



HAL
open science

Optimisation d'assemblages par métamodèles à gradients et stratégie multiparamétrique

Luc Laurent, Pierre-Alain Boucard, Bruno Soulier

► **To cite this version:**

Luc Laurent, Pierre-Alain Boucard, Bruno Soulier. Optimisation d'assemblages par métamodèles à gradients et stratégie multiparamétrique. 11e Colloque National en Calcul des Structures (CSMA 2013), CSMA, May 2013, Giens, France. hal-01431805

HAL Id: hal-01431805

<https://hal.science/hal-01431805>

Submitted on 11 Jan 2017

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.



Distributed under a Creative Commons Attribution 4.0 International License

Optimisation d'assemblages par métamodèles à gradients et stratégie multiparamétrique

Luc LAURENT¹, Pierre-Alain BOUCARD¹, Bruno SOULIER¹

¹ LMT-Cachan, (ENS Cachan/CNRS/UPMC/PRES UniverSud Paris), 61 av. du Président Wilson, F-94230 Cachan, France.
{laurent,boucard,soulier}@lmt.ens-cachan.fr

1 Introduction

Alors que l'emploi de méthodes d'optimisation évoluées apparaît comme de plus en plus incontournable au sein des bureaux d'études, les problèmes à résoudre et les stratégies employées pour y parvenir deviennent de plus en plus complexes. Les coûts de calcul relatifs à la résolution de ces problèmes deviennent également importants ce qui rend les stratégies d'optimisation classiques non viables. Afin de remédier à ce problème, il est possible d'agir de deux manières : la première consiste à modifier la stratégie de résolution en simplifiant le problème étudié ou en proposant une méthode de résolution moins coûteuse, tandis que la seconde consiste à déployer une stratégie d'optimisation plus adaptée. Dans le cadre de ce papier, nous proposons de nous intéresser aux problèmes d'optimisation paramétrique de structures assemblées en déployant une procédure d'optimisation à deux niveaux basée sur une stratégie de résolution mécanique dédiée, désignée sous le nom de *stratégie multiparamétrique* et l'emploi d'un métamodèle de type *krigeage* construit à partir des évaluations et des gradients de la fonction objectif. L'objectif, ici, est de réduire de manière significative le temps de calcul inhérent aux multiples résolutions mécaniques, fortement non linéaires, nécessaires au processus d'optimisation.

2 Optimisation à deux niveaux de modèles

La stratégie que nous proposons d'employer ici s'inscrit dans le contexte de l'optimisation multi-niveaux de modèles proposée par [1]. Constituée ici de deux niveaux (cf. FIG. 1), elle fera appel à deux modèles assurant la mise en œuvre de deux optimisations : la première consiste en une optimisation globale assurée par un algorithme génétique et réalisée sur un modèle approché nommé *métamodèle* tandis que la seconde consiste en une optimisation locale assurée par un algorithme basé sur la descente dans la direction du gradient utilisant les réponses fournies par un solveur dédié. L'optimisation locale fournira alors un optimum approché et un jeu de paramètres associés qui serviront de point d'initialisation de l'optimisation locale. La construction du métamodèle nécessitera également les réponses fournies par le solveur mais en seulement quelques points (jeux de paramètres). Les nombreuses évaluations nécessaires à l'optimiseur global pour converger se feront alors à un coût quasiment nul (par rapport au coût d'une évaluation fournie par le solveur mécanique).

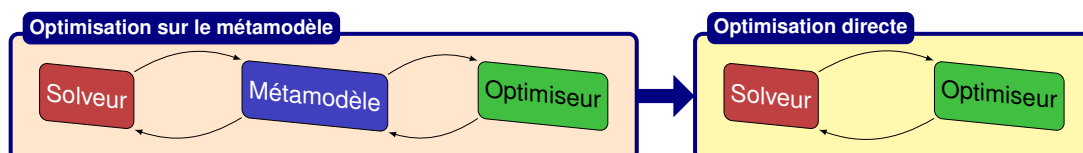


Fig. 1 – Optimisation à deux niveaux de modèles

3 Une stratégie de calcul dédiée aux études paramétriques

Dans le contexte de l'optimisation paramétrique de structures assemblées, les problèmes mécaniques à résoudre sont très similaires dans le sens où le problème de base et la géométrie sont identiques. Les problèmes diffèrent seulement par les valeurs prises par les différents paramètres pris en compte pour l'optimisation. C'est pourquoi une méthode dédiée à la résolution de ces problèmes très similaires est employée. Basée sur l'approche LATIN, la stratégie multiparamétrique assure une importante réduction des coûts de calcul grâce à une initialisation intelligente de l'algorithme de résolution.

3.1 La méthode LATIN

Introduite dans les années 80 par P. Ladevèze [2, 3] au LMT-Cachan pour résoudre des problèmes non-linéaires, la méthode LATIN (LARGE Time INcrement) permet de résoudre de manière itérative des problèmes d'assemblage. Dans le cadre de ce papier, les structures seront considérées comme élastique linéaire sous l'hypothèse des petites perturbations. Ainsi, les seules non-linéarités présentes correspondent aux comportements considérés entre les pièces de l'assemblage (contact et frottement). La stratégie de résolution LaTIn s'appuie sur 3 points principaux :

- la structure considérée sera en réalité constituée de *sous-domaines* et d'*interfaces*. Dans le cadre des assemblages, ce partitionnement est relativement naturel si l'on considère les pièces de l'assemblage comme étant les *sous-domaines* et les zones de contact inter-pièces comme étant les *interfaces*. On considérera une méthode de décomposition de domaine mixte (où forces et déplacements aux interfaces sont des inconnues du problème) ;
- l'algorithme itératif présenté ci-dessous sera employé pour résoudre le problème ;
- les opérateurs de la méthode restent constants et sont indépendants des chargements et des paramètres (coefficient de frottement, jeu) aux interfaces.

Finalement, le problème se compose en réalité de deux problèmes régis chacun par un groupe d'équations :

- un groupe d'équations *linéaires* désigné par \mathbf{A}_d , relatif aux sous-domaines ;
- un groupe d'équations *locales* (qui peuvent être non-linéaires ce qui est le cas ici) désigné par Γ qui, dans le cadre de cette étude, est relatif aux interfaces.

La résolution du problème complet consiste alors à résoudre ces deux groupes d'équations. L'algorithme LATIN propose de les résoudre successivement en faisant intervenir des directions de recherche assurant la communication entre les grandeurs bords des sous-domaines et les grandeurs d'interfaces (respectivement issues de la résolution des groupes d'équations \mathbf{A}_d et Γ [3]). La figure 2 illustre le fonctionnement de l'algorithme LATIN. La solution approchée obtenue à chaque itération de l'algorithme est définie sur

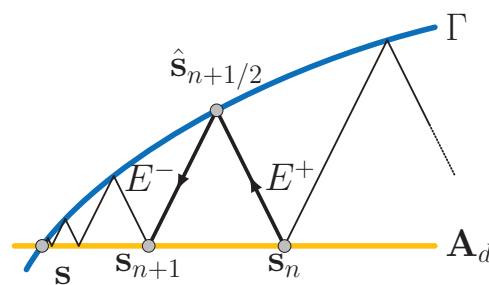


Fig. 2 – Représentation schématique de l'algorithme LATIN

l'intégralité des pas de chargement et en tous points de la structure. Finalement, au fur et à mesure des itérations, la solution s'enrichit jusqu'à convergence (qui est régie par un indicateur d'erreur [4]). Dans ce contexte, où le comportement des sous-domaines est linéaire élastique, la solution du problème est entièrement définie par la connaissance des déplacements (ou vitesses) \mathbf{W}_i et efforts \mathbf{F}_i dans les interfaces. On peut ainsi l'écrire sous la forme : $\mathbf{s} = \bigcup_{\text{interfaces}} (\mathbf{W}_i, \mathbf{F}_i)$.

3.2 La stratégie multiparamétrique

La stratégie multiparamétrique [5] s'appuie sur la méthode LATIN en proposant de réinitialiser "intelligemment" l'algorithme en utilisant une solution convergée précédemment calculée. Telle que le représente schématiquement la figure 3, cette réinitialisation permet d'obtenir une nouvelle solution associée à un nouveau jeu de paramètres en un nombre d'itérations plus faible ce qui se traduit par une réduction importante du temps de calcul. De plus, il a été montré que plus la variation de paramètres est faible, plus la réduction du nombre d'itérations est importante [6].

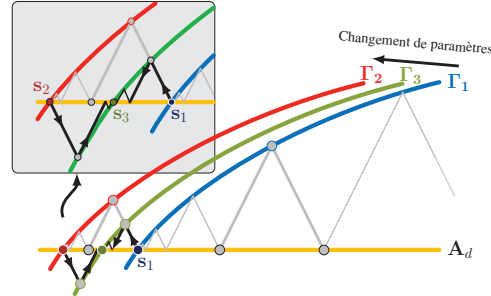


Fig. 3 – Représentation schématique de la stratégie multiparamétrique

3.3 Performances

Afin d'illustrer les performances de la stratégie multiparamétrique, un cas test comportant 5 paramètres a été mis en oeuvre. Ce dernier porte sur une modélisation 2D axisymétrique d'un montage fretté [7]. Plus précisément, on s'attache à calculer le couple maximum transmissible par une roue dentée assemblée sur un arbre par frettage en fonction des paramètres qui régissent le problème. Les figures 4a et 4b présente la géométrie du problème et sa modélisation 2D axisymétrique. Les paramètres pris en compte dans ce problème seront 3 coefficients de frottement :

- $\mu_{\text{inner/outer}}$ entre les bagues intérieure et extérieure de la frette ($0.05 \leq \mu_{\text{inner/outer}} \leq 0.5$) ;
- $\mu_{\text{gear/inner}}$ entre la bague intérieure de la frette et la roue dentée ($0.05 \leq \mu_{\text{gear/inner}} \leq 0.5$) ;
- $\mu_{\text{shaft/gear}}$ entre la roue dentée et l'arbre ($0.05 \leq \mu_{\text{shaft/gear}} \leq 0.5$) ;

et 2 jeux :

- $j_{\text{shaft/gear}}$ entre la roue dentée et l'arbre ($0\mu\text{m} \leq j_{\text{shaft/gear}} \leq 49\mu\text{m}$) ;
- $j_{\text{gear/inner}}$ entre la bague intérieure de la frette et la roue dentée ($36\mu\text{m} \leq j_{\text{gear/inner}} \leq 106\mu\text{m}$).

Les performances de la stratégie multiparamétrique sont présentées ici par le biais de 3 études basées sur l'étude du temps de calcul nécessaire à l'obtention de la réponse du modèle mécanique (couple maximum transmissible) pour un ensemble de jeux de paramètres. L'ensemble des combinaisons de paramètres étudiées est obtenu grâce à une stratégie de tirage de type LHS [8]. Le tableau 1 regroupe les caractéristiques de calcul relatives aux calculs des réponses pour 30 jeux de paramètres sans stratégie multiparamétrique, puis avec la stratégie multiparamétrique et enfin avec la stratégie multiparamétrique tout en calculant les gradients du couple maximum transmissible par rapport aux paramètres (en utilisant une méthode de différences finies à l'ordre 1).

Afin de comparer les performances de la stratégie un critère a été employé. Il quantifie le gain en terme de coût de calcul assuré par la stratégie multiparamétrique par rapport à la situation sans stratégie multiparamétrique. Il est calculé de la manière suivante :

$$\text{Gain} = \frac{\text{Nombre de calcul} \times \text{Temps CPU du premier calcul}}{\text{Temps CPU total avec la stratégie multiparamétrique}} \quad (1)$$

Au bilan, la stratégie multiparamétrique permet d'assurer le calcul des gradients pour un coût relativement réduit : en effet, dans le cas présenté ci-dessus, les réponses et gradients associés pour 30 jeux de paramètres sont obtenus avec un coût plus faible que celui nécessaire à l'obtention de **seulement** 30 réponses sans la stratégie multiparamétrique. Par ailleurs, les gradients (nécessitant 150 évaluations supplémentaires) sont obtenus avec un coût équivalent à celui du calcul des réponses.

C'est pour profiter au maximum de ces performances que l'emploi de métamodèles capables de prendre en compte les gradients de la fonction a été envisagé.

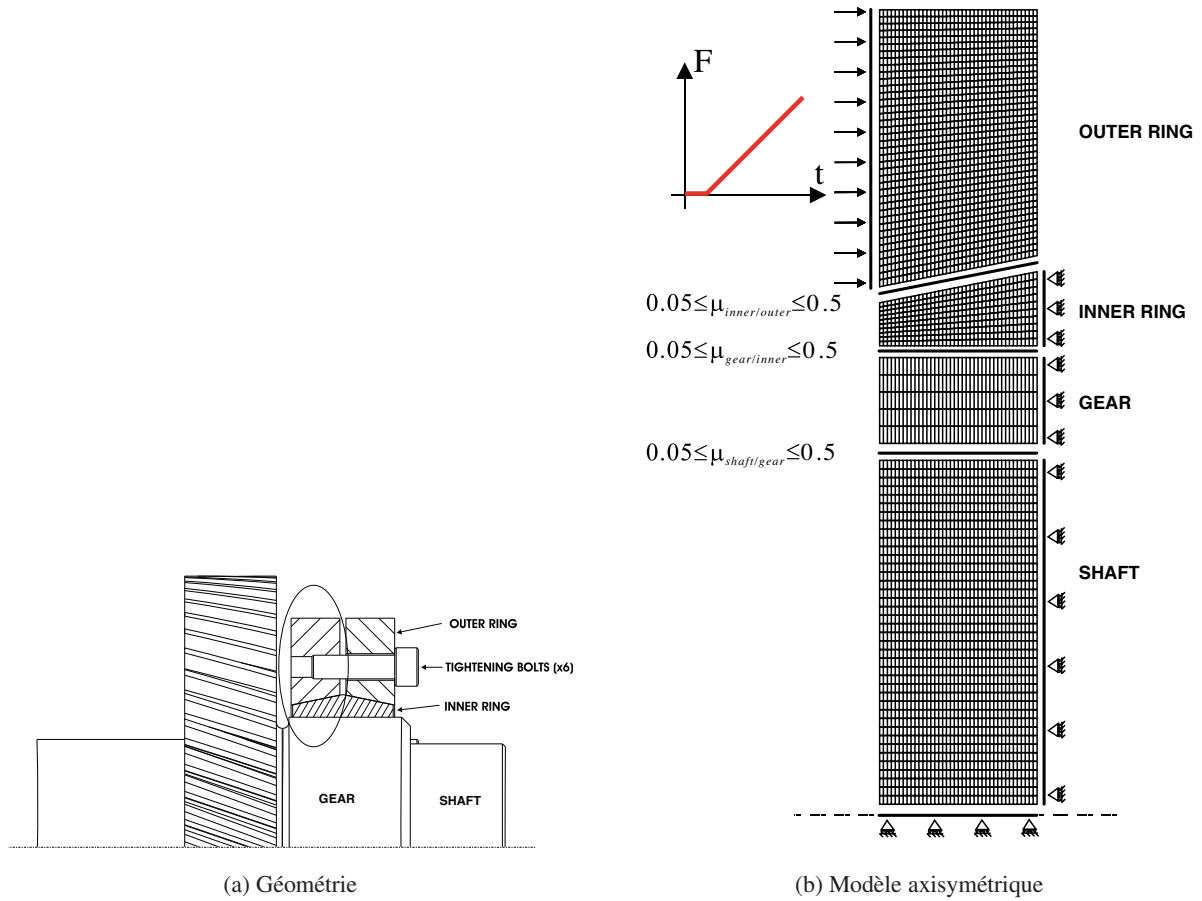


Fig. 4 – Assemblage fretté étudié

Multiparamétrique	Non	Oui	Oui
Calcul des gradients	Non	Non	Oui
Nombre d'évaluations	30	30	$30 + 5 \times 30$
Tps CPU total	202s	70s	147s
Tps CPU moyen par calcul	6.7s	2.3s	0.82s
Tps CPU minimum	4.3s	0.68	0.24s
Tps CPU maximum	9.3s	6.27s	6.2s
Nombre d'itérations	4765	1626	3066
Nombre moyen d'itérations	158	54	17
Nombre minimum d'itérations	101	15	5
Nombre maximum d'itérations	216	146	146
Gain	--	2.68	7.57

Tableau 1 – Illustration des performances de la stratégie multiparamétrique sur le cas test frette (5 paramètres)

4 Métamodèle à gradients

4.1 Notations

Dans cette partie, les notations suivantes seront employées : $\mathbf{x}^{(i)}$ et $\mathbf{x}^{(0)}$ désignent des points de l'espace de conception \mathcal{D} respectivement un des n_s points échantillonnés et un point quelconque de \mathcal{D} , x_i désigne la $i^{\text{ème}}$ composante d'un point \mathbf{x} , $Y(\mathbf{x})$ et $\tilde{Y}(\mathbf{x})$ désignent respectivement la réponse d'une fonction analytique ou d'un modèle mécanique et la réponse approchée fournie par le métamodèle. $R(\mathbf{x}^{(i)}, \mathbf{x}_j)$ est une fonction de corrélation.

4.2 Principe du cokrigage

Initialement introduit dans un contexte de géostatistique [9, 10], le principe du krigeage a été étendu à l'approximation de fonctions déterministes à la fin des années 80 [11]. Le cokrigage, quant à lui, est issu de la géostatistique multivariable [12] et a été étendu aux réponses et gradients par [13]. Egalement employé dans le contexte des processus gaussiens, le cokrigage fournit une approximation de la fonction connue en n_s points échantillonnés par sa réponse $Y(\mathbf{x}^{(i)})$ et son gradient $\frac{\partial Y}{\partial \mathbf{x}}(\mathbf{x}^{(i)})$. Le processus de construction consiste à considérer la fonction initiale déterministe connue en ces quelques points comme les réalisations d'un processus aléatoire gaussien $Y(\mathbf{x})$. On cherchera ce processus sous la forme d'une somme d'une fonction déterministe $\mu(\mathbf{x})$ (représentant la tendance du processus) et d'un processus aléatoire à moyenne nulle $Z(\mathbf{x})$ (correspondant aux corrections autour du modèle de tendance). On se placera dans le cas du krigeage ordinaire où la fonction déterministe $\mu(\mathbf{x})$ sera considérée constante.

4.3 Construction de l'approximation

La fonction $Y(\mathbf{x})$ ne peut être directement obtenue, on va donc chercher un estimateur $\tilde{Y}(\mathbf{x})$ de cette dernière sous la forme d'une fonction aléatoire décomposée sous la forme d'une combinaison linéaire décrite par l'équation 2.

$$\tilde{Y}(\mathbf{x}^{(0)}) = \sum_{i=1}^{n_s} \lambda_{0i}(\mathbf{x}^{(0)}) Y(\mathbf{x}^{(i)}) + \sum_{j=1}^{n_d} \sum_{i=1}^{n_s} \lambda_{ij}(\mathbf{x}^{(0)}) \frac{\partial Y}{\partial x_j}(\mathbf{x}^{(i)}) \quad (2)$$

La phase de construction consiste ensuite à déterminer les coefficients $\lambda_{ij}(\mathbf{x}^{(0)})$ tels que l'estimateur soit sans biais et minimise la variance de prédiction. Le problème de krigeage peut alors s'écrire sous la forme proposée en équation 3 (où \mathbb{E} et Var désignent respectivement l'espérance et la variance statistiques).

$$\forall \mathbf{x} \in \mathcal{D}, \lambda(\mathbf{x}) = \underset{\substack{\lambda^*(\mathbf{x}) \\ \text{tel que } \mathbb{E}[\varepsilon(\mathbf{x})]=0}}{\text{argmin}} \text{Var}[\varepsilon(\mathbf{x})] \text{ où } \varepsilon(\mathbf{x}) = Y(\mathbf{x}) - \tilde{Y}(\mathbf{x}) \quad (3)$$

Le développement de l'espérance et de la variance de l'erreur d'estimation $\varepsilon(\mathbf{x})$ nécessite l'introduction de relations de covariance s'appuyant sur la fonction de corrélation R qui dans notre cas sera la fonction Matérn [14] choisie avec $\nu = 3/2$. Elles établissent des relations entre la fonction $Z(\mathbf{x})$ et ses dérivées (équations 4-7).

$$\forall (i, j) \in \llbracket 0, n_s \rrbracket, \forall (\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \in \mathcal{D}^2, \text{cov} \left[Z(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)}) \right] = \sigma^2 R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{ij} \quad (4)$$

$$\text{Cov} \left[\frac{\partial Z}{\partial x_k}(\mathbf{x}^{(i)}), Z(\mathbf{x}^{(j)}) \right] = -\sigma^2 \frac{\partial R}{\partial x_k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{kij} \quad (5)$$

$$\text{Cov} \left[Z(\mathbf{x}^{(i)}), \frac{\partial Z}{\partial x_k}(\mathbf{x}^{(j)}) \right] = -\sigma^2 \frac{\partial R}{\partial x_k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{kij} \quad (6)$$

$$\text{Cov} \left[\frac{\partial Z}{\partial x_k}(\mathbf{x}^{(i)}), \frac{\partial Z}{\partial x_l}(\mathbf{x}^{(j)}) \right] = -\sigma^2 \frac{\partial^2 R}{\partial x_k \partial x_l}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 c_{kilj} \quad (7)$$

La résolution de ce problème de minimisation conduit à l'obtention de l'estimateur suivant :

$$\tilde{Y}(\mathbf{x}^{(0)}) = \hat{\beta}_c + \mathbf{c}_{c0}^T \mathbf{K}_c^{-1} (\mathbf{Y}_{sc} - \mathbf{X}_c \hat{\beta}_c) \quad (8)$$

$$\begin{aligned} \hat{\beta}_c &= (\mathbf{X}_c^T \mathbf{K}_c^{-1} \mathbf{X}_c)^{-1} \mathbf{X}_c^T \mathbf{K}_c^{-1} \mathbf{Y}_{sc} & \mathbf{c}_{c0} &= [c_{01} \quad c_{02} \quad \cdots \quad c_{0n_s} \quad c_{110} \quad c_{210} \quad \cdots \quad c_{n_d n_s, 0}] \\ \text{où } \mathbf{X}_c &= [1, 1, \dots, 1, 0, 0, \dots, 0]^T & \mathbf{K}_c &= \begin{bmatrix} \mathbf{C} & \mathbf{C}_d \\ \mathbf{C}_d^T & \mathbf{C}_{dd} \end{bmatrix} \end{aligned}$$

Les vecteur et matrice \mathbf{c}_{c0} et \mathbf{K}_c sont obtenus respectivement en calculant les corrélations du point $\mathbf{x}^{(0)}$ aux points échantillonnés et en calculant les corrélations entre les sites. La matrice \mathbf{C} est la matrice de corrélation classiquement utilisée pour le krigeage, la matrice \mathbf{C}_d est la matrice de corrélation liant les réponses et les gradients aux sites échantillonnés et la matrice \mathbf{C}_{dd} est la matrice de corrélation liant les

gradients aux sites échantillonnés. Le vecteur \mathbf{Y}_{sc} contient quant à lui les réponses et gradients calculés aux sites échantillonnés.

Les paramètres de la fonction de corrélation (fonction gaussienne ou Matérn dans notre cas) et la variance du processus stochastique sont estimés à l'aide d'une méthode de maximisation de la vraisemblance [15].

Par ailleurs, l'utilisation de telles stratégies d'approximation permet également d'obtenir un estimateur de la qualité de prédiction du métamodèle. Ce dernier peut-être vu comme un estimateur de l'erreur des moindres carrés et être formulé de la manière suivante :

$$\text{MSE} \left[\tilde{Y}(\mathbf{x}^{(0)}) \right] = \sigma^2 \left[1 - [\mathbf{1}_{n_s}^T \quad \mathbf{r}_{c0}^T] \begin{bmatrix} \mathbf{0} & \mathbf{X}_c^T \\ \mathbf{X}_c & \mathbf{C}_c \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{1}_{n_s} \\ \mathbf{r}_{c0} \end{bmatrix} \right] \quad (9)$$

4.4 Exemple d'approximation

Afin d'illustrer les performances liées à l'ajout des gradients dans l'approximation par krigeage, une fonction d'une variable¹ est utilisée. Les réponses et dérivées sont calculées en 5 points tirés aléatoirement et ces informations sont exploitées pour construire les métamodèles de krigeage et cokrigeage (cf. FIG. 5). Par ailleurs les intervalles de confiance sont calculés de la manière suivante :

$$IC^\pm(\mathbf{x}) = \tilde{Y}(\mathbf{x}) \pm j \sqrt{\text{MSE} \left[\tilde{Y}(\mathbf{x}^{(0)}) \right]} \quad \text{où } j = \begin{cases} 1 & \text{pour IC à 68\%} \\ 2 & \text{pour IC à 95\%} \\ 3 & \text{pour IC à 99\%} \end{cases}$$

L'ajout des gradients permet une sensible amélioration de la qualité de prédiction rendant le métamodèle

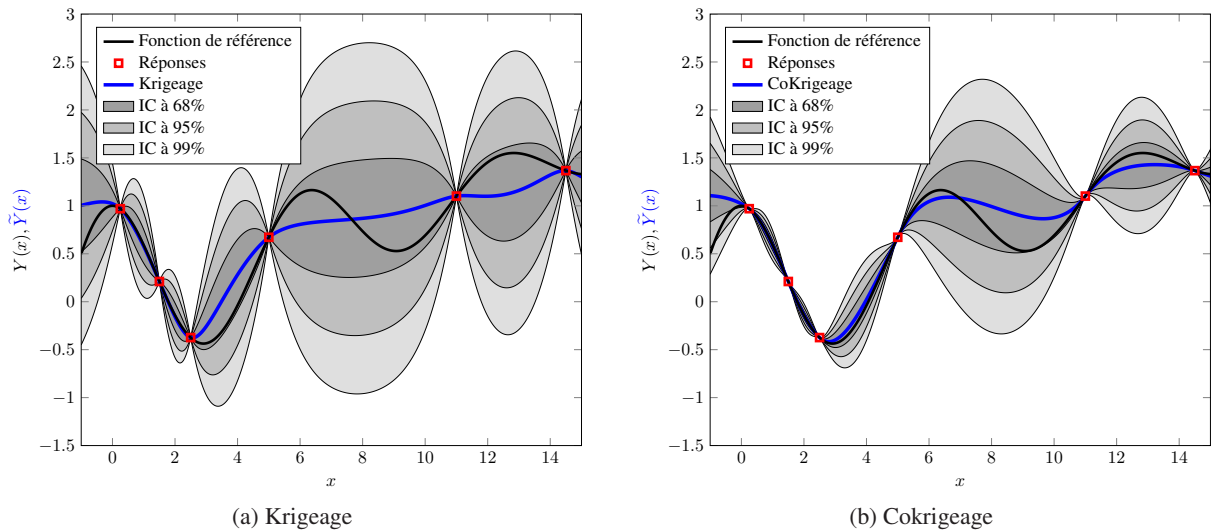


Fig. 5 – Krigage et cokrigeage ordinaires, et intervalles de confiance

interpolant pour les dérivées (en plus du fait qu'il soit interpolant pour les réponses). Des résultats relatifs à l'usage de ce type de métamodèle à gradients peuvent être trouvés dans [16] et [17].

4.5 Enrichissement

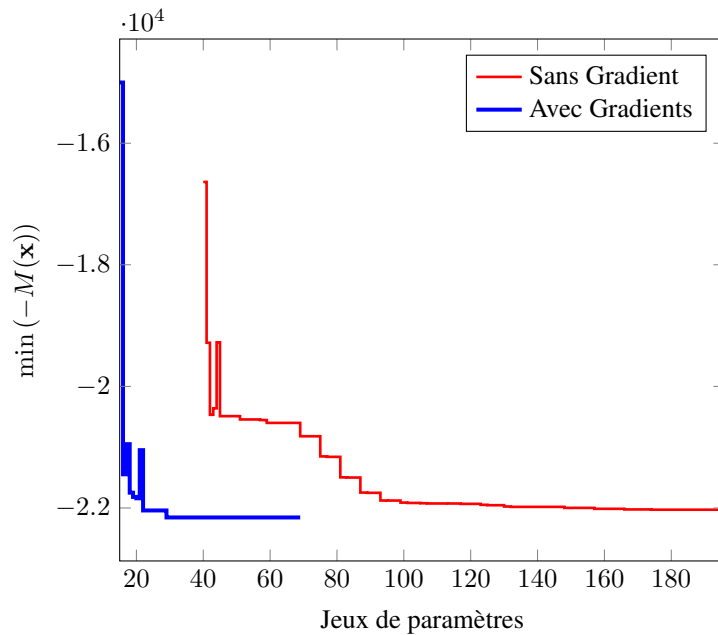
Lors de la phase de recherche du minimum global, la méthode pourrait se limiter simplement à la construction de l'approximation et à la recherche de l'optimum sur cette dernière. Cependant, dans l'optique où l'on souhaite minimiser le nombre d'appels au modèle mécanique, il est nécessaire de parvenir à localiser le minimum grâce au métamodèle avec un nombre limité d'appels au solveur. La stratégie de tirage ne fournissant qu'un échantillonnage a priori, il est nécessaire de trouver une méthode afin d'ajouter *intelligemment* de nouveaux jeux de paramètres afin de parvenir à localiser au mieux le minimum.

1. $y(x) = \exp(-x/10) \cos(x) + x/10$

La stratégie employée s'appuie sur l'algorithme EGO [18] : dans un premier temps, le métamodèle est construit en utilisant n_s points, puis, en utilisant les informations fournies par le métamodèle, un critère d'enrichissement nommé *amélioration attendue* (expected improvement) est élaboré. La recherche du maximum de ce critère dans l'espace des paramètres fournit alors un nouveau point à ajouter au métamodèle qui est alors reconstruit en intégrant ce nouveau point, la réponse et le gradient qui lui sont associés. Ce processus peut alors être répété jusqu'à convergence.

5 Optimisation

La démarche d'optimisation complète introduite au début de l'article est alors mise en oeuvre sur le cas test mécanique proposé dans la partie 3.3. L'objectif est ici de trouver le jeu de paramètres maximisant le couple transmissible [7]. Afin d'illustrer les performances de la démarche multiniveaux proposée, 2 approches sont considérées : elles se basent sur l'approche à deux niveaux proposée dans ce papier en utilisant tout d'abord un métamodèle sans gradient puis un métamodèle capable de les prendre en compte. La figure 6a présente le minimum de l'inverse du couple maximum transmissible au fur et à mesure du processus d'optimisation basé sur le métamodèle. Le premier point de chaque courbe correspond au minimum approché fourni par le métamodèle construit avec les premiers jeux de paramètres obtenus grâce à un tirage LHS (soit 15 jeux de paramètres pour le cokrigeage et 30 pour le krigeage). La phase d'enrichissement incorpore 8 nouveaux jeux de paramètres au métamodèle de cokrigeage et 5 à celui de krigeage.



(a) Minimums atteints par la stratégie à deux niveaux

Gradients	Sans	Avec
Nombre d'appels	197	180
Temps CPU total	204s	155s
Temps CPU moyen	1s	0.9s
Temps CPU mini	0.2s	0.2s
Temps CPU maxi	11s	9s
Nb. iterations total	4631	2996
Nb. itérations moyen	24	17
Nb. itérations mini	5	5
Nb. itérations maxi	266	196
Gain	10.6	22

(b) Caractéristiques des calculs

Fig. 6 – Maximisation du couple maximum transmissible par l'assemblage fretté

La prise en compte des gradients assure une réduction importante du coût de calcul tout en gardant une qualité d'approximation suffisante pour permettre au métamodèle de fournir un minimum approché suffisamment proche du véritable minimum global.

6 Conclusion

L'ajout des gradients lors de la construction du métamodèle permet l'obtention d'une approximation globale de bonne qualité tout en permettant une réduction importante des coûts de calcul associés à la résolution de problèmes d'assemblage qui, par définition, sont fortement non linéaires. Ce gain est possible grâce à l'emploi de la stratégie multiparamétrique. Par ailleurs, une optimisation globale sur le métamodèle préalablement enrichi assure l'obtention d'un minimum approché permettant une localisation plus

fine du minimum réel et ce, une nouvelle fois avec un coût de calcul fortement réduit. Le constat est alors double, la vitesse de convergence de la stratégie proposée permet de localiser le minimum d'une fonction coût en un nombre d'appels au solveur nettement plus faible par rapport au cas où un algorithme de recherche globale serait employé et ces appels au solveur sont obtenus de manière peu coûteuse grâce l'emploi de la stratégie multiparamétrique. À l'exemple 2D-axisymétrique proposé ici, viendront par la suite s'ajouter de nouveaux exemples plus complexes (du point de vue de la géométrie mais également du point de vue du nombre de paramètres pris en compte).

Remerciements

Ces travaux ont été effectués dans le cadre du projet ANR-08-COSI-007-10, OMD2 financé par l'Agence Nationale de la Recherche.

Références

- [1] Gordon M. Robinson and Andy J. Keane. A case for multi-level optimisation in aeronautical design. *Aeronautical Journal*, 103(1028):481–485, 1999.
- [2] Pierre Ladevèze. Sur une famille d'algorithmes en mécanique des structures. *Comptes Rendus de l'Académie des Sciences*, 300(2):41–44, 1985.
- [3] Pierre Ladevèze. *Nonlinear computational structural mechanics : new approaches and non-incremental methods of calculation*. Mechanical Engineering Series. Springer Verlag, 1999.
- [4] Laurent Champaney. *Une nouvelle approche modulaire pour l'analyse d'assemblages de structures tridimensionnelles*. PhD thesis, Ecole Normale Supérieure de Cachan, 1996.
- [5] Pierre-Alain Boucard and Laurent Champaney. A suitable computational strategy for the parametric analysis of problems with multiple contact. *International Journal for Numerical Methods in Engineering*, 57(9):1259–1281, 2003.
- [6] Bruno Soulier and Pierre-Alain Boucard. A multiparametric strategy for the large-scale multilevel optimization of structural assemblies. In *8th World Congress on Structural and Multidisciplinary Optimization, Lisbon, Portugal*, 2009.
- [7] Bruno Soulier and Pierre-Alain Boucard. A multiparametric strategy for the two step optimization of structural assemblies. *Structural and Multidisciplinary Optimization*, 47(4):539–553, 2013.
- [8] Michael D. McKay, William J. Conover, and R.J. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.
- [9] Daniel G. Krige. A statistical approach to some mine valuation and allied problems on the Witwatersrand. Master's thesis, University of the Witwatersrand, 1951.
- [10] Georges Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.
- [11] Jerome Sacks, Susannah B. Schiller, and William J. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.
- [12] H. Wackernagel. *Multivariate geostatistics : an introduction with applications*. Springer Verlag, 1995.
- [13] Max D. Morris, Toby J. Mitchell, and Donald Ylvisaker. Bayesian design and analysis of computer experiments : use of derivatives in surface prediction. *Technometrics*, 35(3):243–255, 1993.
- [14] B. Matérn. *Spatial Variation (Lecture Notes Statist. 36)*. Springer, Berlin, 1960.
- [15] K.V. Mardia and R.J. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135, 1984.
- [16] Luc Laurent, Pierre-Alain Boucard, and Bruno Soulier. Generation of a cokriging metamodel using a multiparametric strategy. *Computational Mechanics*, 51(2):151–169, February 2013.
- [17] Luc Laurent, Pierre-Alain Boucard, and Bruno Soulier. A dedicated multiparametric strategy for the fast construction of a cokriging metamodel. *Computers & Structures*, 124(0):61–73, 2013. Special Issue : KRETA.
- [18] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.