

## Optimisation topologique par lignes de niveaux : mise en œuvre non intrusive

F. Bordeu<sup>1</sup>, J. Cortial<sup>1</sup>, C. Rey<sup>1</sup>

<sup>1</sup> Safran Tech, Safran SA, Magny-Les-Hameaux, {felipe.bordeu, julien.cortial, christian.rey}@safrangroup.com

**Résumé** — On considère la mise en œuvre de la méthode des lignes de niveaux pour l'optimisation topologique de pièces mécaniques. Afin de pouvoir utiliser les capacités de calcul haute-performance de solveurs d'éléments finis existants, on développe un démonstrateur logiciel faiblement intrusif. Cette contribution présente les choix d'implémentation effectués et les leçons à tirer de ces travaux.

**Mots clefs** — Optimisation topologique ; Lignes de niveaux ; Non-intrusif.

### 1. Introduction

L'optimisation topologique est une discipline d'optimisation de structures qui consiste à déterminer, au sein d'un espace de conception donné, la distribution de matériau la plus efficace possible telle que la solution obtenue respecte un certain nombre de contraintes mécaniques.

Ses caractéristiques intrinsèques la rendent très attractive pour les utilisateurs industriels. Premièrement, elle permet en effet une liberté de conception plus large que l'optimisation paramétrique voire l'optimisation de forme classique et facilite l'obtention des concepts non contraints de manière artificielle. Deuxièmement, son utilisation ne nécessite pas de disposer d'une géométrie préliminaire complète et détaillée d'une pièce à optimiser, mais seulement d'un cahier des charges fonctionnel.

Différentes méthodes d'optimisation topologique ont été développées dans la littérature scientifique, parmi lesquelles la méthode de densités variable (ou SIMP, acronyme de *Solid Isotropic Material with Penalization*) [1] et la méthode des lignes de niveau (souvent désignée par le terme anglais *level set method*) [2][3].

La méthode SIMP est utilisée dans l'immense majorité des solutions d'optimisation topologique disponible dans le commerce. Elle consiste à discrétiser le volume de conception (typiquement, par une méthode d'éléments finis) puis à introduire un champ de densité pouvant varier continûment entre 0 (absence de matière) et 1 (présence de matière). Ainsi, le problème d'optimisation est susceptible d'être traité par une approche de type gradient. La méthode SIMP tire son nom de la nécessaire pénalisation des densités intermédiaires, dont la présence est indésirable dans la solution finale, introduite dans la relation entre champ de densité et propriétés matériaux. L'avantage de cette méthode est sa simplicité conceptuelle et sa relative facilité de mise en œuvre au sein d'un solveur d'éléments finis. Dans le cas le plus élémentaire (minimisation de la compliance avec une contrainte de volume), la formulation du problème discrétisé est la suivante :

$$\min_{\rho} c(\rho) = \int_{\Omega} \varepsilon(u) \mathbf{K}(\rho) \varepsilon(u) dV \quad (1)$$

sous la contrainte

$$\int_{\Omega} \rho dV \leq V_s \quad (2)$$

Ici  $c$  est la compliance de la structure,  $\mathbf{K}$  l'opérateur de Hooke,  $\varepsilon$  la déformation,  $u$  le déplacement solution du problème mécanique et  $V_s$  la borne supérieure du volume de la pièce. Ici les deux intégrales

sont définies sur le volume de conception ( $\Omega$ ). Un désavantage de la méthode est l'interprétation de la solution finale. Malgré la présence d'une pénalisation, le champ final  $\rho$  comporte généralement des densités intermédiaires, et l'utilisateur doit choisir un seuil afin d'obtenir une interface nette et bien définie. Cette ambiguïté complique également la définition de contraintes de fabrication, souvent exprimées sous la forme de critères géométriques (distances, courbures, etc.) [4].

La méthode de lignes de niveau, qui fait l'objet du présent travail, ne souffre pas de cet inconvénient. Elle repose sur une description implicite de l'interface entre vide et matière, définie comme la ligne de niveau zéro d'un champ scalaire  $\varphi$ , typiquement initialisé comme la distance signée à la frontière. Ce champ peut alors être actualisé de manière itérative pour faire varier la forme de manière très libre, ce choix de paramétrage permettant des changements arbitraires de topologie.

Le problème d'optimisation pour la méthode de lignes de niveaux peut s'écrire sous la forme :

$$\min_{\varphi} c = \int_{\Omega(\varphi)} \varepsilon(u) K \varepsilon(u) dV \quad (3)$$

sous la contrainte,

$$\int_{\Omega(\varphi)} dV \leq V_s \quad (4)$$

Dans ce cas, les intégrales sont définies sur le volume effectif de la pièce  $\Omega$ , lui-même paramétré par le champ  $\varphi$ .

Pour mettre en œuvre des approches de type gradient, il est nécessaire de calculer la sensibilité des différents critères d'optimisation (objectif et contraintes) par rapport aux variables de conception ( $\rho$  pour la méthode SIMP et  $\varphi$  pour la méthode de lignes de niveaux). Pour les critères qui dépendent de la mécanique, comme par exemple la compliance, le solveur doit être capable de résoudre le problème adjoint en plus du problème direct.

## 2. Mise en œuvre

Le démonstrateur d'optimisation par lignes de niveaux possède les composants suivants :

1. Un solveur physique (ici, un solveur éléments finis)
2. Un module de gestion et d'évolution du champ distance  $\varphi$  (ou machinerie *level set*)
3. Un moteur d'optimisation qui pilote les itérations successives

La Figure 1 montre les différentes interactions entre ces trois sous-systèmes.

Un aspect important afin d'atteindre une intrusivité minimale est la capacité du solveur physique (ici le solveur élément finis) de calculer une solution ( $u$ ), la valeur de la fonction coût, et la sensibilité par rapport au paramètre de conception (ou sortir l'information nécessaire afin d'évaluer la sensibilité en post traitement). Ceci permet de confiner tout la complexité liée à la physique du problème à l'intérieur du code de calcul et ainsi d'utiliser, par exemple, les capacités de calcul parallèle disponibles.

Un autre point important est la séparation du module de gestion du champ  $\varphi$  (machinerie *level set*) et du moteur d'optimisation. Ceci permet d'utiliser différents algorithmes d'optimisation (méthode de la plus grande pente, Lagrangien augmenté...).

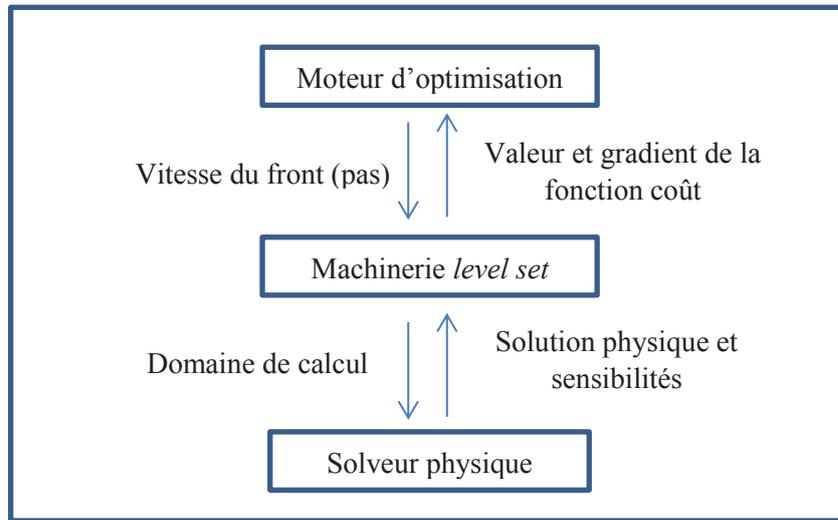


Figure 1 : Interaction entre les différents composants du démonstrateur

La principale difficulté réside dans la manière que les algorithmes d'optimisation classiques calculent un nouveau point. En général ceci est fait par une opération d'addition,

$$u_{new} = u_{old} + pv \quad (5)$$

Avec  $p$  le pas défini par l'algorithme d'optimisation et  $v$  la direction de descente. Dans notre cas un nouveau point est obtenu par la solution d'une équation de type Hamilton-Jacobi :

$$\frac{\partial \varphi}{\partial t} + v|\nabla \varphi| = 0 \quad (6)$$

Cette opération rend parfois difficile l'utilisation des algorithmes d'optimisation en boîte noire et nous oblige de changer l'API (interface de programmation) des algorithmes.

Enfin, l'algorithme d'optimisation topologique se déroule de la manière suivante (à partir d'un champ  $\varphi$  initial donné):

- 1) Calcul de la solution par le moteur physique pour une forme donné ( $\varphi$ ).
  - a. Evaluation des critères d'optimisation
  - b. Calcul des sensibilités
- 2) Calcul d'une direction de descente par la machinerie *level set*.
  - a. Définir le nouveau point  $\varphi$  comme la solution de l'équation d'Hamilton-Jacobi avec  $v$  (direction de descente) et  $p$  le pas définie par l'algorithme de minimisation.
- 3) Calcul de la nouvelle solution par le moteur physique et évaluation des quantités dérivées
- 4) Si la condition d'arrêt n'est pas satisfaite, revenir au point 2.

### 3. Résultats

L'ensemble des composants présentés précédemment ont été programmés en Python. Pour le moment, l'algorithme d'optimisation disponible est sur une méthode de gradient où les contraintes d'inégalité sont imposées par pénalisation.

La machinerie lignes de niveau est aussi programmée en Python. Elle repose sur une grille (un maillage structuré) et permet de faire toutes les manipulations courantes sur un champ de distance : initialisation, régularisation, résolution de l'équation Hamilton-Jacobi, sorties et post traitement.

Deux solveurs physiques sont disponibles. Le premier est un solveur 100% Python qui permet de faire de calculs de mécanique linéaire élastique isotrope sur un maillage structuré. Le deuxième solveur est la suite de calcul ZSet. L'intérêt est de pouvoir utiliser toutes les capacités proposées par ZSet dans un futur proche (calcul parallèle, calcul non linéaire). La Figure 2 montre le résultat du processus d'optimisation topologique d'une barre soumise à un effort de torsion.

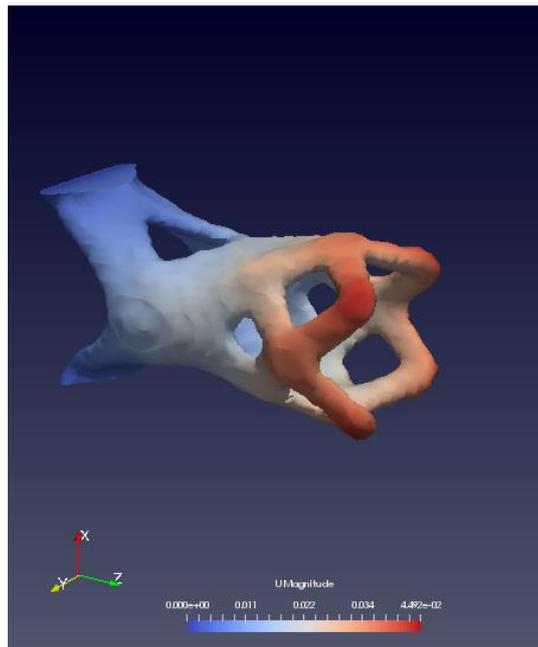


Figure 2 : Exemple de résultat de l'algorithme optimisation topologique par ligne de niveau (encastrement dans la zone bleu à gauche, force de torsion appliquée sur les quatre extrémités à droite)

### 4. Conclusion et perspectives

La mise en œuvre d'un algorithme d'optimisation topologique avec la technologie de lignes de niveau de manière faiblement intrusive impose un nombre important de contraintes. La principale est l'indépendance vis-à-vis du solveur physique, qui doit être capable de simuler la pièce et de calculer simultanément la sensibilité des critères d'optimisation par rapport aux variables de conception (champ  $\varphi$ ). L'utilisation d'un solveur physique parallèle va permettre d'exploiter plus facilement les moyens de calcul modernes et de diminuer considérablement la durée du processus d'optimisation.

## Références

- [1] M. Bendsøe et O. Sigmund: *Topology Optimization: Theory, Methods, and Applications*, Springer, 2004.
- [2] G. Allaire G., F. Jouve et A.-M. Toader: *A level-set method for shape optimization*, CR Acad. Sci. Paris, Série I, 334, 1125-1130, 2002.
- [3] G. Allaire G., F. Jouve et A.-M. Toader: *Structural optimization using sensitivity analysis and a level set method*, J. Comp. Phys., Vol 194/1, pp 363-393, 2004.
- [4] T. Abballe et al. : *RODIN project, topology optimization 2.0 ?*, Actes du congrès "Simulation" de la Société des Ingénieurs de l'Automobile (SIA), 2015.