



HAL
open science

YETI – Nouvelles fonctionnalités d’optimisation de forme en IGA

Arnaud Duval, Thomas Elguedj

► **To cite this version:**

Arnaud Duval, Thomas Elguedj. YETI – Nouvelles fonctionnalités d’optimisation de forme en IGA. 16ème Colloque National en Calcul de Structures (CSMA 2024), CNRS; CSMA; ENS Paris-Saclay; CentraleSupélec, May 2024, Hyères, France. hal-04611069

HAL Id: hal-04611069

<https://hal.science/hal-04611069v1>

Submitted on 3 Dec 2024

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.

YETI – Nouvelles fonctionnalités d’optimisation de forme en IGA

A. Duval¹, T. Elguedj²

¹ Univ Lyon, CNRS, INSA Lyon, LaMCoS, UMR5259, 69621 Villeurbanne, France, arnaud.duval@cnrs.fr

² Univ Lyon, INSA Lyon, CNRS, LaMCoS, UMR5259, 69621 Villeurbanne, France, thomas.elguedj@insa-lyon.fr

Résumé — YETI (YET another Iga code) est une bibliothèque Python publiée sous licence LGPL et proposant une implémentation de l’analyse isogéométrique basée sur les NURBS et B-Splines. Cet article détaille les dernières fonctionnalités implémentées et qui sont dédiées à l’optimisation de forme de structures volumiques. La paramétrisation de la géométrie fait appel à une modélisation *immergée* afin de simplifier la modification de forme. Les gradients des quantités usuelles à optimiser (souplesse, contrainte, ...) sont exprimés de façon analytique, plus performant en terme de temps de calcul par rapport à une évaluation *via* des différences finies.

Mots clés — analyse isogéométrique, optimisation de forme, logiciel libre

1 Introduction

Les primitive géométriques telles que les NURBS ou B-Spline sont couramment utilisées en conception assistée par ordinateur (CAO). L’analyse isogéométrique (IGA) se base sur ces dernières pour décrire la solution d’un problème aux dérivées partielles et ainsi renforcer le lien CAO-calcul [1]. Ce lien fort entre modélisation géométrique et analyse mécanique permet de traiter efficacement certaines classes de problèmes en optimisation de forme où l’on cherche à minimiser des quantités telles que la souplesse d’une structure ou son niveau maximal de contrainte.

YETI est une bibliothèque *open source*, utilisable au sein d’un environnement Python, qui propose une implémentation de l’IGA. On s’attache ici à présenter les nouvelles fonctionnalités portant sur l’optimisation de forme de structures solides présentant des géométries de type congé ou raidisseur, modélisées via une méthode dite *immergée* qui simplifie la définition du problème d’optimisation. Dans une optique d’utilisation avec une méthode à gradient, les sensibilités au regard des variable de *design* sont exprimées de façon analytique.

2 Optimisation de forme en IGA

Les cas d’optimisation de forme pour les problèmes de mécanique auto-adjoints (élasticité linéaire avec des fonctionnelles faisant intervenir volume, souplesse, déplacement, contrainte et dont les variables de *design* sont directement liées aux points de contrôle de la géométrie NURBS) peuvent être traités par YETI. L’optimisation fait appel à deux modèles :

- un modèle d’optimisation qui fait le lien entre les variables de *design* du problème d’optimisation et la position des points de contrôle ;
- un modèle d’analyse, plus fin que le modèle d’optimisation, sur lequel la solution et les sensibilités sont calculées.

Ces deux modèles sont intrinsèquement liés via un opérateur de raffinement, qui est constant durant tout le processus d’optimisation. Un boucle type d’un algorithme d’optimisation à gradient (voir figure 1) met à jour le modèle d’optimisation en fonction des valeurs courantes des variables de *design*. La géométrie est ensuite raffinée en modèle d’analyse sur lequel sont évaluées la solution du problème mécanique et les sensibilités. Ces dernières sont ensuite agrégées sur le modèle d’optimisation, permettent une mise à jour des variables de *design* pour l’itération suivante.

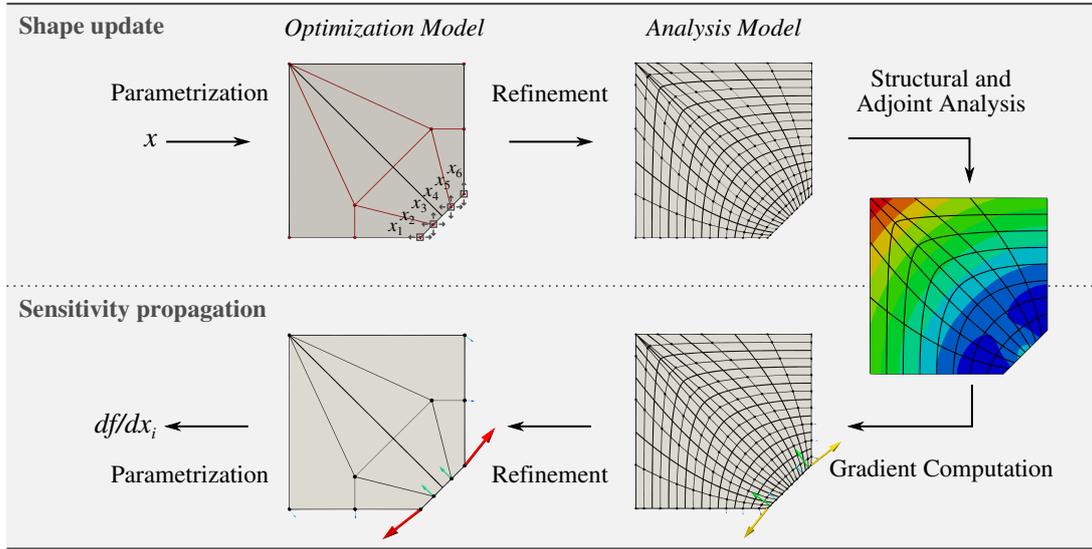


FIGURE 1 – Boucle type d’un algorithme d’optimisation de forme IGA utilisant une méthode à gradient (d’après [2]).

3 Modélisation *immergée*

Les structures présentant des géométries *filles* telles que des raidisseurs ou des congés peuvent présenter des difficultés lorsque l’on souhaite optimiser la forme de la géométrie *mère*. En effet, lorsque la forme de cette dernière est mise à jour au cours des itérations d’optimisation, il faut s’assurer que les géométries *filles* soient également modifiées afin d’assurer la connectivité de l’ensemble du modèle géométrique (voir figure 2).

La solution proposée au sein de YETI consiste à faire une modélisation dite *immergée* (voir [4]) des géométries *filles*. Ainsi la géométrie fille est immergée au sein d’un volume NURBS, lui-même issu d’une des faces de la géométrie *mère*. La géométrie réelle est ainsi obtenue via une composition de NURBS. Ce type de modélisation est illustré à la figure 3.

4 Calcul analytique des gradients

Comme démontré dans [2], lorsque l’on souhaite résoudre un problème d’optimisation dont le but est de minimiser une fonction objectif f , il est possible, dans le cadre de l’IGA, d’exprimer les gradients de façon analytique.

On définit une fonction objectif f , qui dépend des variables de *design* x et des variables d’état u :

$$f := g(x, u(x)) \quad (1)$$

où u est la solution d’un système linéaire $Ku = F$.

Les dérivées de la fonction objectif par rapport aux variables de *design* s’écrit :

$$\frac{df}{dx_i} = \frac{\partial g}{\partial x_i} + \frac{\partial g}{\partial u} \cdot \frac{du}{dx_i} \quad (2)$$

Le terme du/dx_i est inconnu et peut être obtenu par différenciation du problème linéaire :

$$\frac{du}{dx_i} = K^{-1} \left(\frac{\partial F}{\partial x_i} - \frac{\partial K}{\partial x_i} u \right) \quad (3)$$

Cette expression fait apparaître l’inverse de la matrice de raideur K^{-1} , ce qui implique la résolution d’un problème linéaire. On introduit alors un problème adjoint :

$$Ku^* = \frac{\partial f}{\partial u} \quad (4)$$

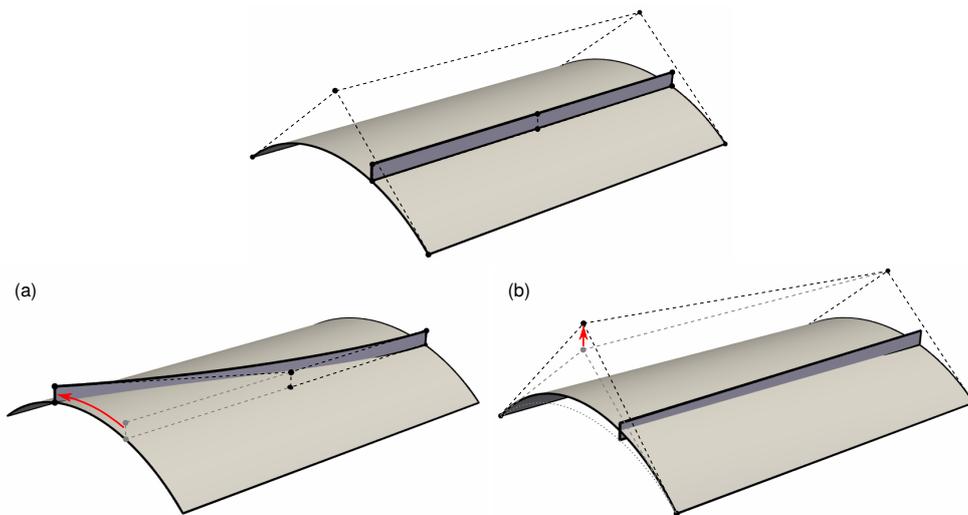


FIGURE 2 – Illustration d’une structure composée d’une géométrie *mère* (une portion de cylindre) et d’une géométrie *fille* (un raidisseur). En cas de modification d’une géométrie, il faut être en mesure de garantir la connectivité *mère-fille*. (a) Modification de la géométrie *fille*. (b) Modification de la géométrie *mère*. (d’après [3])

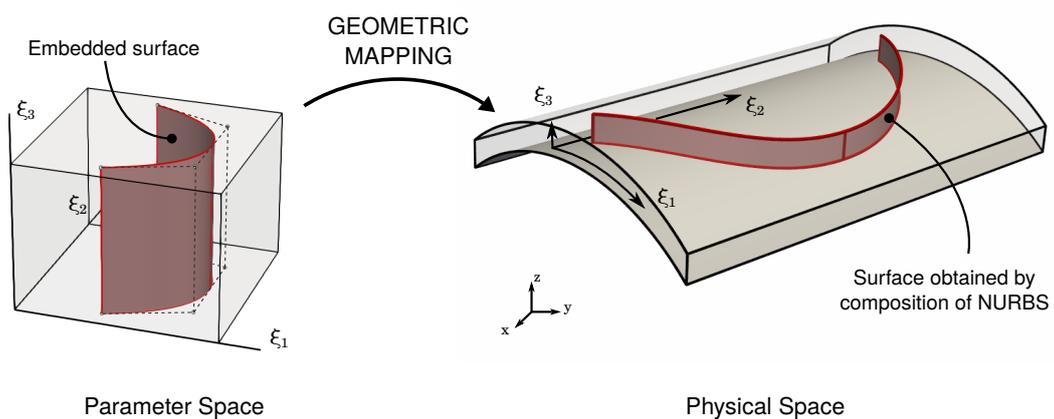


FIGURE 3 – Modélisation immergée d’une géométrie *fille* (ici un raidisseur). (d’après [3])

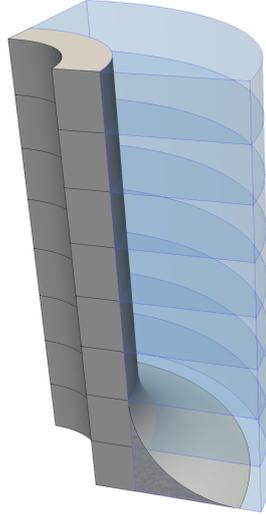


FIGURE 4 – Géométrie du modèle d’optimisation. L’enveloppe dans laquelle le congé est immergé est représentée en bleu.

La solution du problème adjoint ne fait pas intervenir les variables de *design*, ce qui permet de réutiliser la solution adjointe u^* pour chaque variable de *design*.

La dérivée de la fonction objectif s’écrit alors :

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + u^* \cdot \left(\frac{\partial F}{\partial x_i} - \frac{\partial K}{\partial x_i} u \right) \quad (5)$$

Les variables de design étant liées à la position des points de contrôle, les termes $\partial F/\partial x_i$ et $\partial K/\partial x_i$ peuvent être calculés élément par élément puis assemblés. À ce stade, la dérivée de la fonction objectif est calculée sur le modèle d’analyse.

Pour agréger ces résultats sur le modèle d’optimisation, on utilise l’opérateur de raffinement R qui permet de lier les points de contrôle Q du modèle d’analyse aux points de contrôle P du modèle d’optimisation *via* la relation :

$$Q = RP \quad (6)$$

Ainsi, les quantités dérivées par rapport aux points de contrôle du modèle d’optimisation peuvent s’exprimer en fonction des dérivées des même quantité par rapport aux points de contrôle du modèle d’analyse :

$$\frac{\partial \bullet}{\partial P} = R^T \frac{\partial \bullet}{\partial Q} \quad (7)$$

Les détails de calcul sont disponibles dans [2].

5 Exemple d’application

On s’intéresse ici à un démonstrateur composé d’une structure simple : un tube modélisé par des éléments solides et encastré à une extrémité. La face intérieure du tube est soumise à une pression hydrostatique. La structure est renforcée par un congé au niveau de l’encastrement. Ce congé est défini *via* une géométrie immergée dont les points de contrôle de l’enveloppe sont liés à ceux du cylindre. Par symétrie, seulement un quart de la structure est considéré (voir figure 4).

Le problème d’optimisation consiste à minimiser la souplesse de la structure en modifiant l’épaisseur du tube tout en conservant la masse initiale de la structure (une variation de masse de 2,5% est autorisée). L’évolution de la géométrie au cours de itérations d’optimisation est présentée à la figure 5. On peut constater que la modélisation immergée permet bien d’assurer la connexion entre le congé et le cylindre lorsque ce dernier est modifié.

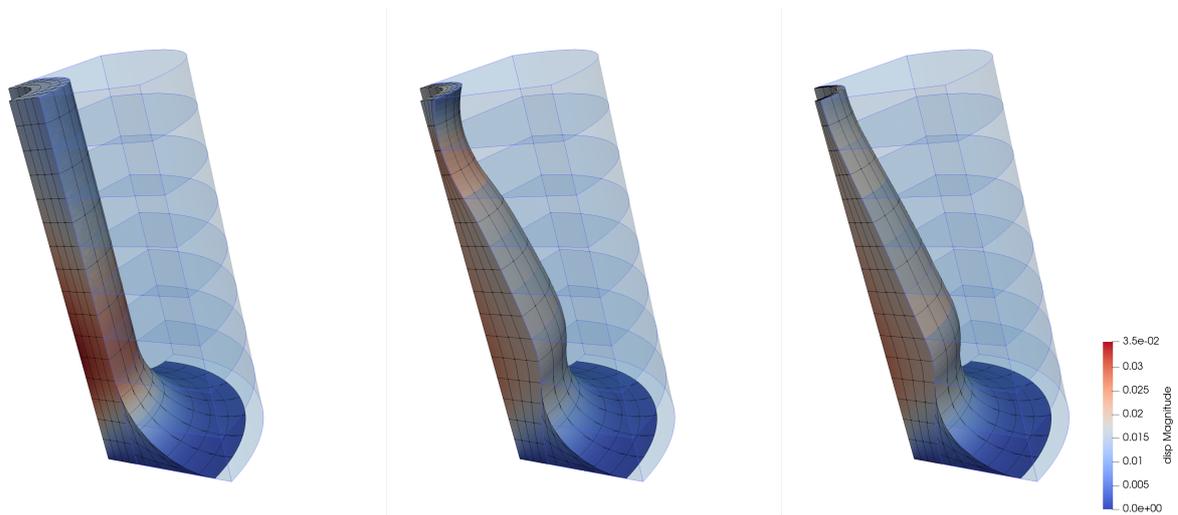


FIGURE 5 – Évolution de la géométrie au cours des itérations d’optimisation 0/13, 4/13 et 13/13. L’enveloppe dans laquelle le congé est immergé est représentée en bleu et la norme du champ de déplacement (calculé sur la modèle d’analyse) est représentée sur la géométrie.

Dans le cas étudié ici, la forme optimale est obtenue après 13 itérations et permet une diminution de 11% de la structure.

6 Conclusion

YETI est développé depuis 2015 au LaMCoS et publié sous licence LGPL depuis 2022 [5]. Il s’agit d’une plateforme ouverte à la communauté souhaitant implémenter des méthodes basées sur l’analyse isogéométrique. De plus, sa licence libre non contaminante permet de l’utiliser en tant que brique logicielle pour un outil métier à visée industrielle.

Les dernières fonctionnalités développées permettent l’optimisation de structures solides 3D *via* une méthode *immergée* permettant la prise en compte de géométries *filles* telles que des raidisseurs ou des congés.

Références

- [1] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric analysis : toward integration of CAD and FEA*. Wiley, Chichester, West Sussex, U.K. ; Hoboken, NJ, 2009. OCLC : ocn335682757.
- [2] T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. A New Lighting on Analytical Discrete Sensitivities in the Context of IsoGeometric Shape Optimization. *Archives of Computational Methods in Engineering*, September 2020.
- [3] T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. The embedded isogeometric Kirchhoff–Love shell : From design to shape optimization of non-conforming stiffened multipatch structures. *Computer Methods in Applied Mechanics and Engineering*, 349 :774–797, June 2019.
- [4] M. Guerder, A. Duval, T. Elguedj, and D. Dam. Volumetric Embedded Entities for the IsoGeometric Analysis of Complex Structures. *Computer Methods in Applied Mechanics and Engineering*, in press, corrected proof, 2023.
- [5] A. Duval. YETI, YET another Iga code. <https://lamcosplm.insa-lyon.fr/projects/yeti>.