



**HAL**  
open science

## abqNURBS: implémentation d'éléments isogéométriques dans Abaqus et outils de pré- et post-traitement dédiés

Arnaud Duval, Thomas Elguedj, Hassan Al-Akhras, Florian Maurin

### ► To cite this version:

Arnaud Duval, Thomas Elguedj, Hassan Al-Akhras, Florian Maurin. abqNURBS: implémentation d'éléments isogéométriques dans Abaqus et outils de pré- et post-traitement dédiés. 12e Colloque national en calcul des structures, CSMA, May 2015, Giens, France. hal-01514302

**HAL Id: hal-01514302**

**<https://hal.science/hal-01514302>**

Submitted on 26 Apr 2017

**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.

Public Domain

# abqNURBS : implémentation d'éléments isogéométriques dans Abaqus et outils de pré- et post-traitement dédiés

A. Duval, T. Elguedj, H. Al-Akhras, F. Maurin

Université de Lyon, CNRS  
INSA-Lyon, LaMCoS UMR5259, F-69621, France  
abqnurbs@insa-lyon.fr

---

**Résumé** — L'analyse isogéométrique est une technique de discrétisation d'équations aux dérivées partielles. Elle est basée sur des fonctions de forme NURBS, identiques à celles utilisées en CAO. Ceci permet notamment, contrairement à la discrétisation par la méthode des éléments finis, de conserver la géométrie initiale lors du raffinement de maillage. Afin d'utiliser cette technique au sein du code de calcul Abaqus, un paquet logiciel nommé abqNURBS, disponible sous licence libre CeCILL a été développé. Il se compose d'outils de pré- et post-traitement et de routines utilisateur Abaqus.

**Mots clés** — analyse isogéométrique, NURBS, Abaqus.

---

## 1 Introduction

L'analyse isogéométrique est basée sur les primitives géométriques utilisées en CAO (NURBS, T-splines, ...). Elle permet notamment, en comparaison avec l'analyse par éléments finis, une modélisation précise de la géométrie qui ne se dégrade pas lors du raffinement de maillage et de meilleures propriétés d'approximation [3]. Afin de permettre son utilisation industrielle, il est nécessaire d'implémenter cette méthode au sein des codes de calcul habituellement utilisés en bureau d'études. De plus, des outils de pré- et post-traitement adaptés doivent être proposés afin de faciliter la mise en données et l'exploitation des résultats.

## 2 Implémentation

L'implémentation d'éléments NURBS a été réalisée au sein du code de calcul Abaqus par le biais d'une *subroutine* permettant à l'utilisateur de formuler ses propres éléments. Une *subroutine* nommée UELMAT (User ELEMENT with abaqus MATERIAL), disponible à partir de la version 6.9 d'Abaqus [1], permet en outre d'utiliser la bibliothèque de lois de comportement proposée par le logiciel pour l'implémentation d'éléments utilisateur.

Un calcul de structure à l'aide d'éléments NURBS nécessite des données d'entrée supplémentaires par rapport à un calcul par éléments finis (voir tableau 1). À partir de l'indice d'une fonction de forme et d'une direction paramétrique, le tableau INC renvoie la coordonnée NURBS [4]. Le tableau IEN renvoie l'indice global de la fonction de forme à partir de l'indice local et du numéro d'élément. Enfin, le tableau ID retourne la position dans l'équation matricielle en fonction de l'indice de fonction de forme global et du numéro du degré de liberté considéré. Ainsi, si les coordonnées des points de contrôle du maillage NURBS et la table de connectivité IEN peuvent être passés *via* les fichiers d'entrée classiques d'Abaqus, les poids des points de contrôle ainsi que les vecteurs *knot* sont entrés dans un fichier supplémentaire, dont la lecture et la gestion des données sont effectuées par une seconde *subroutine* utilisateur nommée UEXTERNALDB, appelée préalablement à l'exécution du calcul. Le tableau IEN est construit lors du pré-traitement à l'aide de la matrice INC. Lors du calcul de la matrice de raideur, seules certaines composantes de INC sont nécessaires. On peut dès lors se contenter de passer ces composantes *via* le fichier d'entrée supplémentaire sous la forme d'un tableau NIJK.

La figure 1(a) montre le déroulement d'un calcul pour une structure composée d'un seul patch NURBS. Les blocs verts indiquent les différences par rapport à calcul EF classique. Les blocs bleus sont identiques, qu'il s'agisse d'analyse isogéométrique ou d'analyse par éléments finis. La figure 1(b)

TABLE 1 – Données nécessaires à un calcul EF ou IGA. Les données d’entrée supplémentaires figurent en gras

Analyse par éléments finis	Analyse isogéométrique NURBS
Coordonnées nodales	coordonnées des points de contrôle
Type d’élément	degré polynomial
connectivité des éléments (ID, IEN, LM)	connectivité des éléments (ID, IEN, LM, <b>INC</b> )
règle d’intégration numérique	règle d’intégration numérique
	<b>Poids des points de contrôle</b>
	<b>Vecteur knot</b>

montre la proposition de Hugues *et al.* [4] d’implémentation d’un calcul faisant intervenir plusieurs patches. Celle-ci implique une boucle supplémentaire sur les patches qui n’a pas d’équivalent en éléments finis. Cette stratégie est donc impossible à mettre en place dans un code existant pour lequel l’accès au code source est limité. Par simplicité, nous nous contentons d’un couplage  $C^0$  entre les patches. Le collage  $C^0$  est identique au collage  $C^0$  utilisé en éléments finis (figure 2). Le couplage  $C^1$  ne pose toutefois pas de difficulté supplémentaire en terme de structure de données. Les étapes de mise en données d’un calcul multi-patches sont les suivantes :

- le pré-traitement est réalisé pour chaque patch séparément ;
- une renumérotation globales des points de contrôle et des éléments est effectuée en fusionnant les points de contrôle identiques aux interfaces entre patches ;
- le fichier `.inp` est généré pour l’ensemble du domaine comme s’il s’agissait d’un unique patch ;
- les informations propres à chaque patch (vecteur knot etc.) sont passés dans un fichier `.NB`.

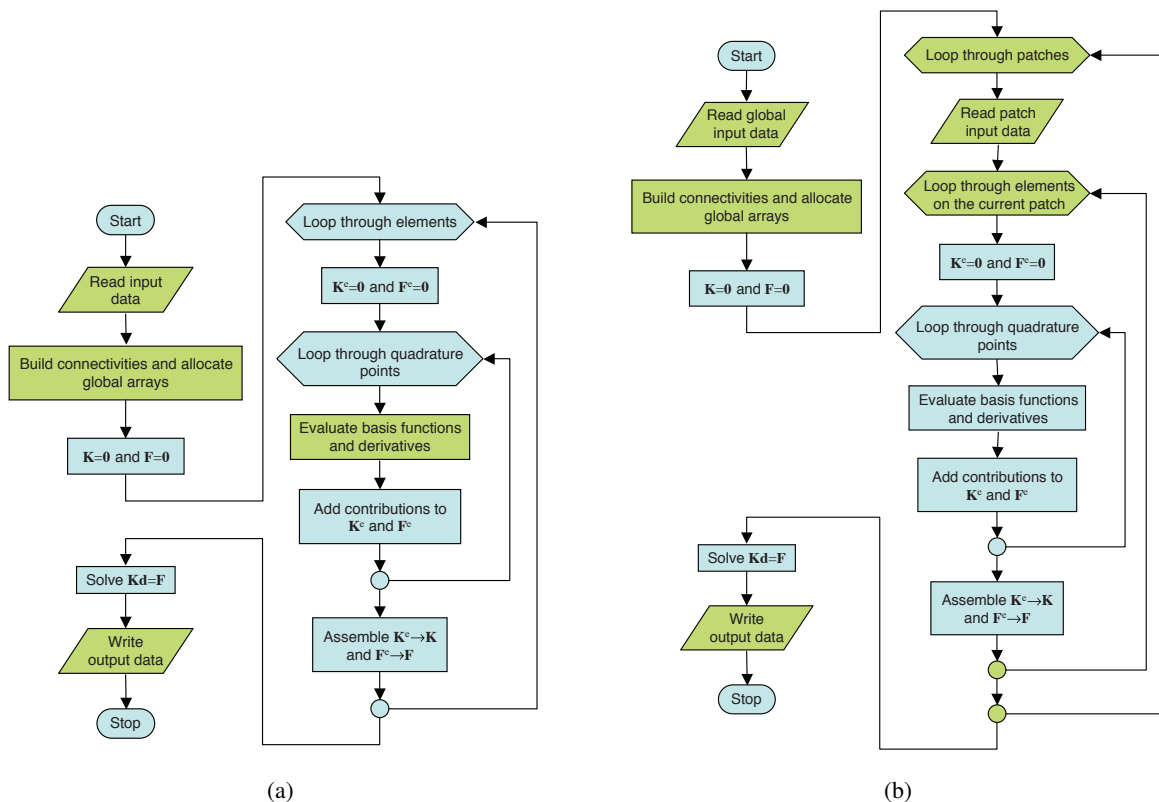


FIGURE 1 – Déroulement d’un calcul par éléments NURBS dans le cas d’un unique patch (a) et d’une géométrie multi-patches (b) (d’après [4]).

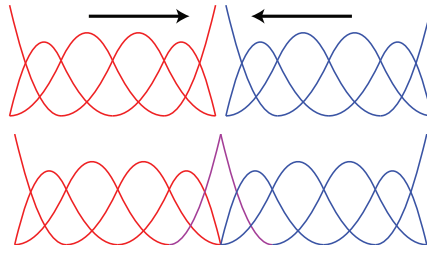


FIGURE 2 – Collage  $C^0$  entre deux patches.

### 3 Pré- et post-traitement

La construction de géométries NURBS complexes ne peut être faite qu'à l'aide d'outils dédiés. Pour ce faire, des extensions du logiciel Rhino ont été développées afin de pouvoir imposer des conditions aux limites sur une géométrie, créer des géométries NURBS 3D volumiques (Rhino étant à la base un outil dédié au rendu, il ne peut, en standard, modéliser que la peau d'une géométrie 3D) et générer les fichiers de données nécessaires au calcul Abaqus. Une extension au logiciel GeopDEs [2] a également été développée afin de créer les fichiers d'entrée nécessaires à Abaqus en ligne de commande.

Les résultats du calcul étant définis sur des éléments NURBS, ils ne peuvent être visualisés directement par l'interface graphique Abaqus CAE, cette dernière ne gérant que les éléments disponibles en standard dans le code de calcul. Un script de post-traitement a donc été développé afin de projeter les résultats calculés sur un maillage NURBS vers un maillage éléments finis. Chaque élément NURBS est subdivisé en un maillage éléments finis, les fonctions de forme NURBS étant utilisées pour interpoler les grandeurs définies aux points de contrôle du maillage NURBS (forces et déplacements) vers les nœuds du maillage EF projeté. Les grandeurs définies aux points de Gauss des éléments NURBS (contraintes, déformations, ...) sont quant à elles préalablement projetées par une méthode de moindres carrés globale aux points de contrôle NURBS. Cette étape est coûteuse en temps de calcul car elle fait intervenir l'assemblage et l'inversion de la matrice de masse du maillage NURBS. L'interface de programmation C++ d'Abaqus a donc été privilégiée par rapport à l'interface de programmation Python également disponible pour garantir un temps raisonnable de post-traitement.

### 4 Exemples d'étude

La figure 3 illustre les résultats obtenus pour un calcul effectué sur une géométrie 3D constituée d'un seul patch. La comparaison porte sur les valeurs de la composante  $\sigma_{33}$  du tenseur des contraintes. Les résultats obtenus à l'aide des éléments NURBS implémentés dans Abaqus concordent bien avec ceux obtenus par Hugues [3].

L'implémentation réalisée permet de prendre en compte différents types de non-linéarités : géométriques par le biais de la formulation des éléments qui permet la prise en compte des grands déplacements et matériau grâce à l'utilisation de la bibliothèques de lois de comportement d'Abaqus. Ainsi, la figure 4 représente un calcul en déformations planes d'un cylindre sous pression, dans le cas d'un matériau élastique ou parfaitement plastique.

### 5 Conclusion

Une implémentation de l'analyse isogéométrique basée sur les NURBS a été réalisée au sein du code de calcul Abaqus. Elle permet de prendre en compte les géométries comportant un seul ou plusieurs patches, par le biais d'une continuité  $C^0$  aux frontières des patches. L'utilisation d'une sous-routine utilisateur UELMAT donne un accès direct aux différentes lois de comportement disponibles dans Abaqus en petites et grandes déformations (NLGEOM). Des outils de post-traitement utilisant les interfaces Python/C++ d'Abaqus sont accessibles directement depuis l'interface d'Abaqus CAE afin de pouvoir visualiser les résultats obtenus. Enfin, des outils de pré-traitement ont été développés permettant la mise en données

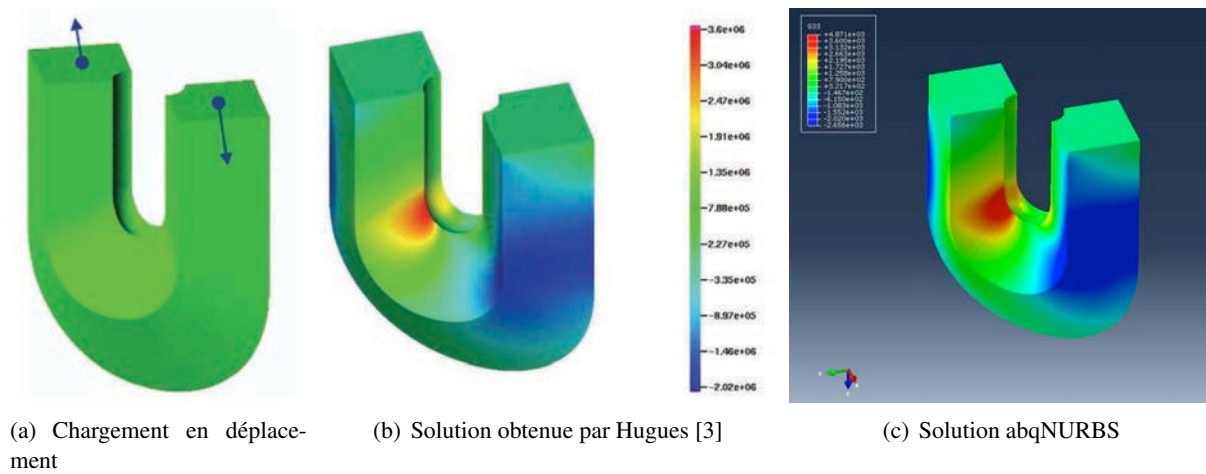


FIGURE 3 – Exemple de calcul 3D sur une géométrie de type « fer à cheval » constituée d'un seul patch NURBS [3]. Comparaison entre le résultat obtenu par Hugues et la solution obtenue à l'aide des éléments implémentés dans Abaqus. La vue isocouleur représente la composante  $\sigma_{33}$  du tenseur des contraintes.

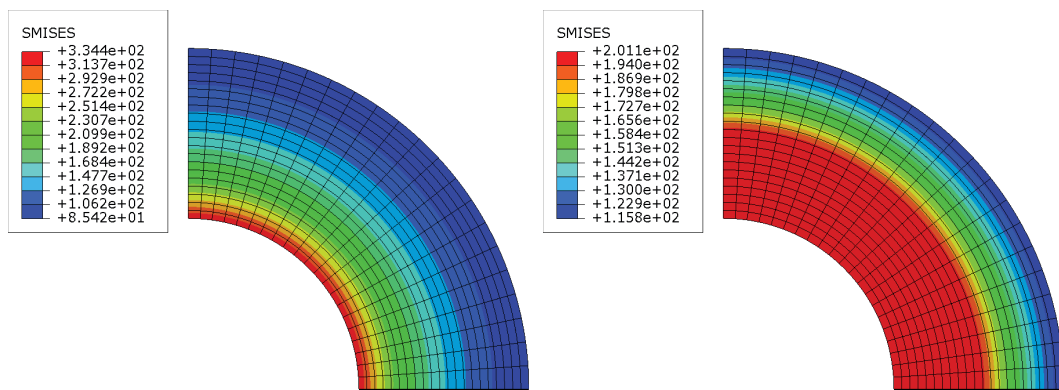


FIGURE 4 – Contrainte de Von Mises dans le cas d'un cylindre soumis à une pression interne.  $R = 2$  m,  $r = 1$  m,  $P = 145$  MPa,  $E = 200\,000$  MPa,  $\nu = 0,3$ . La figure de gauche illustre le cas élastique et celle de droite le cas parfaitement plastique ( $\sigma_Y = 200$  MPa).

d'études 2D/3D *via* le logiciel Rhino ou *via* Matlab/GeoPDEs.

Les développements futurs viseront à introduire des degrés de liberté en température afin de pouvoir réaliser des études thermomécaniques. De plus, l'utilisation des fonctions d'interpolation NURBS permet d'assurer une continuité  $C^1$  et donc d'utiliser des lois de comportement basées sur le gradient du tenseur de déformations. Enfin, l'implémentation de l'analyse isogéométrique basée sur l'extraction de Bezier est envisagée. Les différents outils logiciels présentés sont développés sous licence CeCILL et disponibles à l'adresse <http://abqnurbs.insa-lyon.fr>.

## Références

- [1] Dassault Systèmes, *Abaqus 6.9-EF online documentation*, 2009
- [2] C. de Falco, A. Reali, R. Vazquez. *GeoPDEs, a research tool for Isogeometric Analysis of PDEs*, Advances in Engineering Software, vol. 42, 1020-1034, 2011
- [3] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs. *Isogeometric analysis : CAD, finite element, NURBS, exact geometry and mesh refinement*, Computer Methods in Applied Mechanics and Engineering, 4135-4195, 2005.
- [4] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs. *Isogeometric Analysis : Toward Intregration of CAD and FEA*, Wiley, 2009.