité de fonction objectif contienne suffisamment d'informations pour être viable pour le métamodèle, mais que le coût de cette information basse fidélité reste faible. Ce coût est lié aussi au nombre de points initiaux utilisés pour construire le métamodèle.

3.2 Génération de métamodèles utilisant des modèles réduits

L'association métamodèles - modèles réduits se fait naturellement au sein de l'algorithme 1 : les deux méthodes pouvant communiquer en boite noire. Dans un premier temps, on génère un métamodèle BF qui s'appuie sur n_{init} points par dimension de $\mathcal{D}$ et sur une troncature à l'ordre m_{BF} de la solution du solveur. Dans un second temps, le métamodèle est enrichi en générant des points haute fidélité via la résolution exacte du problème. Les points sont choisis dans $\mathcal{D}$ suivant un critère d'amélioration comme l'erreur aux moindres carrés ou la *probabilité d'amélioration EI* [3].

L'objectif final est de diminuer le coût de calcul, qui dépend en particulier du choix de n_{init} et m_{BF}.

3.3 Un cas-test simple

Une première étude de cet algorithme a été effectuée avec une structure bi-matériau constituée d'un assemblage de poutres métalliques formant une structure auxétique "Bow tie". Au sein de chaque motif

Algorithm 1 Génération de métamodèles par utilisation de modèles réduits

1: n_{init} tirages par dimension dans $\mathcal{D}$: $(\boldsymbol{\mu_i})$
2: Evaluation basse fidélité des tirages via le solveur PGD :

$$\tilde{u}_{BF}^{(\boldsymbol{\mu_i})}(\boldsymbol{x},t) \equiv \sum_{k=1}^{m_{BF}} \lambda_k^{(\boldsymbol{\mu_i})}(t)\boldsymbol{\Lambda}_k^{(\boldsymbol{\mu_i})}(\boldsymbol{x}) \tag{8}$$

3: Evaluation de la fonction coût aux points calculés :

$$Y(\boldsymbol{\mu_i}) = \mathcal{F}(u^{(\boldsymbol{\mu_i})}(\boldsymbol{x},t)) \tag{9}$$

4: Création du Métamodèle : $\bar{x}(\boldsymbol{\mu}), \sigma^2(\boldsymbol{\mu})$
5: Evaluation du Métamodèle :

$$\hat{Y}(\boldsymbol{\mu}) = \langle f(\boldsymbol{\mu}), \beta \rangle + Z(\boldsymbol{\mu}) \tag{10}$$

6: **while** Critère d'arrêt ν non atteint **do**
7: Recherche zones à enrichir dans $\mathcal{D}$: $(\boldsymbol{\mu_j})$
8: Evaluation haute fidélité des zones :

$$\tilde{u}_{HF}^{(\boldsymbol{\mu_i})}(\boldsymbol{x},t) \equiv \sum_{k=1}^{m_{HF}} \lambda_k^{(\boldsymbol{\mu_i})}(t)\boldsymbol{\Lambda}_k^{(\boldsymbol{\mu_i})}(\boldsymbol{x}) \tag{11}$$

9: Evaluation de la fonction coût aux points calculés : $Y(\boldsymbol{\mu_i})$
10: Enrichissement du Métamodèle : $\bar{x}(\boldsymbol{\mu}), \sigma^2(\boldsymbol{\mu})$
11: **end while**

est placée une mousse.

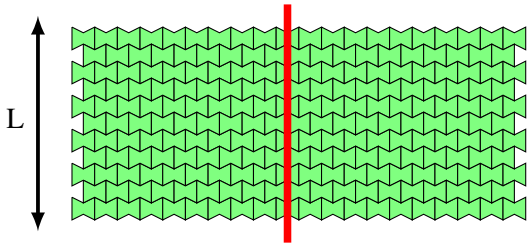


FIGURE 1 – Structure complète
En vert : la mousse interne à chaque structure
En rouge : l'axe vertical central

On souhaite une déformation transversale minimale de l'axe vertical central lors de son changement. Pour cela, on définit la fonction objectif Y suivante qu'on souhaite minimiser :

$$Y(\boldsymbol{\mu}) = \max_{t \in \mathcal{I}} |\Delta L^{(\boldsymbol{\mu})}(t)| \tag{12}$$

L est visible sur la figure 1. Les paramètres d'optimisation sont le ratio entre les modules d'Young des poutres et de la mousse, et le coefficient de Poisson de la mousse.

La complexité du chargement mis en place nécessite 10 modes pour définir la solution exacte du problème.

Nous allons évaluer sur ce cas-test l'impact du nombre de points par dimension n_{init} et la troncature

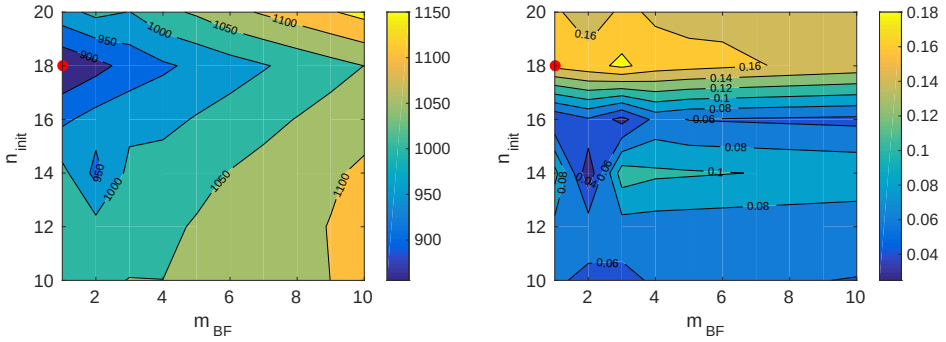
BF m_{BF} réalisée lors du tirage sur le temps de calcul, sachant que l'enrichissement est réalisé avec la solution exacte. L'algorithme a été testé dans un premier temps pour obtenir $\hat{Y}$ sur l'espace de conception $\mathcal{D}$. $\hat{Y}$ est comparé à $\hat{Y}_{ref}$ obtenue en générant le métamodèle avec $\boldsymbol{\mu_i}$ choisis sur une grille régulière 30×30 de $\mathcal{D}$ tous calculés avec la solution exacte du déplacement. On considère le métamodèle convergé lorsque l'erreur relative en norme 2 entre les fonctions atteint 5%.

Les deux étapes de création du métamodèle font générer des modes par le solveur. Le coût de calcul du métamodèle est proportionnel au nombre total m_{total} de modes générés lors de la création du métamodèle, et on s'appuit sur cette constatation pour caractériser l'impact de n_{init} et m_{BF} sur le temps de calcul. Considérons le cas où on initialise le métamodèle avec $n_{init} = 10$ points par dimension et avec une troncature de la solution à l'ordre $m_{BF} = 1$, et supposons qu'il faut $n_{enrich} = 20$ itérations à l'étape d'enrichissement, avec $m_{HF} = 10$, on aurait $m_{total} = 2 \times n_{init} \times m_{BF} + n_{enrich} \times m_{HF}$ soit 220 modes générés.

Le tirage initial LHS dans $\mathcal{D}$ des n_{init} points a un impact sur la génération du métamodèle. Pour s'affranchir de son influence, dix générations de métamodèles ont été réalisés pour chaque couple (n_{init},m_{BF}). La moyenne du nombre de modes totaux générés $moy(M_i)$ et l'ecart-type normalisé associé $\sigma = \dfrac{std(M_i)}{moy(M_i)}$ sont ainsi calculés.

Les résultats présentés figure 2 sont le nombre de modes M générés par le solveur pour obtenir $\hat{Y}$ sur $\mathcal{D}$ avec une erreur en norme 2 de 5%, en fonction du nombre de points par dimension n_{init} et de l'ordre de la troncature basse fidélité m_{BF}. Le meilleur résultat est le minimum.

L'analyse montre qu'initialiser à 18 points par dimension et une troncature à 1 mode est le meilleur choix pour limiter le coût de calcul.



(a) Moyenne des 10 générations

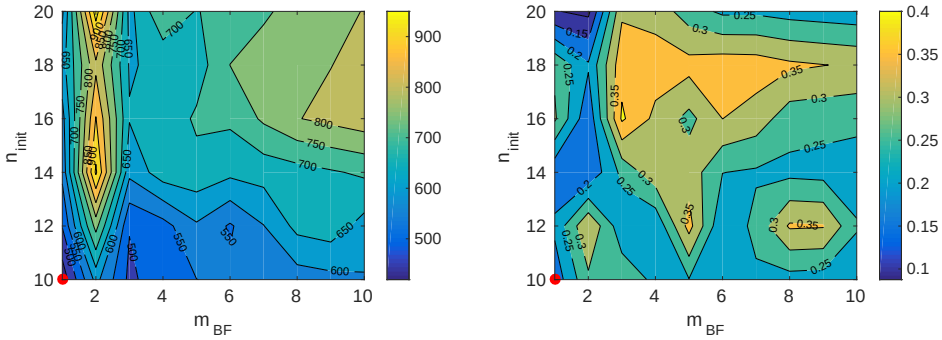
(b) Ecart type normalisé

FIGURE 2 – Génération du métamodèle complet

L'algorithme a été testé dans un second temps pour optimiser la fonction objectif sur l'espace de conception. Le critère d'arrêt est le suivant :

ARRET si $\min_{\boldsymbol{\mu_i} \in \mathcal{D}} |\hat{Y}(\boldsymbol{\mu_i}) - Y(\boldsymbol{\mu_i})| < 10^{-3}$ et $\boldsymbol{\mu}_m = \arg\min_{\boldsymbol{\mu_i} \in \mathcal{D}} |\hat{Y}(\boldsymbol{\mu_i}) - Y(\boldsymbol{\mu_i})|$ invariant pendant 5 itérations

Les résultats sont présentés en figure 3. L'analyse montre qu'initialiser à 10 points par dimension et une troncature à 1 mode est le meilleur choix pour réduire le temps de calcul tout en ayant un résultat robuste.



(a) Moyenne des 10 générations (b) Ecart type normalisé

FIGURE 3 – Détermination d'un optimum garanti

4 Conclusion

Ce premier exemple simple permet de montrer l'intérêt de coupler métamodèles et modèles réduits pour diminuer le temps de calcul dans le cadre de l'optimisation multi-fidélité. Il est néanmoins limitant, car il s'agit de la résolution d'un problème linéaire, ce qui explique sans doute que le gain optimal soit obtenu pour des données basse fidélité issues d'une troncature à l'ordre 1. Pour aller plus loin, il est nécessaire de mettre en oeuvre cette stratégie dans un cadre non-linéaire ce qui fait l'objet de travaux actuellement.

Références

[1] L. Laurent, P.A. Boucard, B. Soulier Generation of a cokriging metamodel using a multiparametric strategy Computational Mechanics 51 (2013) 151–169.

[2] N. Courrier, P.A. Boucard, B. Soulier The use of partially converged data simulations in building surrogate models. Advances in Engineering Software. 67 (2014) 186-197.

[3] D.R. Jones, Efficient global optimization of expensive black-box functions, Journal of Global optimization, 4 (1998) 455–492.

[4] D.G. Krige, A statistical approach to some mine valuation and allied problems on the Witwatersrand, (1951).

[5] A.S. Goldberger, Best Linear Unbiased Prediction in the Generalized Linear Regression Model, Journal of the American Statistical Association, 57 (1962) 369–375.

[6] Z. Han, R. Zimmermann, S. Görtz, A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data, AIAA Paper, 1225 (2010).

[7] A. Forrester, N. Bressloff, A. Keane, Optimization using surrogate models and partially converged computational fluid dynamics simulations, Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences, 462 (2006) 2177–2204.

[8] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations : An Introduction, Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences, 462 (2006) 2177–2204.

[9] P. Ladevèze, Nonlinear Computational Structural Mechanics - New Approaches and Non-Incremental Methods of Calculation, Springer-Verlag (1999).

[10] A. Nouy, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Comput Methods Appl Mech Eng 199 (2010) 1603–1626.

[11] D. Néron, P. Ladevèze, Proper generalized decomposition for multiscale and multiphysics problems. Arch Comput Methods Eng 17 (2010) 351–372

[12] V. Vapnik, S. Golowich, A. Smola Support vector method for function approximation, regression estimation, and signal processing. Advances in neural information processing systems 9 (1996).

[13] A. Chatterjee, An introduction to the proper orthogonal decomposition. Current science (2000).

[14] G. Matheron, The theory of regionalized variables and its applications. Vol. 5. École nationale supérieure des mines (1971).

[15] D. Néron, P.A. Boucard, N. Relun, Time-space PGD for the rapid solution of 3D nonlinear parametrized problems in the many-query context, Int. J. Numer. Meth. Engng. Vol 103. Num 4. (2015) 275–292.