

PISCO : Une plateforme de R&T industrielle en conception optimale

F. Bordeu¹, J. Cortial¹.

¹Safran Tech, M&S, Rue des Jeunes Bois, Châteaufort, 78114 Magny-Les-Hameaux, France felipe.bordeu@safrangroup.com

Résumé — L'optimisation topologique est couramment utilisée dans l'industrie pour la conception de pièces mécaniques et de nombreuses solutions logicielles disponibles dans le commerce sont disponibles à cet effet. Cependant, certains verrous techniques subsistent, comme par exemple la prise en compte des spécificités des procédés de fabrication additive, les problèmes multi-physiques (par exemple, mécanique, thermique et fluide) et la conception de structures à plusieurs échelles. PISCO (Plateforme Industrielle Safran pour la Conception Optimale) est un environnement de prototypage rapide pour l'optimisation topologie par la méthode des lignes de niveaux. Il est caractérisé par une architecture logicielle modulaire qui vise à découpler autant que possible le moteur d'optimisation topologique et le solveur physique, et par la mise en œuvre d'une méthode de lignes de niveau combinée à une procédure de remaillage afin de maintenir une discrétisation conforme à l'interface pendant le processus d'optimisation.

Mots clefs — Optimisation Topologique, Méthode des lignes de niveaux, Plateforme de prototypage.

1 Contexte

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 performante possible telle que la solution obtenue respecte un certain nombre de contraintes fonctionnelles.

Ses caractéristiques intrinsèques la rendent très attractive pour les utilisateurs industriels. Premièrement, l'optimisation topologique 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. Cet aspect offre également une bonne synergie avec les procédés de fabrication additive, qui ne sont généralement pas aussi restrictifs que les procédés traditionnels. Deuxièmement, la mise en œuvre de l'optimisation topologique ne nécessite pas de disposer d'une géométrie préliminaire complète et détaillée d'une pièce à concevoir, mais seulement d'un cahier des charges.

La plateforme PISCO vise à répondre aux objectifs suivants :

- Améliorer la compréhension des aspects numériques de l'optimisation topologique par la méthode des lignes de niveaux ;
- Accompagner la montée en maturité de technologies prometteuses, en particulier l'utilisation d'un maillage du domaine de calcul conforme à l'interface [1] ;
- Rendre possible l'utilisation de logiciels externes pour la résolution des problèmes physiques pour profiter de leurs capacités de calcul (méthodes numériques avancées, calcul haute-performance, etc.) ;
- Permettre le prototypage rapide de développements spécifiquement adaptés aux besoins du groupe Safran.

2 Architecture de la plateforme

Le langage Python a été utilisé du fait de sa simplicité d'accès, de la familiarité de nombreux développeurs avec ce langage, et de la disponibilité de bibliothèques comme NumPy¹ et SciPy² qui permettent d'obtenir un niveau de performance acceptable.

Toutefois, plusieurs routines ont été programmées en langage C++ pour obtenir des performances acceptables pour certaines opérations très spécifiques, comme par exemple l'assemblage des matrices élémentaires. Nous avons utilisé Cython³ pour l'interfaçage entre Python et C++.

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

- Un moteur d'optimisation qui pilote les itérations successives ;
- Une machinerie lignes de niveaux pour la gestion et l'évolution du champ distance ϕ ;
- Un solveur physique (ici, un solveur mécanique éléments finis).

La Figure 1 schématise les différentes interactions entre ces trois sous-systèmes. Chaque composant est suffisamment indépendant pour être modifié ou remplacé séparément. Par exemple, l'interface Z-set⁴ peut se substituer au solveur éléments finis intégré à la plateforme.

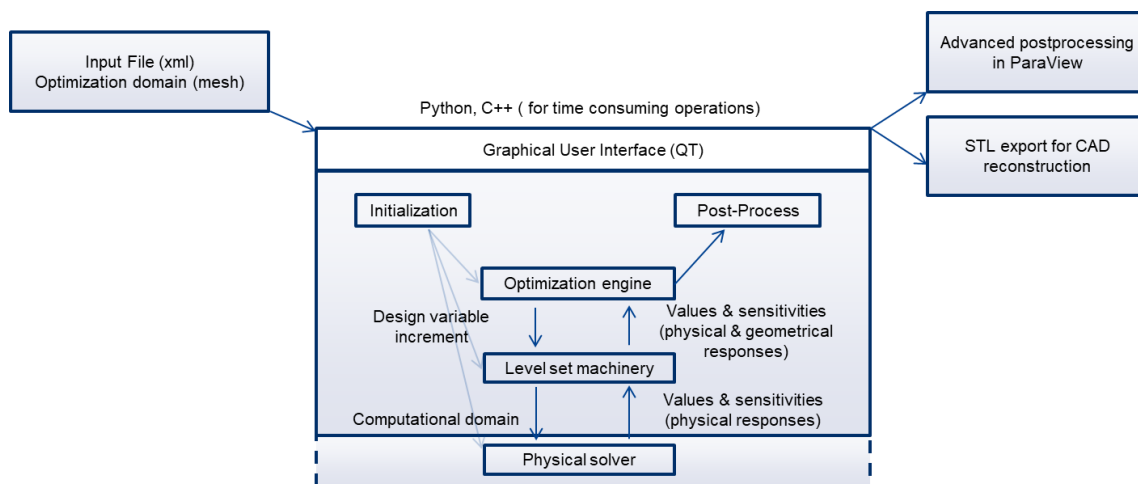


Figure 1 : Interactions entre les différents composants du démonstrateur

Dans la mesure du possible, les différentes briques ont été découplées pour obtenir une bibliothèque de composants modulaires, capable de s'adapter aux besoins futurs. Par exemple, il n'est pas nécessaire d'avoir la même discrétisation pour le problème éléments finis et pour le support de la variable de conception, c'est-à-dire, dans le cadre d'une méthode de lignes de niveaux, le champ de distance signé. De même, l'utilisateur peut décider d'interfacier un solveur physique différent et ainsi effectuer une optimisation topologique sur des problèmes de natures physiques différentes (par exemple, un problème couplé fluide/structure).

¹ <https://www.numpy.org>

² <https://scipy.org>

³ <https://cython.org>

⁴ <http://www.zset-software.com>

Une infrastructure de « plug-ins » permet à l'utilisateur d'intégrer ses propres fonctionnalités (lecture de maillages, algorithmes d'optimisations, problèmes d'optimisation, solveur physique, etc.) sans devoir modifier le code source existant. En particulier, la flexibilité obtenue permet d'isoler les développements expérimentaux des fonctionnalités plus matures disponibles dans la plateforme.

La méthode d'optimisation topologique par la méthode des lignes de niveaux repose sur une série d'opérations sur le champ de distance signée. La nature des algorithmes associés dépendent du type de support de calcul, en particulier maillage structuré fixe (Figure 2 : Ligne de niveau et interprétation sous la forme d'un champ de densité dans le cadre d'un maillage fixe.) ou non-structuré (Figure 3 : Champ de distance signée, domaine remaillée conforme à l'interface, maillages intérieur et extérieur pour la simulation physique.). De plus, dans le cas de maillages non-structurés épousant l'interface, une opération de remaillage doit être appliquée.

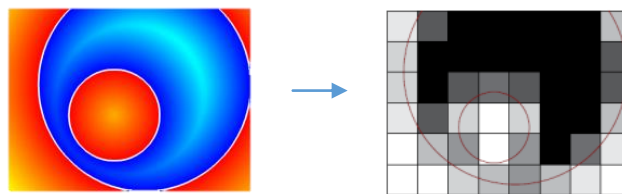


Figure 2 : Ligne de niveau et interprétation sous la forme d'un champ de densité dans le cadre d'un maillage fixe.

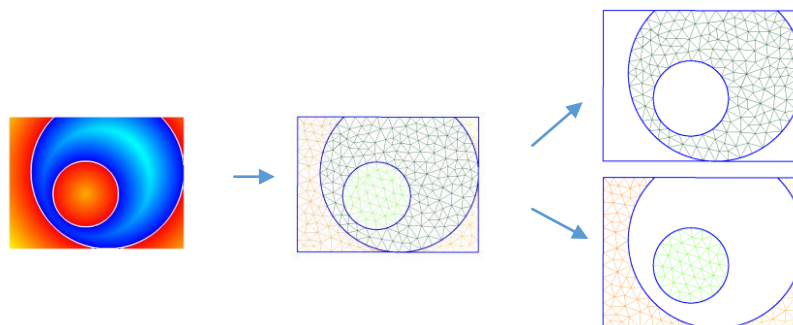


Figure 3 : Champ de distance signée, domaine remaillée conforme à l'interface, maillages intérieur et extérieur pour la simulation physique.

3 Application graphique interactive

Toutes les fonctionnalités présentées précédemment ont été intégrées dans une application graphique qui utilise un fichier d'entrée au format XML permettant d'initialiser l'ensemble des composants nécessaires au calcul. La Figure 4 montre la fenêtre principale de l'application. Cette dernière a été programmée entièrement en langage Python en utilisant les bibliothèques Qt⁵ pour la gestion de fenêtres et VTK⁶ pour la visualisation 3D.

⁵<https://www.qt.io/>

⁶<https://www.vtk.org/>

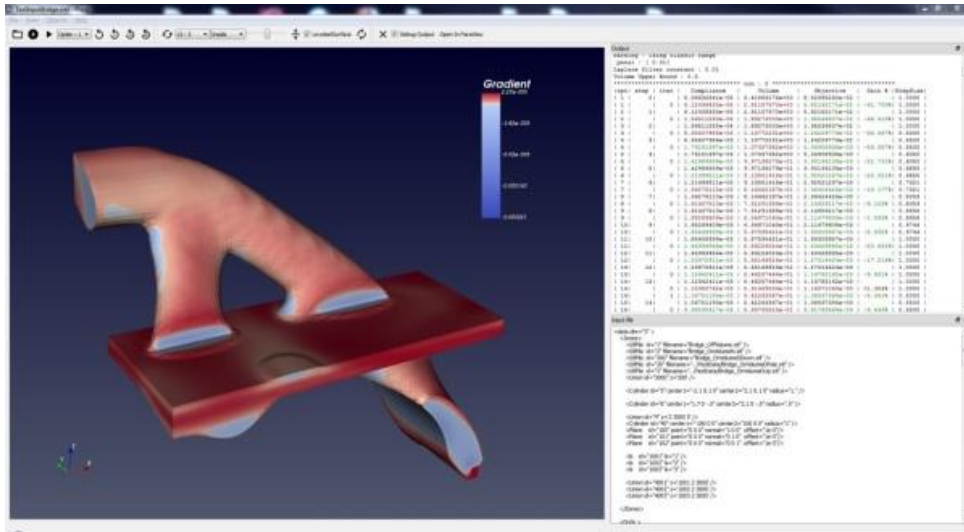


Figure 4 : Interface graphique utilisateur de PISCO

Références

- [1] C. Dapogny. Shape optimization, level set methods on unstructured meshes and mesh evolution. Thèse de doctorat, Université Pierre et Marie Curie - Paris VI. Récupéré sur <https://tel.archives-ouvertes.fr/tel-00916224>.