

Optimisation structurale par interpolation séquentielle des contraintes du programme mathématique

N. ABBOUD^a, A. KHAMLI^a, P. DUBUJET^b

a. Laboratoire AMS, Faculté des Sciences de Tétouan, BP. 2121 M'Hannech, Tétouan 93002, Maroc

b. LTDS UMR 5513, ENISE, 77 Rue Jean Parot, Saint-Etienne 42042, France

Résumé :

Le problème d'optimisation structurale se formule comme un programme mathématique non linéaire et non convexe. Ce problème est très difficile à résoudre dans la pratique car la fonction objectif et les contraintes sont des fonctions implicites des variables de conception de base. Les méthodes numériques utilisées actuellement pour le résoudre sont fondées sur des techniques d'évolution ou de sensibilités. Elles proposent de coupler un algorithme d'optimisation avec un code de calcul structural de type éléments finis par exemple. Ces méthodes sont lourdes, elles sont sujettes aux erreurs numériques et n'intègrent pas automatiquement la faculté de globalisation. Nous proposons dans le cadre de ce travail un nouvel algorithme conçu à partir de la notion de méta-modèle local. Une séquence de méta-modèles est ainsi construite selon un processus itératif utilisant les interpolations de la fonction objectif et des contraintes sur des domaines qui s'emboîtent et qui tendent vers un voisinage immédiat de l'optimum. Cet algorithme est validé dans le cas d'une structure élastique linéaire formée par une poutre continue à deux travées. Sa convergence a été analysée en fonction du choix des fonctions d'interpolation utilisées.

Abstract :

The structural optimisation problem has the general form of a nonlinear and non convex mathematical program. This problem is hard to tackle because the objective function and the constraints are implicit functions of the basic design variables. Methods of solution that exist actually are either evolution or sensitivity based techniques. They consist in coupling an optimisation strategy with a structural code such as finite element software. These methods are time consuming; they are sensitive to numerical errors and lack the faculty of automatically guarantying globalisation. A new algorithm is proposed in this work which is based on the concept of local metamodel approximation. A sequence of metamodels are constructed according to an iterative process, by using interpolations of the objective function and the constraints on embedded domains that tend to an immediate neighbourhood of the optimum. Validity of this algorithm was assessed in the particular case of a linear elastic structure having the form of a continuous beam with two spans. Analysis of its convergence was discussed in terms of the used interpolating functions.

Mots clefs : optimisation, structures, programme mathématique non linéaire, interpolation

1 Introduction

Lorsque la forme de la structure est donnée, l'objectif de l'optimisation structurale consiste à déterminer les valeurs de n variables de conception $x = \{x_1, x_2, \dots, x_n\}$ où x_i décrit en général des propriétés géométriques ou mécaniques: aires des sections des barres, épaisseurs des plaques, modules d'Young des matériaux, etc...

Les variables de conception doivent minimiser une fonction objectif de la forme $f(x) = f(x_1, x_2, \dots, x_n)$ où f peut par exemple être le poids, la section de coffrage, le coût de réalisation, ou toute autre fonction associée à un critère de coût relatif à la conception, la réalisation, la maintenance ou la durabilité.

Les variables de conception doivent aussi satisfaire à un ensemble de m contraintes de conception de la forme: $g(x) = g_i(x_1, x_2, \dots, x_n) \leq 0$ avec $i = 1, \dots, m$.

Les équations des contraintes se réfèrent en général aux contraintes et aux déplacements admissibles eu égard à la réglementation et aux capacités intrinsèques des matériaux. Elles peuvent aussi être associées à des critères de stabilité, des critères vibratoires ou de fatigue. Dans la suite, seules des contraintes portant

exclusivement sur les flèches admissibles seront considérées pour présenter le développement d'une nouvelle approche d'optimisation structurale basée sur une approximation itérative du problème qui est obtenue par interpolation de la fonction objectif et des contraintes et par le choix d'une fonction mérite assurant la globalisation.

Normalement, les contraintes du programme mathématique sont évaluées en utilisant la solution du problème aux limites décrivant la déformation de la structure. Seul le problème statique en petites déformations est considéré dans la suite et seuls des treillis de poutre seront considérés bien que la démarche puisse être généralisée sans aucun problème aux autres types de structures.

Les problèmes statiques en calcul des structures sont formulés en écrivant les équations d'équilibre et les équations de compatibilité. En général, celles-ci sont obtenues à l'aide du principe des puissances virtuelles qui s'écrit sous la forme

$$\int_{\Omega} \sigma \delta \varepsilon d\Omega = \int_{\Omega} f_v \delta u d\Omega + \int_{\partial\Omega} f_s \delta u d\Gamma \quad (1)$$

où δu est un champ de déplacements virtuels, $\delta \varepsilon$ la déformation virtuelle qui en résulte, $\delta \sigma$ les contraintes internes dans la structure, et f_v et f_s désignent les chargements appliqués respectivement dans le volume du domaine et sur sa frontière.

Le principe des puissances virtuelles fournit une formulation variationnelle naturelle du problème, utilisée au dans le contexte de la méthode des éléments finis pour calculer les déplacements et en déduire ensuite les contraintes mécaniques.

Remarquons que dans la pratique, en plus du problème délicat qui relève du choix des variables de conception, on ne sait pas toujours écrire de manière explicite les contraintes du problème d'optimisation tel que par exemple celles s'exprimant en fonction des déplacements. L'expression explicite d'un déplacement, dans le cas d'un problème continu, en fonction des charges appliquées et des autres paramètres de conception ne peut pas être obtenue sans l'inversion du système discret assemblé associé à toute la structure. Ainsi, même si l'on sait calculer numériquement point par point la solution en termes des déplacements, on ne sait pas exprimer sous forme analytique explicite les déplacements et par suite les contraintes en fonction des données. Les fonctions f et g du programme mathématique ne peuvent donc pas être explicitées et on ne peut qu'évaluer point par point les valeurs qu'elles prennent une fois la solution est calculée. Ceci ne permet pas de résoudre directement le programme mathématique dans une approche différentielle directe car les dérivées des fonctions f et g sont nécessaires. Partant de ce constat, deux approches ont été envisagées. Il y a d'une part les méthodes qui utilisent le couplage d'un module de calcul éléments finis avec un algorithme d'optimisation, et d'autre part des méthodes utilisant une approximation du programme mathématique obtenue au moyen de l'interpolation de la fonction objectif et des contraintes afin de construire des surfaces de réponse. La première approche est très lourde dans la pratique car le volume des calculs est important. La deuxième approche s'est révélée intéressante mais le problème de la convergence de la méthode est soulevé.

Dans le cas particulier de l'optimisation structurale d'une poutre continue vis-à-vis de l'état limite de service des déformations, il existe la possibilité d'écrire de manière explicite le programme mathématique en termes des charges et des paramètres géométriques par application par exemple du théorème des trois moments. Le programme mathématique obtenu est cependant non convexe et de nature fortement non linéaire.

La méthodologie s'appuyant sur l'interpolation du programme mathématique a été présentée dans [1]. Elle a pour objectif d'offrir un cadre général permettant de pratiquer l'optimisation des structures. La stratégie de résolution du problème d'optimisation est simple et peut être facilement implémentée dans n'importe quel environnement de calcul. Sa généralisation à d'autres domaines comme les problèmes de mécanique des fluides ou de thermique peut être envisagée sans difficultés.

Le grand problème qui se manifeste quand on veut mettre en œuvre cette stratégie est la convergence de la solution obtenue vers la solution exacte du problème [2]. En d'autres termes, quelle est l'influence du mode et du domaine d'interpolation sur la précision de la solution. Nous envisageons dans le cadre de ce travail d'approfondir cet aspect de convergence et nous proposons pour cela une interpolation dynamique du programme mathématique. Nous introduisons un ingrédient de globalisation permettant au cours des itérations de s'assurer toujours de la possibilité de poursuivre la solution optimale (globale).

2 Méthodologie d'optimisation structurale par interpolation

Les étapes majeures de la méthodologie suivie pour formuler le problème approché d'optimisation structurale au moyen de la technique d'interpolation sont la construction du problème approché et sa résolution.

Le problème approché est construit de manière explicite à l'aide de la technique d'interpolation. Supposons que les variables de conception au nombre de n s'écrivent $X = [x_1, x_2, \dots, x_n]^t$. En fixant le domaine de ces variables et en choisissant un ensemble fini de points dans ce domaine, on peut interpoler les résultats obtenus par calcul éléments finis afin de définir une approximation de la fonction objectif et des contraintes. L'interpolation peut selon le cas considéré être polynomiale, à base de splines ou de fonctions trigonométriques. Le schéma d'interpolation doit être adapté au problème considéré: choix du type d'interpolation, choix du domaine d'interpolation et choix des points d'interpolation dans ce domaine. A titre d'exemple, considérons deux bases génératrices de fonctions contenant k fonctions destinées à interpoler la fonction objectif et les contraintes et qui s'écrivent $\{p_1(x), \dots, p_k(x)\}$ et $\{q_1(x), \dots, q_k(x)\}$. On peut donc développer les fonctions f et g sous la forme suivante

$$f(x_1, \dots, x_n) \approx \tilde{f}(x_1, \dots, x_n) = \sum_{j=1}^{np} c_j p_j(x_1, \dots, x_n) \quad (2)$$

$$g_i(x_1, \dots, x_n) \approx \tilde{g}_i(x_1, \dots, x_n) = \sum_{j=1}^{nq} d_{ij} q_j(x_1, \dots, x_n) \quad (3)$$

avec \tilde{f} et \tilde{g} qui représentent les approximations de la fonction objectif et des contraintes du problème.

Il y a donc $(1+m)k$ constantes inconnues c_j et d_{ij} qui doivent être identifiées.

Afin d'estimer ces constantes d'interpolation, on fait varier les variables de conception dans un domaine convenable. Pour chaque point de conception, il est possible de calculer la solution associée par résolution des équations du problème de calcul des structures en utilisant la méthode des éléments finis. Il sera possible alors d'exprimer la fonction objectif et les contraintes pour ce point. Il suffit alors de générer un nombre, N , suffisant de solutions associées à des points de conception judicieusement choisis afin d'écrire un nombre d'équations suffisant permettant par inversion de trouver une estimation des constantes c_j et d_{ij} . Celles-ci seront obtenues par résolution au sens des moindres carrés des équations

$$\begin{bmatrix} \tilde{f}_1 \\ \vdots \\ \tilde{f}_N \end{bmatrix} = \begin{bmatrix} p_1(X_1) & \cdots & p_k(X_1) \\ \vdots & \ddots & \vdots \\ p_1(X_N) & \cdots & p_k(X_N) \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \tilde{g}_{i1} \\ \vdots \\ \tilde{g}_{iN} \end{bmatrix} = \begin{bmatrix} q_1(X_1) & \cdots & q_k(X_1) \\ \vdots & \ddots & \vdots \\ q_1(X_N) & \cdots & q_k(X_N) \end{bmatrix} \begin{bmatrix} d_{i1} \\ \vdots \\ d_{ik} \end{bmatrix} \quad (5)$$

Afin d'augmenter la précision au niveau de l'estimation des constantes d'interpolation, il est utile de faire intervenir un nombre de points, N , qui dépasse largement le strict minimum k . Le fait de disposer alors de plus d'équations que d'inconnues permet de mieux rendre compte des effets des différents bruits et des erreurs qui sont présents dans le problème discrétisé.

A l'aide des interpolations sous forme de surface de réponse de la fonction objectif et des contraintes, le problème d'optimisation initial qui est implicite

$$\begin{cases} \min f(X) \\ g_i(X) \leq 0 \quad i = 1, \dots, m \end{cases} \quad (6)$$

est remplacé par le problème d'optimisation approché explicite suivant

$$\begin{cases} \min \tilde{f}(X) \\ \tilde{g}_i(X) \leq 0 \quad i = 1, \dots, m \end{cases} \quad (7)$$

Pour résoudre ce dernier problème, on peut utiliser n'importe quelle technique de minimisation ayant fait ses preuves dans la résolution des programmes non linéaires avec contraintes. Dans ce travail, nous utilisons la méthode de programmation séquentielle quadratique (SQP).

Rien ne permet a priori de dire que le problème approché admet une solution \tilde{X} qui est suffisamment proche

de celle du problème initial X , dans le cas d'une interpolation statique obtenue par les équations (4) et (5) lorsqu'elle est effectuée en une seule étape. Nous proposons d'adapter la méthodologie d'approximation précédente, en considérant au lieu de l'interpolation statique un processus d'interpolation dynamique dans lequel le domaine d'interpolation est forcé d'être de plus en plus dans un voisinage immédiat de la solution au fur et à mesure que le nombre des itérations d'interpolation augmente. Nous construisons ainsi des domaines d'interpolation qui s'emboîtent et dont la taille se réduit d'un pas d'itération à un autre avec le facteur de $1/2^n$ où n est la dimension du problème. L'idée consiste à subdiviser le domaine polyédrique initial en 2^n polyèdres réguliers qui partitionnent le polyèdre initial. Sur chacun des sous polyèdres, on construit un problème approché par la technique d'interpolation et on résout les 2^n programmes mathématiques ainsi construits. On compare leurs solutions et on choisit le domaine où la solution converge et minimise la fonction objectif. Ce nouveau domaine permet d'itérer le processus et continuer la descente de manière récursive vers la solution optimale.

La méthodologie proposée est certes de nature heuristique car nous n'avons pas montré mathématiquement la convergence de la solution obtenue par ce processus itératif vers la solution exacte (solution exacte du problème exact), mais les essais numériques que nous avons conduits nous ont permis de constater la convergence. Le critère de convergence que nous avons choisi est de type asymptotique et nous arrêtons les itérations dès que l'amélioration de la solution ne dépasse plus un seuil spécifié sur un paquet d'itérations.

Dans la suite nous détaillons la construction de l'algorithme récursif d'interpolation dynamique du problème d'optimisation dans le cas d'une poutre continue à une seule travée. Dans ce cas particulier, une formulation analytique explicite du problème d'optimisation initial existe. Ceci, nous permet de comparer la solution exacte avec la solution approchée obtenue suite à la convergence du processus dynamique d'interpolation.

3 Optimisation d'une poutre continue à deux travées

On considère la poutre continue homogène à deux travées 1 et 2 de longueurs respectivement L_1 et L_2 , montrée sur la figure 1. La section droite transversale de la poutre est supposée rectangulaire de largeur uniforme b et de hauteur constante par travée notée respectivement x et y . On suppose que le matériau constituant la poutre est élastique linéaire homogène de module d'Young E et que chaque travée est soumise à une force concentrée appliquée à mi-portée, d'intensité respectivement P_1 et P_2 .

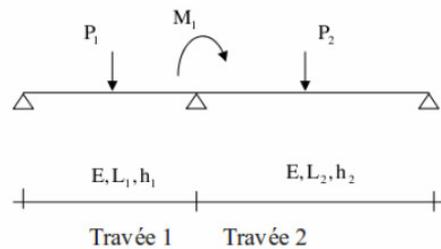


FIG. 1 – Poutre simplement appuyée sur ses deux extrémités et chargée par deux forces concentrées

Le problème consiste à déterminer les deux variables de conception x et y solutions du problème d'optimisation structurale associé à la minimisation du poids total de la poutre: $\rho b(L_1x + L_2y)$ où ρ est la densité du matériau de la poutre. Du fait que la quantité ρb est constante, la fonction objectif peut s'écrire sous la forme : $f(x, y) = L_1x + L_2y$. On considère le seul état limite de service vis-à-vis des déformations pour lequel on cherche à limiter les flèches des travées de la poutre aux valeurs admissibles fixées forfaitairement respectivement à δ_{a1} et δ_{a2} .

Ce problème admet une solution exacte qui s'obtient par résolution du programme non linéaire et non convexe suivant :

$$\begin{aligned}
 & \text{Minimiser } L_1x + L_2y \text{ sous les contraintes} \\
 & g_1(x, y) \leq 0; \quad g_2(x, y) \leq 0; \quad g_3(x, y) \leq 0; \quad g_4(x, y) \leq 0 \\
 & x_{\min} \leq x \leq x_{\max}; \quad y_{\min} \leq y \leq y_{\max}
 \end{aligned} \tag{8}$$

avec

$$\begin{aligned}
 g_1(x, y) &= 875P_1L_1^3y^3 + 2000P_1L_2L_1^2x^3 - 1125P_2L_1L_2^2x^3 - 32bEL_1x^3y^3 - 32bEL_2x^6 \\
 g_2(x, y) &= 2000P_2L_1L_2^2y^3 - 875P_2L_2^3x^3 - 1125P_1L_2L_2^2y^3 - 32bEL_1y^6 - 32bEL_2x^3y^3 \\
 g_3(x, y) &= -875P_1L_1^3y^3 - 2000P_1L_2L_1^2x^3 + 1125P_2L_1L_2^2x^3 - 32bEL_1x^3y^3 - 32bEL_2x^6 \\
 g_4(x, y) &= -2000P_2L_1L_2^2y^3 - 875P_2L_2^3x^3 + 1125P_1L_2L_2^2y^3 - 32bEL_1y^6 - 32bEL_2x^3y^3
 \end{aligned} \tag{8}$$

$$x_{\min} = \frac{L_1}{100}; x_{\max} = \frac{L_1}{5}; y_{\min} = \frac{L_2}{100}; y_{\max} = \frac{L_2}{5}$$

Lorsque les paramètres du problème sont par exemple les suivants: $E = 2 \times 10^{11} Pa$, $P_1 = 10 kN$, $P_2 = 20 kN$, $L_1 = 1 m$, $L_2 = 2 m$ et $b = 5 cm$, la commande *fmincon* de Matlab permet de trouver la solution optimale: $x_{exact} = 0.012$; $y_{exact} = 0.034$. Il faut souligner que du fait que le problème est non convexe, obtenir la solution exacte du problème exact par la commande *fmincon* a nécessité de procéder à de multiples initialisations. C'est seulement pour certaines initialisations telles que celles appartenant à l'ensemble $\{(0.0210, 0.0420); (0.0315, 0.0630); (0.0420, 0.0840)\}$ que la solution converge vers l'optimum. Pour d'autres initialisations, du genre $(0.0105, 0.0210)$ ou bien $(0.0630, 0.1260)$ l'algorithme ne converge pas. L'algorithme génétique de Matlab *ga* peut également être utilisé au lieu de *fmincon*. Mais, les essais que nous avons faits ont montré que cet algorithme peut ne pas converger ou bien converger vers une fausse solution. Lorsque *ga* converge le temps CPU est 40 fois supérieur à celui de la commande *fmincon*. Comme la fonction objectif est déjà donnée sous forme d'un polynôme linéaire en fonction des variables de base, on envisage d'approcher les contraintes (9), qui sont des polynômes incomplets de degré 6, par des interpolations polynomiales quadratiques en fonction de x et y . Le domaine initial d'interpolation doit être le plus large possible afin de contenir la solution du problème. On prend alors le rectangle $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ pour lequel l'approximation par la théorie des poutres est applicable. On effectue une subdivision du rectangle précédent en introduisant les points: x_{\min} , $x_{moy} = (x_{\min} + x_{\max})/2$, x_{\max} ; y_{\min} , $y_{moy} = (y_{\min} + y_{\max})/2$, y_{\max} . Pour construire les polynômes d'interpolation $\tilde{g}_i(x, y)$, on utilise les 9 sommets des sous rectangles obtenus par subdivision du rectangle initial conformément à la figure 2.

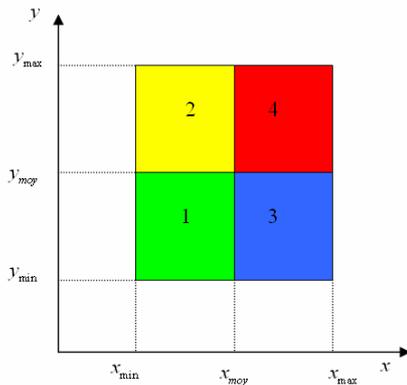


FIG. 2 – Points utilisés lors de l'interpolation des contraintes $\tilde{g}_i(x, y)$, $i = 1, \dots, 4$

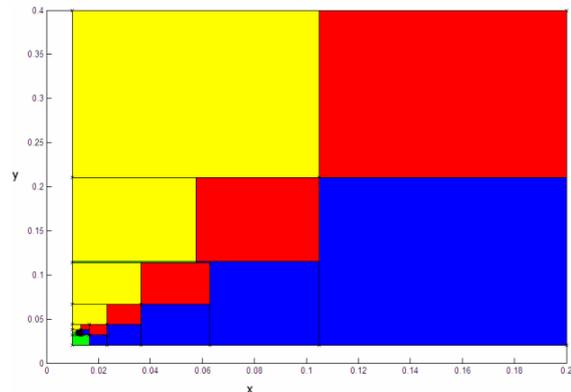


FIG. 3 – Séquence des rectangles imbriqués d'interpolation des contraintes au cours des itérations

La solution du problème approché construit sur chacun des domaines rectangulaires de la figure 2 a donné les quatre résultats suivants :

- Rectangle 1 : $\tilde{x} = 0.0391$, $\tilde{y} = 0.0796$;
- Rectangle 2 : $\tilde{x} = 0.0434$, $\tilde{y} = 0.2100$
- Rectangle 3 : $\tilde{x} = 0.1050$, $\tilde{y} = 0.6901$;
- Rectangle 4 : $\tilde{x} = 0.1050$, $\tilde{y} = 0.2100$

On compare entre elles les solutions obtenues en cas de convergence et on sélectionne, via une fonction de mérite, le domaine qui est sensé contenir la solution optimale permettant de réaliser le minimum global de la fonction objectif. La fonction de mérite que nous avons considérée utilise le barycentre des points solutions qui ont convergé affectés des inverses des valeurs que prend la fonction objectif en ces points. Le domaine

rectangulaire sélectionné est alors celui qui contient ce barycentre. En cas de divergence des quatre problèmes, donc impossibilité de définir un barycentre, on initialise le processus en augmentant localement le domaine rectangulaire sur lequel les divergences ont été constatées. Dans l'exemple considéré ici, la sélection du rectangle selon la fonction de mérite signalée a donné le rectangle 1. On peut ensuite itérer ce processus en partant du nouveau domaine rectangulaire obtenu, rectangle 1 pour la première itération. La figure 3 permet d'illustrer ce processus d'interpolation séquentielle.

Le graphe de la figure 4 illustre la convergence de la séquence des solutions approchées trouvées par ce processus itératif vers la solution exacte. On peut constater que la convergence se produit au bout seulement de 6 itérations. La deuxième itération qui donne des valeurs nulles correspond à une phase de changement du domaine d'interpolation.

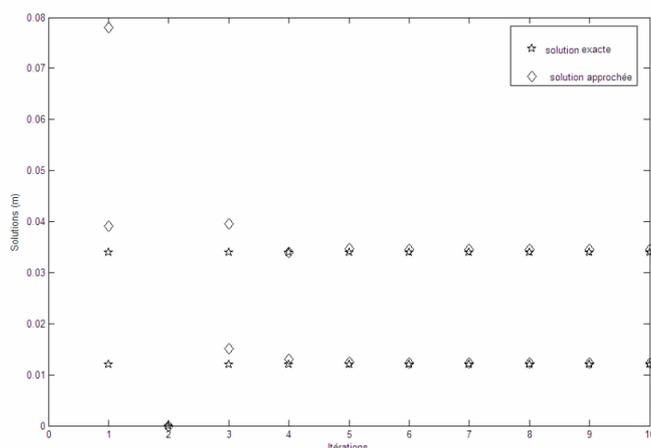


FIG. 4 – Processus de convergence de la solution approchée vers la solution exacte

Le tableau 1 présente les résultats d'une étude portant sur l'influence du choix du type d'interpolation sur la solution. Les polynômes choisis sont des polynômes complets à deux variables: polynôme linéaire, quadratique, cubique ou polynôme de degré 4. Le nombre de points d'interpolation est égal à 25, ce qui correspond au choix de 5 points discrets régulièrement espacés sur chaque axe.

Paramètre	Linéaire	Quadratique	Cubique	Degré 4
x	0.0129	0.0124	0.0124	0.0126
y	0.0317	0.0346	0.0348	0.0343
$f(x, y)$	0.0764	0.0816	0.0819	0.0812
Itération à la convergence	5	5	4	3
Temps CPU	2.94	3.39	3.87	1.778

TAB. 1 – Influence du choix du type d'interpolation

D'après le tableau, l'interpolation linéaire converge mais la précision est mauvaise. Les autres interpolations convergent avec une très bonne précision. Le nombre des itérations nécessaires à la convergence et le temps CPU sont comparables. Dans la pratique, c'est l'interpolation quadratique qui est appliquée car elle est moins lourde que les autres interpolations.

L'algorithme numérique que nous avons présenté dans ce travail a été testé dans d'autres cas plus complexes et la convergence a été observée [3].

References

- [1] Gil L., Andrew A., Blanco E. Integration methodology of different software tools for constrained optimization. *Computers & Structures*, 82, 1639-1647, 2004.
- [2] Khamlichi A., Abboud N., El Bakkali L., Limam A. Weight Optimisation of Structures under Displacement Limitations by Interpolating Constraints. *AJBAS* 4(7): 2081-2092, 2010.
- [3] Khamlichi A., Abboud N. Optimisation des structures par interpolation séquentielle. Editions Universitaires Européennes. ISBN: 978-613-1-54361-6, 2010.