

Un pont global entre AIG et MEF : Calcul et optimisation de forme, isogéométrique, non-intrusif à partir d'un code EF industriel.

R. Bouclier^{1,2}, J.C. Passieux², M. Tirvaudey^{2,3}, L. Chamoin³, M. Chapelier^{1,2}

¹ IMT, Université de Toulouse, UPS/UT1/UT2/INSA/CNRS, Toulouse, bouclier@insa-toulouse.fr

² JCA, Université de Toulouse, CNRS/INSA/ISAE/Mines Albi/UPS, Toulouse {passieux,tirvaude,chapelie}@insa-toulouse.fr

³ LMT, ENS Paris-Saclay/CNRS/Université Paris-Saclay, Cachan, chamoin@insa-toulouse.fr

Résumé — Bien que l'AIG ait suscité un grand intérêt ces dernières années, son déploiement au niveau industriel apparaît marginal. L'une des raisons est que son implémentation n'est pas aisée vis-à-vis de la structure élémentaire des codes EF. Basé sur les extractions de Bézier/Lagrange, un véritable pont global entre les fonctions NURBS et les polynômes de Lagrange est formulé, ce qui permet (1) d'implémenter l'AIG dans un code EF industriel de façon non-intrusive (c'est-à-dire sans toucher aux routines EF) et, (2) de réaliser de l'optimisation de forme isogéométrique à partir de calculs uniquement EF traditionnels.

Mots clés — Analyse isogéométrique, extraction de Bézier, implémentation non-intrusive, optimisation de forme, réduction de modèles.

1 Introduction

L'Analyse IsoGéométrique (AIG) a été introduite à l'origine dans [13] afin de résoudre des problèmes directement sur la géométrie issue d'un modèle de CAO. Pour ce faire, les polynômes de Lagrange sont remplacés dans l'approximation par des splines (c'est-à-dire des fonctions plus régulières) telles que les NURBS (*Non-Uniform-Rational-B-Splines* [18]) qui constituent à l'heure actuelle la technologie la plus répandue en CAO. En plus du lien renforcé entre calcul et CAO, l'AIG permet d'obtenir un meilleur ratio précision sur degrés de liberté [8] du fait de la régularité supérieure des splines, à savoir $C^{(p-1)}$ au travers des éléments du maillage pour un degré polynomial p . Cependant, en dépit de son grand enthousiasme, l'implémentation de l'AIG apparaît toujours relativement intrusive ce qui limite son déploiement au niveau industriel. L'AIG intervient en fait avec un espace paramétrique intermédiaire qui n'est pas compatible directement avec la structure de données élémentaire des codes EF existants. Quelques implémentations dans des environnements commerciaux EF existent (telles que dans LS-Dyna [11], Abaqus [6, 15], ou Radioss [16, 17]) mais cela reste assez marginal.

Dans la course au rapprochement de l'AIG avec la Méthode des Éléments Finis classiques (MEF), l'extraction de Bézier [2] puis de Lagrange [19] apparaissent comme les technologies les plus prometteuses. En effet, l'extraction de Lagrange permet d'exprimer des fonctions régulières B-Splines (respectivement NURBS) comme une combinaison linéaire de polynômes de Lagrange C^0 (resp. de fonctions rationnelles de Lagrange). Une nouvelle façon d'implémenter l'AIG où chaque élément à sa propre paramétrisation peut dès lors voir le jour. Ceci constitue une étape importante quant à l'intégration de l'AIG dans un environnement EF existant. Toutefois, il convient de souligner à ce stade que des modifications assez conséquentes tant au niveau élémentaire (modification de la routine des fonctions de base EF pour obtenir des fonctions rationnelles Lagrange) que vis-à-vis de l'assemblage (modification de la table de connectivité pour être compatible avec la numérotation AIG) doivent encore être apportées au code EF.

Dans ce travail, on s'intéresse dans un premier temps à mener de l'AIG dans un code EF standard de la façon la moins intrusive possible ; c'est-à-dire, en d'autres termes, que l'on cherche à effectuer des calculs isogéométriques à partir d'un code EF traditionnel tout en minimisant les efforts d'implémentation. En commençant avec l'extraction de Lagrange, on adopte un point de vue global et on formule un pont algébrique complet qui permet de passer directement de polynômes de Lagrange aux fonctions NURBS. On peut dès lors construire une nouvelle procédure d'implémentation de l'AIG dans laquelle aucune modification n'est apportée aux routines EF existantes, ce qui permet d'utiliser le code EF comme une véritable boîte noire. L'accès au code industriel complet n'est donc plus nécessaire et les routines EF,

souvent fortement optimisées, peuvent être utilisées telles quelles. En ce sens, l'approche proposée appartient à une classe de méthodes dites *non-intrusives* dont l'intérêt d'un point de vue industriel a pu être récemment démontré [3, 7, 9, 10]. Plus précisément, le code industriel EF *Code_Aster* [1] est envisagé ici pour mener de l'AIG de façon non-intrusive.

Dans un second temps, le pont global développé est utilisé dans ce travail pour réaliser de l'optimisation de forme isogéométrique de structures à partir de calculs uniquement EF traditionnels. Il est alors proposé de compléter l'approche multi-niveau existante en AIG [12, 14] afin de faire dialoguer un modèle d'optimisation NURBS avec un modèle EF fin adapté au calcul de la mécanique sous-jacente. Il en résulte une approche d'optimisation hybride AIG-MEF dans laquelle on tire parti des avantages des deux méthodes : d'un côté des évolutions de formes régulières sont pilotées par les NURBS, et de l'autre, les calculs mécaniques sont réalisés, à chaque itération de l'algorithme d'optimisation, dans un code EF existant. Le lien entre les deux modèles est explicite avec le pont AIG-MEF développé dans ce travail. On parle alors d'optimisation de forme isogéométrique non-intrusive.

2 Lien existant entre IGA et FEA

Pour commencer, on examine les techniques disponibles à l'heure actuelle permettant de lier AIG et MEF. Après un bref rappel concernant la construction des B-Splines et des NURBS, on se focalise ainsi sur l'extraction de Bézier [2] et sur l'extraction plus récente de Lagrange [19]. Pour plus de détails concernant les NURBS et les algorithmes de construction associés, le lecteur intéressé peut se référer à [5, 18].

2.1 L'analyse isogéométrique basée sur les NURBS

2.1.1 B-Splines

Les fonctions B-Splines sont définies à partir d'un vecteur nœud : $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ constitué de coordonnées paramétriques ξ_j rangées dans l'ordre croissant, p étant le degré polynomial des fonctions B-Splines et n le nombre de fonctions générées. La i -ème fonction de base B-Spline de degré p notée $N_{i,p}$ est construite de façon récursive en utilisant la formule de Cox-de Boor [5]. En général, une fonction de degré p possède $p - 1$ dérivées continues. Par ailleurs, si un nœud ξ_j a une multiplicité m_j , alors le nombre de dérivées continues diminue de m_j . À titre d'illustration, les figures 1(a) et 1(b) donnent, respectivement, l'allure des fonctions de base B-Splines et des fonctions de base classiques éléments finis pour un maillage composé de deux éléments quadratiques. Il peut être noté que seules 4 fonctions B-Splines de régularité C^1 en $\xi = 0.5$ sont générées contre 5 fonctions globales éléments finis (C^0 en $\xi = 0.5$). Ceci constitue la caractéristique principale de l'IGA qui permet d'obtenir un meilleur ratio précision sur degrés de liberté dans le cas d'un problème régulier.

Une courbe B-Spline est ensuite définie telle que :

$$C^{BS}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) P_i = \mathbf{P}^{BS T} \mathbf{N}(\xi), \quad (1)$$

où $\mathbf{P}^{BS} = \begin{bmatrix} x_1^1 \dots x_1^d \\ \dots \\ x_n^1 \dots x_n^d \end{bmatrix}$ est la matrice $n \times d$ (d étant la dimension de l'espace) qui regroupe les positions

des points de contrôle et $\mathbf{N} = \begin{bmatrix} N_1 \\ \dots \\ N_n \end{bmatrix}$ est le vecteur renfermant les fonctions de base B-Splines. Pour un domaine à plusieurs dimensions, les fonctions de bases B-Splines sont déterminées en effectuant un produit tensoriel de fonctions B-Splines 1D.

2.1.2 NURBS

Les fonctions NURBS multi-D $\mathbf{R}^{IGA} = \{R_A^{IGA}\}_{A=1}^{n_A}$ se construisent finalement à partir des fonctions B-Splines multi-D $\{N_A\}_{A=1}^{n_A}$ auxquelles il faut ajouter des poids $\{w_A^{IGA}\}_{A=1}^{n_A}$. On aboutit à des fonctions

rationnelles de la forme suivante :

$$\mathbf{R}_A^{IGA} = \frac{N_A W_A^{IGA}}{W^{IGA}}, \quad \text{avec} \quad W^{IGA} = \sum_{A=1}^{n_A} N_A W_A^{IGA}. \quad (2)$$

De façon similaire aux entités B-Splines, les objets NURBS sont ensuite générés à partir des points de contrôle $\mathbf{P}^{IGA} = \{P_A^{IGA}\}_{A=1}^{n_A}$ comme suit :

$$\mathcal{V}^{IGA} = \sum_{A=1}^{n_A} R_A^{IGA} P_A^{IGA} = \mathbf{P}^{IGA T} \mathbf{R}^{IGA}. \quad (3)$$

Comme tel, il est possible de représenter exactement toutes sections coniques (cercles, ellipses, etc) ce qui explique l'importance de cette technologie en CAO.

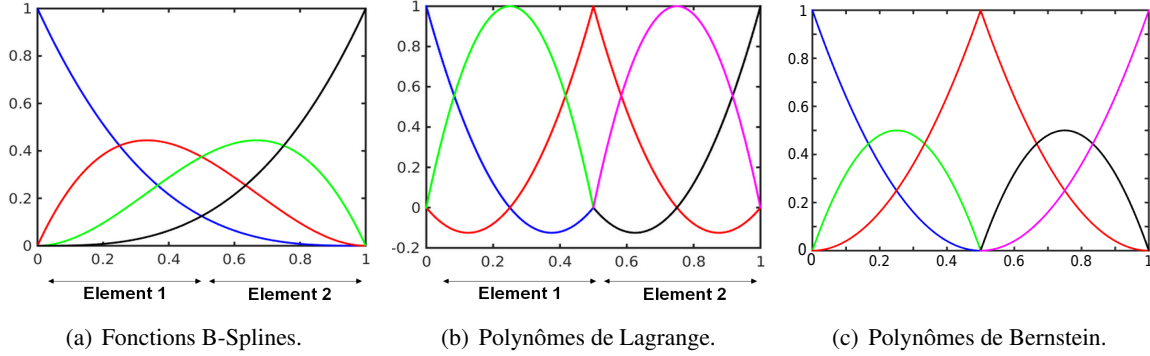


FIGURE 1 – Fonctions B-Splines, de Lagrange, et de Bernstein pour un maillage quadratique composé de deux éléments. Les tracés sont réalisés dans l'espace paramétrique ($\xi \in [0; 1]$).

2.2 L'extraction de Bézier

La première étape pour connecter AIG et MEF consiste à formuler les fonctions polynomiales régulières B-Splines comme une combinaison linéaire de polynômes C^0 sur le bords des éléments. Ceci est bien possible puisque l'espace généré par les fonctions B-Splines est inclus dans celui généré par des polynômes C^0 aux différents nœuds du maillage. En d'autres termes, l'idée ici est d'extraire la partie régulière de fonctions C^0 . C'est ce qui est fait au travers de l'opérateur de Bézier qui transforme une base de polynômes de Bernstein (voir figure 1(c)) en une base B-Spline (cf. figure 1(a)).

2.2.1 Décomposition de Bézier par insertion de nœuds

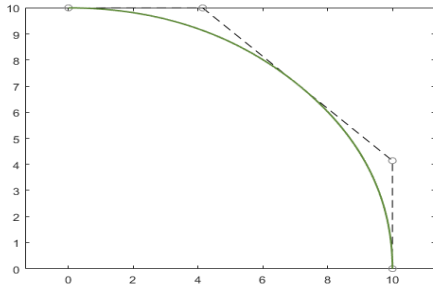
Le passage d'un maillage régulier B-Spline à un maillage C^0 de Bernstein est simple : il suffit pour cela de répéter tous les nœuds intérieurs des vecteurs de nœuds jusqu'à ce que ceux-ci atteignent une multiplicité p . Une procédure d'insertion de nœuds est donc effectuée ce qui a pour effet le calcul de nouveaux points de contrôle pour conserver la géométrie initiale B-Spline. À titre d'exemple, la décomposition est effectuée en figure 2 sur un arc généré à partir de deux éléments quadratiques B-Splines. Les fonctions B-Splines et de Bernstein associées sont celles représentées en figures 1(a) et 1(c), respectivement. Il est intéressant de noter à ce stade l'intérêt des polynômes de Bernstein : ces derniers possèdent une structure élémentaire qui est similaire à celle rencontrée dans la MEF (voir figure 1(c)) par rapport à figure 1(b)).

2.2.2 L'opérateur d'extraction de Bézier

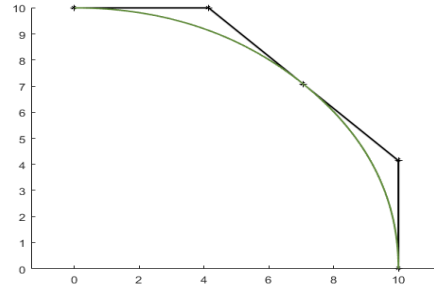
En appliquant la procédure d'insertion de nœuds explicitée ci-avant, une matrice \mathbf{C} d'extraction de Bézier peut être construite afin d'établir un lien direct entre fonctions B-Splines \mathbf{N} et de Bernstein \mathbf{B} :

$$\mathbf{N} = \mathbf{C}\mathbf{B}. \quad (4)$$

Il convient de souligner que les seuls *inputs* nécessaires pour construire cet opérateur sont les vecteurs de nœuds. Les points de contrôle de Bernstein \mathbf{P}^{BER} sont ensuite obtenus à partir de \mathbf{P}^{BS} de sorte à conserver la géométrie exprimée telle que Eq (1). En appliquant (4), il vient alors : $\mathbf{P}^{BER} = \mathbf{C}^T \mathbf{P}^{BS}$.



(a) Maillage de contrôle et géométrie construite à partir de fonctions B-Spline C^1 .



(b) Maillage de contrôle et géométrie construite à partir de fonctions de Bernstein C^0 .

FIGURE 2 – Décomposition de Bézier pour une courbe B-Spline générée à partir de deux éléments quadratiques. Le point de contrôle du milieu est ajouté au travers de la décomposition de Bézier afin de représenter la courbe initiale en terme de polynômes de Bernstein.

2.3 L'extraction de Lagrange

L'opérateur d'extraction de Lagrange transforme directement une base de polynômes de Lagrange en une base de fonctions B-Splines. Ce nouvel opérateur englobe l'opérateur de Bézier et va plus loin en faisant aussi le lien entre les polynômes de Bernstein et les polynômes de Lagrange.

2.3.1 Des polynômes de Lagrange aux polynômes de Bernstein

Les fonctions de Lagrange et de Bernstein de degré donné p génèrent le même espace polynomial (degré p , régularité C^0 aux bords des éléments). De plus, ces polynômes possédant la même structure élémentaire, le lien peut être établi au niveau élémentaire et est constant d'un élément à l'autre. Ainsi, il est aisé de déterminer l'opérateur \mathbf{D}_{LB} tel que : $\mathbf{B} = \mathbf{D}_{LB}\mathbf{L}$ (\mathbf{L} désignant la matrice des polynômes de Lagrange). Pour ce faire, le plus simple est d'utiliser les propriétés d'interpolation des polynômes de Lagrange : on prend les points dans le domaine élémentaire parent tels que $\xi_i = -1, 0$ et 1 , et on obtient alors directement :

$$\mathbf{D}_{LB}^{e1D} = \begin{bmatrix} B_1(-1) & B_1(0) & B_1(1) \\ B_2(-1) & B_2(0) & B_2(1) \\ B_3(-1) & B_3(0) & B_3(1) \end{bmatrix} = \begin{bmatrix} 1 & 1/4 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1/4 & 1 \end{bmatrix}. \quad (5)$$

En multi-D, il suffit ensuite d'appliquer le produit de Kronecker sur les opérateurs 1D correspondants.

2.3.2 Un lien direct entre polynômes de Lagrange et fonctions B-Splines

Maintenant qu'on a construit les opérateurs \mathbf{C} et \mathbf{D}_{LB} , le calcul de l'opérateur d'extraction de Lagrange \mathbf{D} qui transforme directement une base de Lagrange en une base B-Spline est aisé :

$$\mathbf{N} = \mathbf{D}\mathbf{L} \quad \text{avec} \quad \mathbf{D} = \mathbf{C}\mathbf{D}_{LB}. \quad (6)$$

Toutefois, il est à noter que pour plus d'efficacité numérique, \mathbf{D} n'est jamais construit de cette façon. En effet, il est plus judicieux d'appliquer la même procédure que pour l'opérateur Lagrange-Bernstein \mathbf{D}_{LB} à l'opérateur complet \mathbf{D} , c'est-à-dire d'évaluer simplement les fonctions B-Splines aux nœuds du maillage.

2.4 L'extraction dans le cas NURBS

Jusqu'à présent, on s'est cantonné à des géométries B-Splines pour exprimer le lien entre AIG et MEF. Cependant, en général, ce sont des fonctions rationnelles (telles que des NURBS) qui intègrent donc des poids qui génèrent les géométrie en CAO. L'extraction de Lagrange peut être étendue au cas des NURBS mais alors, c'est un lien direct entre NURBS et d'autres fonctions rationnelles qui peut être établi exactement. Plus précisément, il s'agit d'un lien direct entre NURBS et fonctions Lagrange

rationnelles regroupées dans le vecteur \mathbf{R}^{LAG} . Ce lien est matérialisé par la construction d'un opérateur \mathbf{D}_W faisant intervenir les poids des NURBS. On peut écrire :

$$\mathbf{R}^{IGA} = \mathbf{D}_W \mathbf{R}^{LAG} \quad \text{avec} \quad \mathbf{R}^{LAG} = \frac{\mathbf{W}^{LAG} \mathbf{L}}{W^{LAG}}. \quad (7)$$

Cette technologie nécessite par conséquent d'intégrer des poids \mathbf{W}^{LAG} associés aux polynômes de Lagrange \mathbf{L} et d'intégrer une fonction globale poids de Lagrange W^{LAG} . De telles opérations ne font pas partie des codes EF standards qui se limitent à l'utilisation de polynômes.

3 Implémentation non-intrusive de l'AIG

3.1 Pratique courante

La technique d'extraction de Lagrange offre la possibilité d'implémenter l'AIG dans un code EF standard. Toutefois, surtout pour le cas NURBS, il est à noter que cette implémentation nécessite toujours d'effectuer des modifications relativement importantes dans le code EF, tant au niveau élémentaire que vis-à-vis de l'assemblage des opérateurs. Plus précisément, les actions suivantes peuvent être listées : (1) modification de la routine des fonctions de base EF standard pour incorporer les poids de Lagrange et ainsi, construire les fonctions de Lagrange rationnelles à partir des polynômes de Lagrange existants, (2) appliquer l'extraction de Lagrange pour chaque élément en utilisant \mathbf{D}_W^e , (3) modifier la table de connectivité afin de réaliser un assemble AIG.

3.2 Schéma non-intrusif proposé

Dans le premier volet de ce travail, on se propose de minimiser d'avantage le niveau d'intrusivité de l'implémentation en envisageant de conserver tel quel l'ensemble des routines EF (c'est-à-dire que l'on va chercher à supprimer les modifications à l'intérieur de la routine des fonctions de forme EF (point 1 ci-dessus) et concernant l'assemblage (point 3 ci-dessus)). Pour ce faire, on adopte un point de vue global de l'extraction, comme représenté en figure 3.

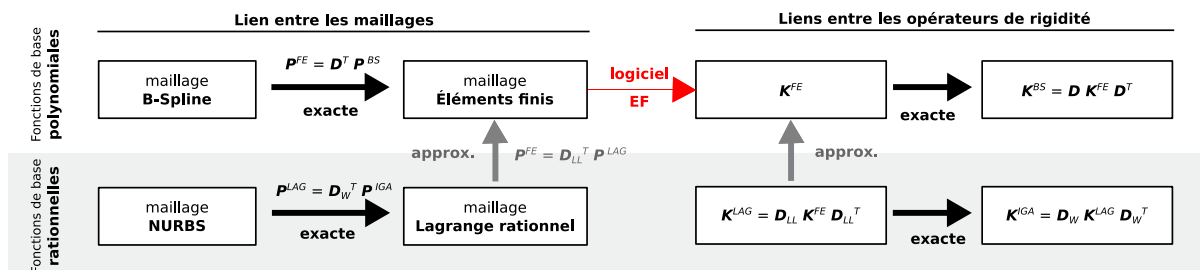


FIGURE 3 – Approche menée pour passer d'un maillage NURBS à un maillage EF à partir de différents opérateurs globaux (partie gauche). Ces opérateurs sont ensuite utilisés pour retrouver la matrice de rigidité NURBS par projection de la matrice EF calculée au travers du code EF classique, utilisé comme une boîte noire. Les flèches noires représentent des liens exacts tandis que les flèches grises décrivent des approximations.

Le chemin à parcourir à partir d'un maillage B-Spline est exact et consiste grossièrement à appliquer la procédure [19], mais d'un point de vue global. Au contraire, le chemin correspondant aux NURBS nécessite la construction d'un opérateur supplémentaire pour transformer une base de polynômes en une base de fonctions rationnelles. Une telle transformation ne peut se faire exactement puisque c'est l'espace des fonctions rationnelles qui inclut les polynômes associés et non l'inverse. Nous avons donc besoin à ce stade d'effectuer une projection. Pour plus de simplicité et d'efficacité, on se propose d'intervenir à l'étape de Lagrange [20], c'est-à-dire que ce sont les fonctions rationnelles de Lagrange que l'on projette sur l'espace généré par les polynômes de Lagrange. L'opérateur \mathbf{D}_{LL} est introduit pour cela. Avec les notations introduites précédemment, on écrit :

$$\mathbf{R}^{LAG} = \mathbf{D}_{LL} \mathbf{L}. \quad (8)$$

Les points de contrôle d'une discrétisation de type Lagrange rationnelle interpolent la géométrie. Ainsi, une façon simple de procéder consiste à prendre $\mathbf{P}^{FE} = \mathbf{P}^{LAG}$, soit $\mathbf{D}_{LL} = \mathbf{I}$ (c'est-à-dire l'identité). Pour plus de rigueur sur le plan mathématique, on propose aussi une projection locale (c'est-à-dire, élément par élément) utilisant la norme L^2 . Dans nos expérimentations numériques, il s'est avéré que la démarche pragmatique a mené à des précisions similaires à l'approche plus rigoureuse mathématiquement tout en limitant le coût de calcul.

Une fois que le maillage EF est créé à partir du maillage AIG dans une phase de pré-traitement, celui-ci est utilisé comme un *input* pour le code EF classique. On peut dès lors obtenir la matrice de rigidité EF correspondante \mathbf{K}^{FE} que l'on peut ensuite projeter sur l'espace isogéométrique plus régulier à l'aide des différents opérateurs développés :

$$\mathbf{K}^{IGA} = \mathbf{D}_W \mathbf{D}_{LL} \mathbf{K}^{FE} \mathbf{D}_{LL}^T \mathbf{D}_W^T. \quad (9)$$

Cette transformation est aussi appliquée pour le second membre :

$$\mathbf{F}^{IGA} = \mathbf{D}_W \mathbf{D}_{LL} \mathbf{F}^{FE}, \quad (10)$$

et alors le système $\mathbf{K}^{IGA} \mathbf{U}^{IGA} = \mathbf{F}^{IGA}$ peut être résolu afin d'obtenir le déplacement \mathbf{U}^{IGA} . Finalement, il est à noter que ce déplacement peut être reconverti en terme de déplacement EF :

$$\mathbf{U}^{FE} = \mathbf{D}_{LL}^T \mathbf{D}_W^T \mathbf{U}^{IGA}, \quad (11)$$

si bien que les routines existantes de post-traitement du code EF peuvent être utilisées.

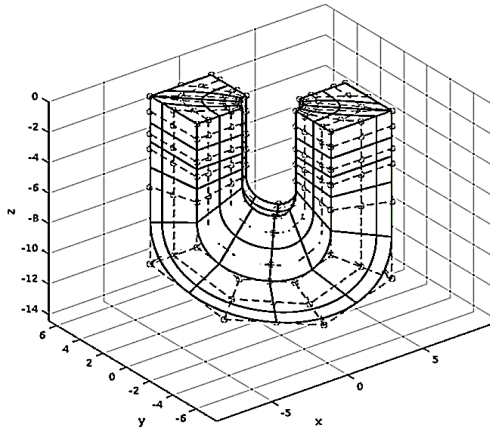
3.3 Résultats numériques

L'approche proposée a tout d'abord été appliquée afin de mener des calculs isogéométriques élastiques 2D et 3D à partir du code industriel EF *Code_Aster* développé par EDF R&D [1]. À titre d'exemple, on présente en figure 4 le cas test du fer à cheval qui est assez populaire dans la communauté AIG. Le fer à cheval est écarté par application de déplacements de signe opposé sur chacune de ses surfaces supérieures. À partir d'une paramétrisation connue NURBS du fer à cheval (voir figure 4(a)), nous avons été capables de construire un modèle EF (voir figure 4(b)) qui, une fois résolu en projection sur la base isogéométrique, nous a permis de retrouver la solution régulière NURBS attendue comme en témoigne le tracé des contraintes (*cf.* figures 4(c) et 4(d)).

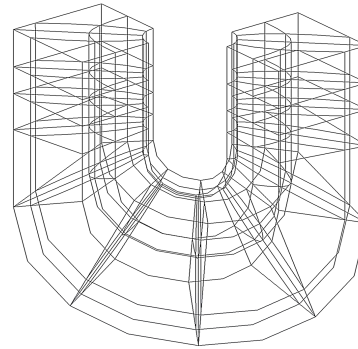
Des expérimentations similaires ont aussi pu être menées en non-linéaire, toujours à partir du code EF industriel *Code_Aster*. Pour cela, un script *python* a été écrit afin de court-circuiter, à chaque itération du solveur Aster « *STAT_NON_LINE in splitted commands* », la résolution du problème tangent. Celle-ci est dès lors réalisée en externe en projection sur la base isogéométrique de sorte à obtenir au final la solution non-linéaire AIG, sans jamais avoir eu à modifier une ligne des routines EF correspondantes au solveur itératif ou à la modélisation complexe non-linéaire. Au cours de l'exposé, nous montrerons des résultats obtenus notamment dans le cadre d'une non-linéarité matériau de type élasto-plastique.

4 Optimisation de forme isogéométrique non-intrusive

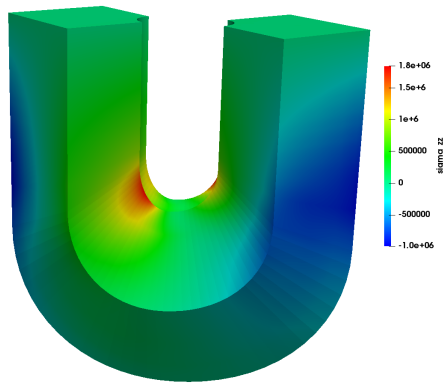
Du fait de son lien étroit avec la CAO, l'AIG est rapidement apparue comme une très bonne alternative à la MEF pour réaliser de l'optimisation de forme de structures [21]. L'approche permet en effet d'éliminer, au cours de l'optimisation, les étapes de maillages successives et/ou les étapes de lissage de la géométrie optimisée. Plus précisément, une approche multi-niveau [12, 14] est communément adoptée : d'un côté, on utilise un modèle AIG de *design* relativement creux pour générer des formes physiques cohérentes et régulariser l'optimisation, et de l'autre, un modèle d'analyse isogéométrique plus fin est mis en place pour garantir une bonne précision de calcul. L'intérêt avec l'AIG est que les différents modèles représentent exactement la même géométrie qui ne sont finalement qu'un raffinement l'un de l'autre. Pour aller plus loin dans le concept de multi-niveau, il est proposé d'ajouter en dernière brique le lien NURBS-EF développé ci-avant afin de passer de façon simple et explicite du modèle de calcul fin AIG à un modèle de calcul fin EF. Ce dernier modèle peut dès lors être calculé en utilisant, en mode boîte noire, les capacités d'un code EF industriel. On aboutit par conséquent à une approche d'optimisation hybride



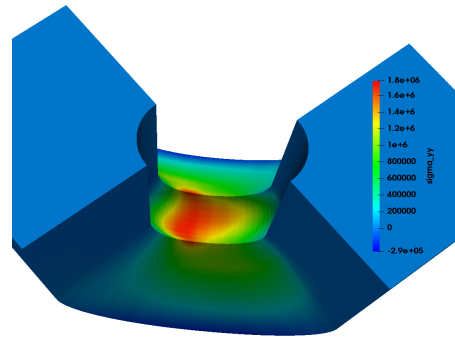
(a) Paramétrisation NURBS du fer à cheval.



(b) Maillage EF associé.



(c) Répartition de la contrainte σ_{zz} .



(d) Répartition de la contrainte σ_{yy} (zoom sur le centre du fer à cheval).

FIGURE 4 – Cas test du fer à cheval résolu à partir du pont global AIG-MEF développé et du code EF *Code_Aster*.

AIG-MEF qui cumule les avantages des deux méthodes : description régulière NURBS des évolutions de forme et calcul de la mécanique en EF au travers d'un code industriel performant. Des résultats d'optimisation de forme isogéométrique faisant appel à un code industriel tel que Abaqus seront présentés au cours de l'exposé.

5 Conclusion

Un pont global entre AIG basée sur les NURBS et EF classiques a été développé dans ce travail, ce qui a permis (1) d'implémenter l'AIG dans un code EF industriel de façon non-intrusive (c'est-à-dire sans toucher aux routines EF) et, (2) de réaliser de l'optimisation de forme isogéométrique à partir de calculs uniquement EF traditionnels. De ce fait, on aboutit à des stratégies de modélisation et de calcul cumulant les avantages des deux méthodes : on bénéficie à la fois de la meilleure représentation géométrique et de la régularité supérieure des NURBS, ainsi que de la possibilité de résolution dans des environnements EF bien établis. D'un point de vue général, ces travaux contribuent à d'avantage combler le gouffre entre AIG et MEF. Ils offrent un nouvel éclairage quant à la relation entre ces deux méthodes : l'AIG peut simplement être interprétée comme la projection de la MEF sur une base réduite [4] plus régulière.

Références

- [1] Électricité de France, *Code_Aster*, <http://www.code-aster.org>, 2014.

- [2] M. Borden, M.A. Scott, J.A. Evans, and T.J.R Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87(1–5) :15–47, 2011.
- [3] R. Bouclier, J.-C. Passieux and M. Salaün. Development of a new, more regular, mortar method for the coupling of NURBS subdomains within a NURBS patch : Application to a non-intrusive local enrichment of NURBS patches. *Computer Methods in Applied Mechanics and Engineering*, 316 :123-150, 2017 .
- [4] F. Chinesta, A. Huerta, G. Rozza, K. Wilcox. Model reduction methods - Part 1 : solids and structures. *Encyclopedia of Computational Mechanics, Second Edition*, 2017.
- [5] E. Cohen, T. Lyche and R. Riesenfeld. Discrete B-spline and subdivision techniques in computer aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14 :87-111, 1980.
- [6] A. Duval, T. Elguedj, H. Al-Akhras, and F. Maurin. *abqNURBS : implémentation d'éléments isogométriques dans Abaqus et outils de pré - et post - traitement dédiés*, In : 12e Colloque national en calcul des structures, Giens, 2015.
- [7] M. Duval, J.C. Passieux, M. Salaün and S. Guinard. Non-intrusive coupling : recent advances and scalable nonlinear domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 23(1) : 17–38, 2016.
- [8] J.A. Evans, Y. Bazilevs, I. Babuška and T.J.R. Hughes. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198(21–26) :1726–1741, 2009.
- [9] L. Gendre, O. Allix, P. Gosselet and F. Comte. Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Computational Mechanics*, 44(2) :233–245, 2009.
- [10] S. Guinard, R. Bouclier, M. Toniolli and J.C. Passieux. Multiscale analysis of complex aeronautical structures using robust non-intrusive coupling. *Advanced Modeling and Simulation in Engineering Sciences*, 5 :1 (2018).
- [11] S. Hartmann, D. Benson, A. Nagy. Isogeometric analysis with LS-DYNA. *J. Phys. Conf. Ser.* 734(3) :032125, 2016.
- [12] T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, J. Morlier. Isogeometric sizing and shape optimization of thin structures with a solid-shell approach. *Structural and Multidisciplinary Optimization*, 2018. DOI : 10.1007/s00158-018-2100-6.
- [13] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194 :4135–4195, 2005.
- [14] J. Kiendl, R. Schmidt, R. Wüchner, K.U. Bletzinger. Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. *Computer Methods in Applied Mechanics and Engineering*, 274 :148-167, 2014.
- [15] Y. Lai, Y.J. Zhang, L. Liu, X. Wei, E. Fang, J. Lua. Integrating CAD with Abaqus : A practical isogeometric analysis software platform for industrial applications. *Computers & Mathematics with Applications*, 74(7) :1648–1660, 2017.
- [16] M. Occelli, T. Elguedj, L. Morancay, S. Bouabdallah. *Implémentation de l'analyse isogométrique dans le solveur explicite Radioss*, In : 13e Colloque national en calcul des structures, Giens, 2017.
- [17] M. Occelli, T. Elguedj, S. Bouabdallah, L. Morancay. LR B-Splines implementation in the Altair Radioss™ solver for explicit dynamics isogeometric analysis. *Advances in Engineering Software*, Submitted.
- [18] L. Piegl and W. Tiller. The NURBS Book (Monographs in Visual Communication), second ed., Springer-Verlag, New York, 1997.
- [19] D. Schillinger, P.K. Ruthala and L.H. Nguyen. Lagrange extraction and projection for NURBS basis functions : A direct link between isogeometric and standard nodal finite element formulations. *International Journal for Numerical Methods in Engineering*, 108 :515–534, 2016.
- [20] M. Tirvaudey, R. Bouclier, J.-C. Passieux, L. Chamoin. Non-intrusive implementation of nonlinear IGA in an industrial finite element code. *International Journal for Numerical Methods in Engineering*, submitted.
- [21] W.A. Wall, M.A. Frenzel, C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197 (2008) 2976-2988.