

YETI : YET another Iga code

A. Duval¹, T. Elguedj²

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

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

Résumé — Publié sous licence LGPL, YETI est un logiciel qui se présente sous la forme d'une bibliothèque Python. Il propose une implémentation de l'analyse isogéométrique *boundary fitted*. Ses fonctionnalités sont dédiées à l'analyse mécanique ou thermique et à l'optimisation de forme. Une modélisation par composition de paramétrisations permet de modéliser efficacement des raidisseurs afin d'optimiser leur forme et position et une approche sans matrice à quadrature pondérée est utilisée pour accélérer les calculs.

Mots clés — analyse isogéométrique, couplage, calcul frugal, logiciel libre

1 Introduction

L'analyse isogéométrique (IGA) repose sur les primitives géométriques issues de la conception assistée par ordinateur (CAO), telles que les NURBS ou B-splines, pour décrire la solution de problèmes aux dérivées partielles [1].

L'IGA assure une représentation exacte de la géométrie et cette dernière n'est pas dégradée par les opérations de raffinement de maillage, renforçant ainsi le lien entre l'analyse et la modélisation géométrique. Pour certaines classes de problèmes d'optimisation de forme, cette approche s'avère particulièrement adaptée.

YETI est une bibliothèque open source [2] proposant une implémentation de cette méthode. Elle offre différentes fonctionnalités, notamment l'analyse de structures solides ou de coques, l'optimisation de forme avec calcul analytique des sensibilités pour les problèmes auto-adjoints, le couplage de patches via une méthode *mortar*, ainsi que la modélisation immergée. Cette dernière facilite la prise en compte et l'optimisation de géométries complexes telles que les congés ou les raidisseurs, tout en simplifiant la définition du problème d'optimisation. Enfin, une approche sans matrice à quadrature pondérée permet d'accélérer significativement les temps de calcul.

2 Architecture logicielle

YETI se présente sous la forme d'une bibliothèque Python [3] permettant à l'utilisateur de lire des fichiers d'entrée inspirés de la syntaxe Abaqus et définissant la paramétrisation NURBS d'un problème de mécanique ou de thermique, de raffiner la paramétrisation, de coupler différents domaines, recouvrants ou non et de calculer les composantes des matrices de masse, raideur ou second membre (sans nécessairement les former) du problème linéaire à résoudre.

Le routines de plus bas niveau sont écrites en Fortran dans un souci d'efficacité numérique et une interface en Python est réalisée à l'aide de l'utilitaire `f2py` [4]. La construction du code est gérée par `CMake` [5], `scikit-build` [6] et `cibuildwheel` [7]. Un jeu de cas tests a été mis en place pour valider la non-régression du code et le qualifier au regard des résultats de la littérature.

3 Fonctionnalités

3.1 Analyse

La fonctionnalité de base de YETI consiste à calculer les termes des matrices de masse, de raideur et le second membre permettant la définition d'un système linéaire utilisé dans la résolution d'un problème

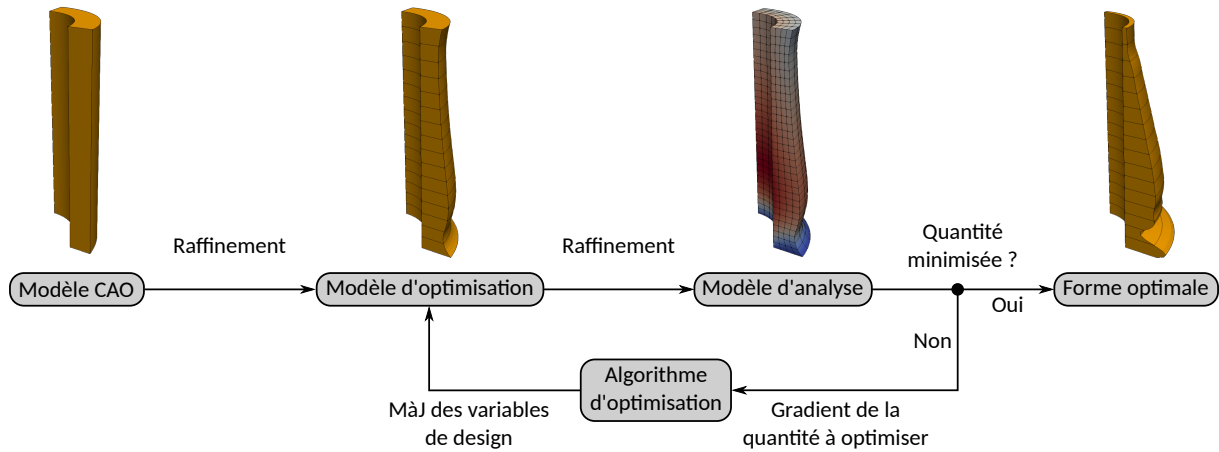


FIGURE 1 – Algorithme d’optimisation basé sur 2 modèles IGA liés entre eux par un opérateur de raffinement et permettant de calculer analytiquement les gradients des fonctionnelles d’optimisation.

de mécanique ou de thermique. Des éléments ont été implémentés afin de pouvoir traiter des cas de solides 2D ou 3D et des coques de Kirchhoff-Love.

La structure en produit tensoriel des fonction NURBS ou B-Spline peut être exploitée *via* une méthode sans matrice à quadrature pondérée pour accélérer significativement les temps de calcul [8, 9].

3.2 Optimisation

YETI permet de traiter des cas d’optimisation de forme pour les problèmes auto-adjoints (fonctionnelles d’optimisation faisant intervenir volume, compliance, déplacement, etc.) et dont les variables de design sont reliées à la position des points de contrôle de la géométrie. L’optimisation fait alors intervenir deux modèles [10] :

- un modèle d’optimisation sur lequel sont définies les variables de design qui pilotent la position des points de contrôle ;
- un modèle d’analyse, obtenu en raffinant le modèle d’optimisation afin d’obtenir une bonne qualité de solution.

YETI permet alors de calculer de façon analytique les sensibilités des fonctionnelles d’optimisation au regard des variables de design, accélérant ainsi de façon significative les algorithmes à gradient par rapport à l’utilisation des différences finies (Fig. 1).

3.3 Couplage

L’analyse isogéométrique impose, de par sa structure en produit tensoriel, de devoir décomposer une géométrie complexe en différentes sous-géométries simples (patches), homéomorphes à des carrés ou des cubes. Ces patches peuvent avoir des paramétrisations non compatibles et être recouvrants ou non.

YETI implémente un couplage faible de type *mortar*. L’égalité des déplacements dans la zone commune aux domaines à coupler est assurée *via* l’introduction de multiplicateurs de Lagrange, résultant en un problème matriciel faisant intervenir des matrices de couplage. Celui-ci peut être résolu soit de façon monolithique, soit en faisant appel à une méthode de décomposition de domaines de type FETI. Les conditions de couplage peuvent être calculées par intégration numérique sur une interface pour le cas de patches non recouvrants [11] ou en utilisant une méthode dérivée de la collocation dans le cas de patches recouvrants, afin de s’affranchir de la définition explicite de la zone d’intersection (Fig. 2 et 3).

3.4 Modélisation par composition de paramétrisation

Le couplage de domaines non compatibles peut être utilisé pour modéliser des éléments géométriques tels que des congés ou des raidisseurs. Cependant, il est nécessaire de pouvoir assurer une bonne connexion (étanchéité) entre ces différents éléments lorsque la géométrie est modifiée (Fig. 4).

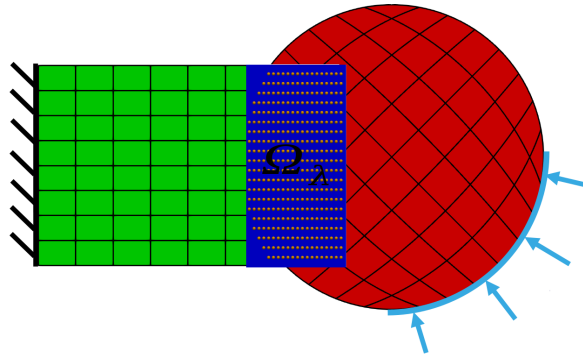


FIGURE 2 – Exemple de couplage recouvrant entre 2 patchs non conformes. Des multiplicateurs de Lagrange sont définis sur le domaine Ω_λ et les conditions de couplage sont formulées aux points de collocation (en orange).

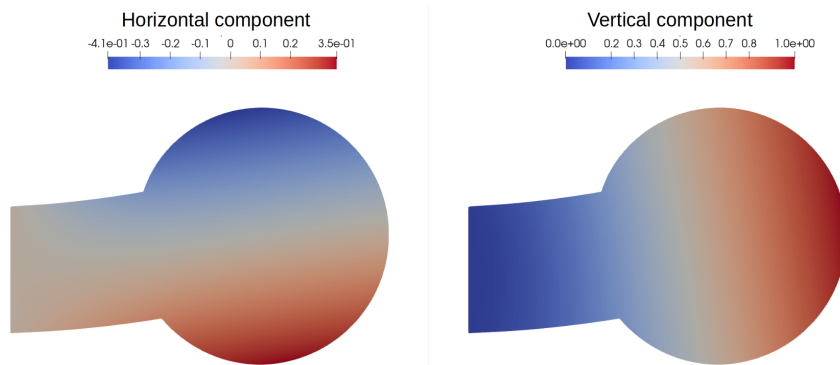


FIGURE 3 – Champ de déplacement obtenu après résolution du problème présenté dans la figure 2 et faisant intervenir 2 domaines recouvrants. Le couplage est résolu *via* une approche hybride *mortar/collocation*.

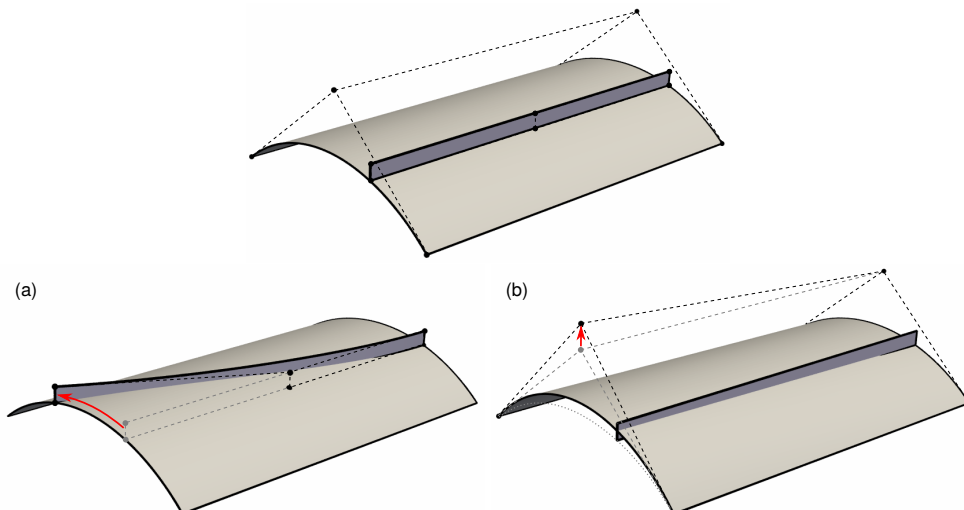


FIGURE 4 – Illustration d'une structure composée d'une géométrie *mère* (une portion de cylindre) et d'une géométrie *filie* (un raidisseur). En cas de modification d'une géométrie, il faut être en mesure de garantir la connectivité *mère-fille*. (a) Modification de la géométrie *filie*. (b) Modification de la géométrie *mère*. (d'après [12])

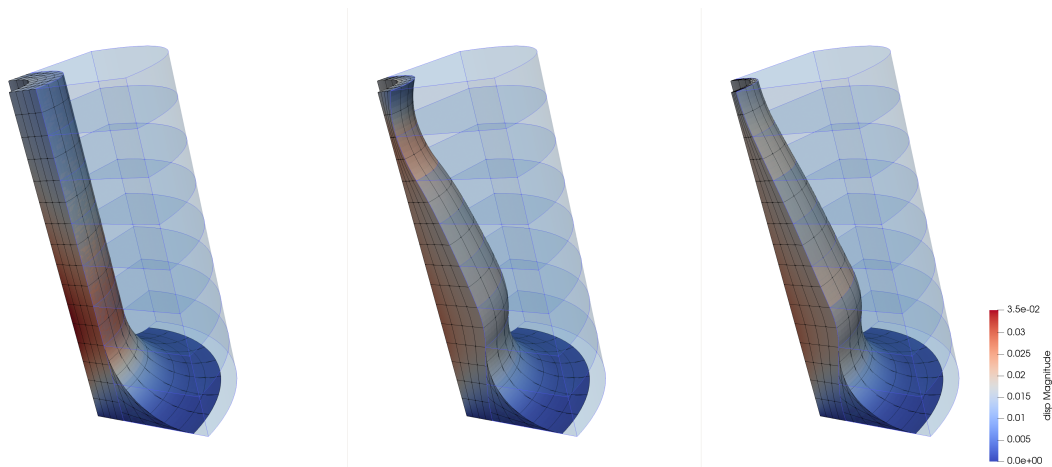


FIGURE 5 – Exemple d’optimisation de la forme d’un cylindre. L’enveloppe dans laquelle le congé est immergé est représentée en bleu. La forme du congé évolue en fonction de la forme du cylindre au cours du processus d’optimisation fin d’assurer une bonne connexion entre les 2 patches.

Pour ce faire, YETI implémente une modélisation par composition de paramétrisations où le raidisseur ou congé est immergé dans un volume NURBS, lui-même issu d’une surface de la géométrie mère [12, 13]. La modification de forme du raidisseur ou congé se fait ainsi en gardant une bonne connexion entre les patches (Fig. 5).

4 Conclusion

YETI est développé depuis 2015 au LaMCoS et publié sous licence LGPL depuis 2022. C’est une plateforme de recherche ouverte à la communauté pour l’implémentation de méthodes basées sur l’analyse isogéométrique. Sa licence non contaminante permet également de l’utiliser comme composant d’outils métier dans un cadre industriel.

Références

- [1] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric analysis : toward integration of CAD and FEA*. Wiley, Chichester, West Sussex, U.K. ; Hoboken, NJ, 2009. OCLC : ocn335682757.
- [2] yeti. <https://github.com/arnaudduval/yeti>.
- [3] yeti-iga. <https://pypi.org/project/yeti-iga/>.
- [4] NumPy. <https://numpy.org/>.
- [5] CMake. <https://cmake.org/>.
- [6] scikit-build. <https://scikit-build.readthedocs.io>.
- [7] cibuildwheel. <https://cibuildwheel.pypa.io/en/stable/>.
- [8] G. Sangalli and M. Tani. Matrix-free weighted quadrature for a computationally efficient isogeometric k-method. *Computer Methods in Applied Mechanics and Engineering*, 338 :117–133, August 2018.
- [9] Joaquín Cornejo Fuentes, Thomas Elguedj, David Dureisseix, and Arnaud Duval. A cheap preconditioner based on fast diagonalization method for matrix-free weighted-quadrature isogeometric analysis applied to nonlinear transient heat transfer problems. *Computer Methods in Applied Mechanics and Engineering*, 414 :116157, September 2023.
- [10] 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.
- [11] T. Hirschler, R. Bouclier, D. Dureisseix, A. Duval, T. Elguedj, and J. Morlier. A dual domain decomposition algorithm for the analysis of non-conforming isogeometric Kirchhoff–Love shells. *Computer Methods in Applied Mechanics and Engineering*, 357 :112578, December 2019.
- [12] 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.

- [13] Marie Guerder, Arnaud Duval, Thomas Elguedj, and Damien Dam. Volumetric embedded entities for the IsoGeometric Analysis of complex structures. *Computer Methods in Applied Mechanics and Engineering*, 417 :116426, December 2023.