Non-invasive implementation of nonlinear isogeometric analysis in an industrial FE software

Marie Tirvaudey, Robin Bouclier, Jean-Charles Passieux, Ludovic Chamoin

Non-invasive implementation of nonlinear IsoGeometric Analysis in an industrial FE software

Marie Tirvaudey [1,2], Robin Bouclier [1,3], Jean-Charles Passieux[1] and Ludovic Chamoin [2]

[1]Institut Clément Ader (ICA), Université de Toulouse, CNRS/INSA/ISAE/Mines Albi/UPS, Toulouse, France

[2] Laboratoire de Mécanique et Technologie (LMT), ENS Paris-Saclay, CNRS, Université Paris-Saclay, Cachan, France

[3] Institut de Mathématiques de Toulouse (IMT), Université de Toulouse, UPS, UT1, UT2, INSA, CNRS, Toulouse, France

August 26, 2019

Abstract

Purpose. The purpose of this paper is to further simplify the use of NURBS in industrial environnements. Although isogeometric analysis (IGA) has been the object of intensive studies over the past decade, its massive deployment in industrial analysis still appears quite marginal. This is partly due to its implementation, which is not straightforward with respect to the elementary structure of finite element (FE) codes. This often discourages industrial engineers from adopting isogeometric capabilities in their well-established simulation environment.

Methodology. Based on the concept of Bézier and Lagrange extractions, a novel method is proposed to implement IGA from an existing industrial FE code with the aim of bringing human implementation effort to the minimal possible level (only using standard input-output of FEA codes, avoid code-dependent subroutines implementation...). An approximate global link to go from Lagrange polynomials to NURBS functions is formulated, which enables the whole FE routines to be untouched during the implementation.

Findings. As a result, only the linear system resolution step is bypassed: the resolution is performed in an external script after projecting the FE system onto the reduced, more regular, isogeometric basis. The novel procedure is successfully validated through different numerical experiments involving linear as well as nonlinear isogeometric analyses using the standard input/ouput of the industrial FE software *Code_ Aster*.

1 Introduction

Isogeometric Analysis (IGA) was introduced by Hughes *et al.* [Hughes *et al.* 2005] and later detailed by Cottrell *et al.* [Cottrel *et al.* 2009] in order to solve problems directly on the geometry extracted from a Computer-Aided-Design (CAD) model. The main idea is to use the same basis, *e.g.* Non-Uniform-Rational-B-Splines (NURBS) [Cohen *et al.* 1980, Piegl and Tiller 1997] or T-Splines [Bazilevs *et al.* 2010], for analysis as the one used to describe the geometry in CAD. Beyond the reinforced link between CAD and analysis, IGA turned out to be a superior computational mechanics technology [Evans *et al.* 2009, Lipton *et al.* 2010, Schillinger *et al.* 2013, Yin *et al.* 2015, Rauen *et al.* 2017]. However, despite its real enthusiasm in the computational mechanics community, the implementation of IGA in existing industrial codes still appears quite invasive, which limits its massive deployment in industry. Indeed, IGA comes with an intermediate space for the definition of the shape functions which questions the element-wise structure of most existing FE codes. Unlike the standard Finite Element Method (FEM) where each element has its own parametrization, the parametric space for NURBS is localized onto a patch that is composed of several elements. Some IGA implementations in commercial FE packages exist such as in LS-Dyna [Hartmann *et al.* 2011, Benson *et al.* 2013, Hartmann *et al.* 2016, Chen *et al.* 2016], Abaqus [Elguedj *et al.* 2012, Duval *et al.* 2015, Lai *et al.* 2017] or Radioss [Occelli *et al.* 2019], but it is still quite a few.

The concept of Bézier extraction has proved to be a milestone to tie IGA and FEM closer together. Originally introduced by Borden *et al.* [Borden *et al.* 2011] for B-Splines and NURBS, the concept has now been generalized to a large variety of advanced splines such as T-Splines [Scott *et al.* 2011, Scott *et al.* 2012], hierarchical B-Splines and NURBS [Schillinger *et al.* 2012, Hennig *et al.* 2016, Angella *et al.* 2018], hierarchical T-Splines [Evans *et al.* 2015, Chen and de Borst 2018], and LR B-Splines [Dokken *et al.* 2013]. Focusing on B-Splines and NURBS, the technology enables to formulate a smooth polynomial B-Spline (respectively NURBS) function in

terms of C^0 polynomial Bernstein (resp. rational Bézier) functions. Nevertheless, it must be stressed, at this stage, that implementing this strategy still requires modifications both at the elementary and at the assembly levels: (1) modify the standard FE shape functions subroutine to introduce a new element based on polynomial Bernstein (or rational Bézier) functions, (2) apply the extraction for each element and, (3) modify connectivity since it is based on the NURBS numbering.

An extension of the idea of Bézier extraction to standard nodal FE functions has been more recently proposed in Schillinger *et al.* [Schillinger *et al.* 2016]. It gave birth to the Lagrange extraction operator that directly links Lagrange nodal basis with smooth B-Spline basis. This new operator encompasses the Bézier operator and offers an alternative and simple implementation: it merely requires to evaluate the B-Spline basis functions at the nodal points associated to the Lagrange polynomials. The Lagrange extraction especially appears of great interest for geometries based on polynomial B-Splines. Indeed, it eliminates the need for introducing Bernstein polynomials as new shape functions, which makes it possible not to touch the FE subroutines of a standard FE code at the element level. The Lagrange extraction thus goes a step further to allow for the integration of IGA in an industrial FE code with minimal invasiveness. However, it has to be underlined from this overview that the procedure cannot be applied to the general situation of NURBS (or any rational splines). In this case, a family of rational Lagrange functions can be introduced and used to directly express the NURBS basis; the problem is that such functions (that integrate weights and a weight function) are not part of a typical FE code which is restricted to polynomial basis functions. Moreover, it may be noticed that only the potential of the methodology has been underlined in [Schillinger *et al.* 2016]. Up to now, it seems that only a very recent attempt that extends this framework to the real open source FE-code FEniCS can be listed [Kamensky and Bazilevs 2019].

If pure numerical efficiency is the only criterion for qualifying an implementation of IGA, the best approach would certainly be to rewrite everything into a dedicated piece of code written using a low-level language (see, *e.g.*, PetIGA [Dalcin *et al.* 2016] for high-performance IGA based on the PETSc library, [Chang *et al.* 2016] for an implementation using Fortran and [Nguyen *et al.* 2015] for global implementation aspects). In addition, let us mention that recent works to implement IGA efficiently start with a modification of the standard looping over elements in the assembly process [Calabro *et al.* 2017], which is incompatible with the use of standard FE codes. All of these approaches may require significant development efforts, especially whenever a new nonlinear constitutive law needs to be added. The approaches based on subroutines, that were mentioned before, represent also interesting alternatives, as constitutive laws of existing industrial codes can be reused. However, these routines depend on the considered code and still require a certain effort of implementation. This is why, in this study, it is chosen to reduce the human time required for the computer implementation (even if, obviously, this one may come with a slight increase of the computational cost).

In other words, this work aims at performing IGA in an available FE software in the least possible invasive manner. Starting with the Lagrange extraction technology, a global view is adopted and a full algebraic bridge that directly goes from Lagrange nodal polynomials to NURBS functions is formulated. It leads to a novel, alternative implementation procedure in which no modification of the whole FE routines is required. As a result, and compared to the current practice, no modification of the shape functions subroutine and of the assembly is performed. More precisely, the strategy simply consists of the projection of the FE linear system on an *ad-hoc* regular reduced basis whose algebraic construction is automatic and represents an objectively moderate implementation effort. In this way, the method solely uses features offered by most modern industrial FE codes, such as, for instance, the possibility to extract the (tangent) stiffness operators. This appears of crucial interest regarding industrial FE environments since the access to the whole code is not needed and the FE routines that may be highly optimized are not touched. In that sense, the proposed method lies in the family of so-called *non-invasive* (also denoted *non-intrusive*) methods that has been recently applied to local-global coupling [Gendre *et al.* 2009, Bouclier *et al.* 2016, Bouclier *et al.* 2017], domain decomposition [Duval *et al.* 2016], contact problems [Oumaziz *et al.* 2017, Oumaziz *et al.* 2019], transient dynamics analysis [Bettinotti *et al.* 2014] and stochastic partial differential equations [Chevreuil *et al.* 2013]. Their ultimate goal is then to tackle real industrial applications (see, *e.g.*, [Guinard *et al.* 2018] for real aeronautical structures). The strategy is applied here to simply incorporate isogeometric capabilities in the industrial FE code *Code_Aster*, which is a familiar open source software package for numerical simulation in structural mechanics developed by the EDF R&D company [Code_Aster 2014].

The paper is organized as follows: Section 2 specifies the existing link between NURBS-based IGA and FEM by reviewing the Bézier and Lagrange extraction technologies. Section 3 is devoted to the development of the proposed implementation procedure. Then, for validation and demonstration purposes, some numerical experiments are first carried out involving 2D and 3D elasticity in Section 4, before performing the more complex simulation of a

nonlinear elastoplastic structure in Section 5. Finally, Section 6 concludes on this work.

2 The link between IGA and FEM

In this section, the fundamentals to tie IGA and FEM closer together are given. Although the Lagrange extractor encompasses the Bézier one, it is chosen here to start with the original Bézier version [Borden *et al.* 2011] before introducing the Lagrange transformation [Schillinger *et al.* 2016] given the importance of the concept of Bézier extraction in the state-of-the-art of IGA. For completeness, note that further details regarding NURBS and related algorithms can be found in [Cohen *et al.* 1980, Piegl and Tiller 1997]. An existing analysis-suitable NURBS or B-Splines representation is considered as an input to this study. To go from the CAD software to the analysis-suitable representation the reader is referred to [Al-Akhras *et al.* 2016].

2.1 NURBS-based IGA

2.1.1 B-Splines.

From a set of non-decreasing coordinates in the parametric space collected in the knot vector $\boldsymbol{\Xi} = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$, a basis of univariate B-Spline functions of order p, denoted by $(N_{i,p})_{i\in\{1..n\}}$, can be defined recursively using the Cox-de Boor formula [Cohen *et al.* 1980]. A B-Spline function of order p has $p-1$ continuous derivatives. Besides, if a knot ξ_l has a multiplicity m_l, the number of continuous derivatives at this knot will decrease by m_l. A comparison between B-Spline and Lagrange shape functions is given in Figure 1. Then, a B-Spline curve $\mathcal{C}^{BS}$ is constructed as follows:

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

where $\mathbf{P}^{BS} = \begin{bmatrix} x_1^1..x_1^d \\ ... \\ x_n^1..x_n^d \end{bmatrix}$ is the $(n \times d)$ matrix that collects the d coordinates of the n control points and $\mathbf{N} = \begin{bmatrix} N_1 \\ ... \\ N_n \end{bmatrix}$ is the vector of the B-Spline basis functions. For a multi-dimensional space domain, the B-Spline basis functions are determined by performing the tensor product of 1D B-Spline functions, so that in 3D it yields:

$$N_A = N_{i,p}(\xi) \times N_{j,q}(\eta) \times N_{k,r}(\zeta), \tag{2}$$

for control point P_A that corresponds to the i^{th}, j^{th} and k^{th} control points in each univariate spatial direction.

2.1.2 NURBS.

The multivariate NURBS functions $\mathbf{R}^{IGA} = \left\{R_A^{IGA}\right\}_{A=1}^{n_A}$ are defined from a set of n_A B-Spline functions $\{N_A\}_{A=1}^{n_A}$. In order to do so, it is necessary to introduce $\left\{w_A^{IGA}\right\}_{A=1}^{n_A}$ as the weights of each of the control points $\mathbf{P}^{IGA} = \left\{P_A^{IGA}\right\}_{A=1}^{n_A}$ associated to the NURBS entity. The rational functions read as follows:

$$R_A^{IGA} = \frac{N_A w_A^{IGA}}{W^{IGA}}, \quad \text{where} \quad W^{IGA} = \sum_{A=1}^{n_A} N_A w_A^{IGA}. \tag{3}$$

Following the definition of the B-Spline entities in Eq. (1), NURBS objects are then defined such that:

$$\mathcal{V}^{IGA} = \sum_{A=1}^{n_A} R_A^{IGA} P_A^{IGA} = \mathbf{P}^{IGA^T}\, \mathbf{R}^{IGA}. \tag{4}$$

The NURBS functions exactly describe the geometry of conic sections. For illustration purpose, an example of a 2D circular beam is considered in Figure 1(a). Starting with a quadratic one-element mesh, one can perform k-refinement [Cottrell *et al.* 2007] in the arc direction to obtain a quadratic mesh composed of two elements.

Remark 1 *The present article is restricted to quadratic functions since almost all industrial FE codes do not go beyond second-order Lagrange finite elements. Nevertheless, we emphasize that the developed implementation could be directly applied to higher-order spline-based functions, provided that the corresponding higher-order finite elements are available in a standard FE code.*

2.2 The Bézier extraction

The first step to link IGA and FEM is to formulate the smooth polynomial B-Spline functions in terms of polynomials that are C^0 on the element edges. This is possible as the space generated by the B-Spline functions is included in the one generated by the C^0 functions. Such a link has been developed in [Borden *et al.* 2011] under the name of Bézier extraction which maps the Bernstein basis onto the B-Spline basis.

2.2.1 Bézier decomposition by knot insertion.

In order to form a structured C^0 Bernstein mesh from a smooth B-spline mesh, one simply needs to repeat all the inside knots of the knot vectors until they reach a multiplicity p (see Figure 1(b) for the example of a quadratic B-Spline curve with two elements). The geometry is preserved with the resulting C^0 Bernstein representation. It must be stressed at this stage that the advantage of Bernstein functions is that they exhibit an elementary structure which is similar to FEM (see Figure 1(c)).

2.2.2 Bernstein polynomial functions.

The Bézier transformation relies on the definition of Bézier curves [Bézier 1986] and Bernstein functions. Originally, a Bézier curve of order p is a linear combination of $p+1$ univariate Bernstein functions $\mathbf{B}(\widetilde{\xi}) = \left\{B_{i,p}(\widetilde{\xi})\right\}_{i=1}^{p+1}$ associated with control points $\mathbf{P}^{BER} = \{P_{i,p}\}_{i=1}^{p+1}$. Regarding the expression of the Bernstein functions, they are defined directly in the parent space since each Bernstein element has its own parametrization. As with B-Spline functions, multi-variate Bernstein functions are built by applying a tensor product between the univariate functions. From here on, the notations $\mathbf{B}$ and $\mathbf{P}^{BER}$ are extended to denote the functions and control points of a multi-variate Bernstein mesh composed of several elements.

2.2.3 Bézier extraction operator.

By making use of the above knot-insertion process, a Bézier extractor matrix $\mathbf{C}$ can be constructed in order to ensure the link between the B-Spline functions and the Bernstein functions as follows:

$$\mathbf{N} = \mathbf{C}\mathbf{B}. \tag{5}$$

In order to determine the positions of the Bernstein control points the equality between the expression of the B-Spline entity (see Eq. (1)) and the Bernstein one is used. By expressing the B-Spline functions using Eq. (5), it can be shown that $\mathbf{P}^{BER} = \mathbf{C}^T\mathbf{P}^{BS}$.

2.3 The Lagrange extraction

The Lagrange extraction operator directly maps a Lagrange nodal basis onto a B-Spline basis. In the following, the remaining link between Bernstein and Lagrange shape functions is first established and then the direct construction of the Lagrange extractor is presented.

2.3.1 From Lagrange to Bernstein polynomials.

Denoting by $\mathbf{L}$ the classical FE Lagrange functions, the goal is here to build operator $\mathbf{D}_{LB}$ that satisfies:

$$\mathbf{B} = \mathbf{D}_{LB}\mathbf{L}. \tag{6}$$

Obviously, operator $\mathbf{D}_{LB}$ is constructed from its elementary representation $\mathbf{D}^e_{LB}$ which is the same for each element. For illustration purpose, one-dimensional Bernstein and Lagrange shape functions of order 2 are plotted in Figures1(b) and 1(c).

To construct operator $\mathbf{D}^e_{LB}$, one simply needs to express the Bernstein functions as a linear combination of the Lagrange functions at some interpolation points. Making use of the interpolatory property of the Lagrange functions, such an operator can be efficiently constructed by evaluating the Bernstein functions at the nodal points associated to the Lagrange basis.

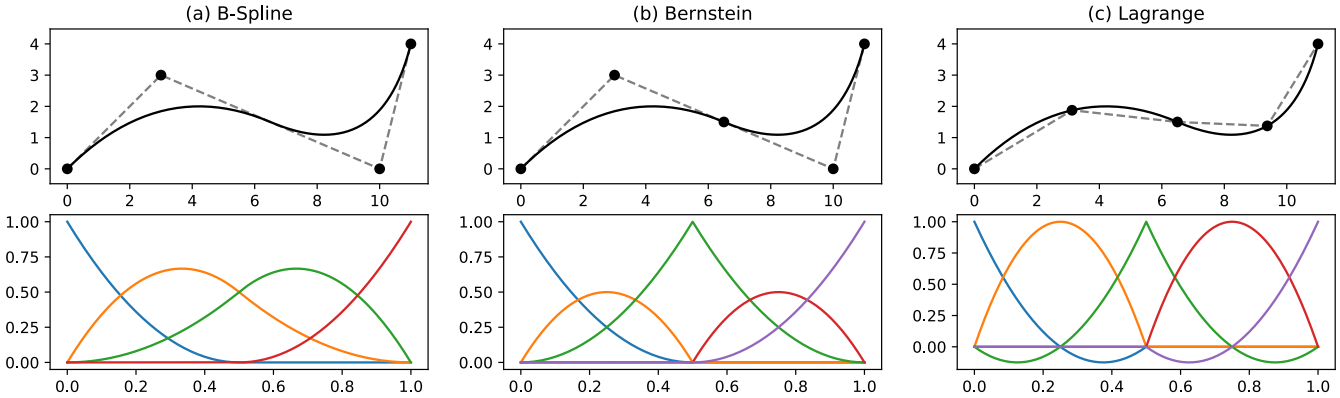


Figure 1: From B-Spline to Lagrange: (a) the initial B-Spline based discretization (four control points associated with four global quadratic B-Spline functions) ; (b) the Bernstein discretization (one control point is added throughout the Bézier decomposition and each element has three local Bernstein shape functions) and (c) Lagrange (FE) discretization (same number of functions as Bernstein, the control points are now on the curve). (top) the curve with the control points and (bottom) the univariate shape functions plots realized in the parameter space ($\xi \in [0;1]$).

Once operator $\mathbf{D}_{LB}$ is built, the same treatment as with the Bézier operator can be performed to construct the FE nodes from the Bernstein mesh: $\mathbf{P}^{FE} = \mathbf{D}_{LB}^T \mathbf{P}^{BER}$. Such nodes could be used to construct the input mesh for classical finite element codes. Obviously, these nodes interpolate the geometry. The B-Spline, Bernstein and Lagrange *control* meshes that generate the same two-element quadratic polynomial beam are plotted in Figures 2(b), 2(c) and 2(d) respectively. As expected, the Bernstein control points are far from the geometry in comparison with the Lagrange nodes that interpolate the geometry.

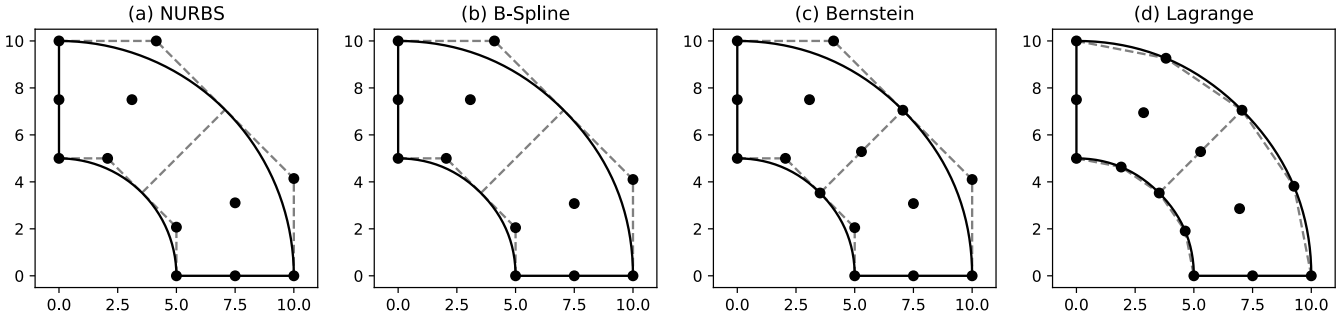


Figure 2: Different representations of a two-element quadratic curved beam analysed in the example section. (a) the NURBS representation using rational functions and weights generates the exact circular geometry; (b) B-Spline, (c) Bernstein and (d) Lagrange control meshes and corresponding geometry which approximate the circles arcs as described in Section 3.2.2.

2.3.2 A direct link between Lagrange and B-Spline functions.

With previous operators $\mathbf{C}$ and $\mathbf{D}_{LB}$ in hand, the computation of the Lagrange extraction operator $\mathbf{D}$ becomes straightforward:

$$\mathbf{N} = \mathbf{D}\mathbf{L} \quad \text{with} \quad \mathbf{D} = \mathbf{C}\,\mathbf{D}_{LB}. \tag{7}$$

Nevertheless, for better numerical efficiency, the Lagrange extraction operator is never computed this way. Indeed, the same procedure as for the Lagrange-Bernstein operator $\mathbf{D}_{LB}$ can be directly applied to the full Lagrange extraction operator $\mathbf{D}$: it merely requires to evaluate the B-spline basis functions at nodal points. An efficient algorithm for this has been proposed in [Schillinger *et al.* 2016].

2.4 The extraction in case of NURBS

Until now, the link between IGA and FEM is restricted to geometries based on polynomial B-Splines. However, in general, spline discretizations consist of rational basis functions that integrate weights associated to control points. The Lagrange extraction can be extended to the case of NURBS. It consists of establishing a link between NURBS and rational Lagrange basis functions, as detailed in [Schillinger *et al.* 2016].

Briefly, it can be extracted from Eqs. (3) and (7) that:

$$\mathbf{R}^{IGA} = \frac{\mathbf{W}^{IGA}\mathbf{D}\mathbf{L}}{W^{IGA}}, \tag{8}$$

where $\mathbf{W}^{IGA}$ is the diagonal matrix of NURBS weights and $W^{IGA} = \sum_{A=1}^{n_A} w_A^{IGA} N_A$ is the NURBS weight function. The NURBS weight function can be rewritten using the Lagrange basis as:

$$\begin{aligned} W^{IGA} &= \sum_{A=1}^{n_A} w_A^{IGA} N_A = (\mathbf{w}^{IGA})^T \mathbf{N} = (\mathbf{w}^{IGA})^T \mathbf{D}\mathbf{L} \\ &= (\mathbf{D}^T \mathbf{w}^{IGA})^T \mathbf{L} = (\mathbf{w}^{LAG})^T \mathbf{D} = W^{LAG}, \end{aligned} \tag{9}$$

where the weights associated to the rational Lagrange control points are:

$$\mathbf{w}^{LAG} = \mathbf{D}^T \mathbf{w}^{IGA}. \tag{10}$$

The rational Lagrange functions are then defined as follows:

$$\mathbf{R}^{LAG} = \frac{\mathbf{W}^{LAG}\mathbf{L}}{W^{LAG}}, \tag{11}$$

where $\mathbf{W}^{LAG}$ is the diagonal matrix of the Lagrange weights. The link between NURBS functions and rational Lagrange functions is finally made using Eqs. (11) and (9) in Eq. (8). Consequently, a new extraction operator $\mathbf{D}_W$ is created as described below:

$$\mathbf{R}^{IGA} = \mathbf{W}^{IGA}\mathbf{D}(\mathbf{W}^{LAG})^{-1}\mathbf{R}^{LAG} = \mathbf{D}_W \mathbf{R}^{LAG}. \tag{12}$$

Therefore, the rational Lagrange control points depend on the NURBS control points: $\mathbf{P}^{LAG} = (\mathbf{D}_W)^T \mathbf{P}^{IGA}$.

Let us emphasize at this stage that this link from rational Lagrange functions to NURBS is exact since both bases are rational and NURBS are of higher-order smoothness. However, one must keep in mind that the definition of these rational Lagrange functions requires, from standard Lagrange polynomials, the incorporation of Lagrange weights $\mathbf{W}^{LAG}$ and of the Lagrange weight function W^{LAG}. Those operations are not part of standard FE codes that are restricted to polynomial basis functions. As shown in Section 3, an additional work thus needs to be achieved to directly go from Lagrange polynomials to NURBS.

3 The non-invasive implementation

Let us recall that our interest is to implement IGA in an available FE software with a reduced level of invasiveness. The potential of the Lagrange extraction technology to allow for the implementation of IGA in standard FE codes has been underlined in [Schillinger *et al.* 2016]. Especially in case of geometries based on polynomial B-Splines, a strategy with minimal invasiveness has been derived. After exposing the current practice for setting up IGA using the previous extraction concepts, the proposed novel implementation procedure is presented.

3.1 A first option

3.1.1 General case of NURBS.

The rational Lagrange geometry that has been created using $\mathbf{D}_W$ (see Eq. (12)) has an elementary structure which makes it more likely to be implemented in a FE software. The resulting implementation scheme and the interaction between the initial NURBS mesh and the computed rational Lagrange mesh are described in Figure 3. On each

branch, the quantities of interest needed to perform the next step are pointed out, for example the global and local rational Lagrange extraction operators $\mathbf{D}_W$ and $\mathbf{D}_W^e$, or the NURBS and rational Lagrange shape functions $\mathbf{R}^{IGA}$ and $\mathbf{R}^{LAG}$. Moreover, the connectivity table used for the matrix assembly is indicated. From this procedure, the following modifications to be done in the FE software can be listed:

1. Modify the standard FE shape functions subroutine to incorporate the Lagrange weights and the weight function and thus, construct the rational Lagrange shape functions from the existing Lagrange polynomials, as expressed in Eq. (11);
2. Apply the extraction for each element using operator $\mathbf{D}_W^e$;
3. Change the connectivity table in order to perform an IGA assembly.

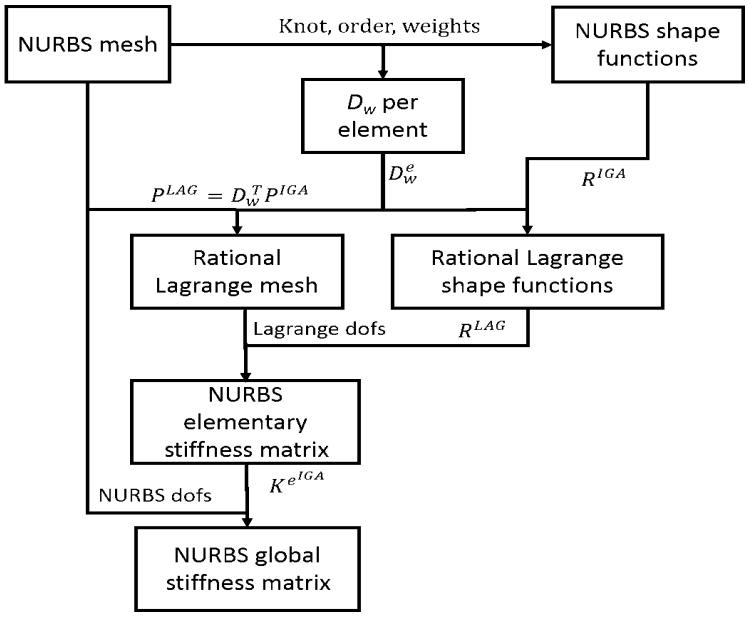


Figure 3: General standard implementation procedure using the Lagrange extraction for NURBS: from a NURBS definition of the geometry and with the creation of operator $\mathbf{D}_W$, a rational Lagrange representation of the geometry is created. By modifying the shape functions subroutine, to express rational Lagrange functions, and applying the local Lagrange extraction, the IGA elementary stiffness matrix is computed. Finally, the assembly subroutine needs to be modified to allow the use of the NURBS degrees of freedom to get the global NURBS stiffness matrix.

3.1.2 Specific case of B-Splines.

In the particular case where the geometry is generated only with B-Spline polynomials (no rational basis functions), the standard FE subroutines do not need to be touched at the element level. In other words, point 1 above does not stand anymore. Indeed, after computing the standard FE elementary stiffness matrices $\mathbf{K}^{e^{FE}}$, only additional matrix-matrix products are required to obtain the corresponding IGA elementary matrices $\mathbf{K}^{e^{IGA}}$:

$$\mathbf{K}^{e^{IGA}} = \mathbf{D}^e \mathbf{K}^{e^{FE}} \mathbf{D}^{e^T}. \tag{13}$$

This is made possible since a direct link is established between the Lagrange polynomials that are present in standard FE software and the smooth B-Spline functions. As a result, part of our work now consists in elaborating a direct link between NURBS and Lagrange polynomials to meet such a non-invasiveness in the case of rational functions as well.

3.2 The proposed strategy

3.2.1 Principle.

As stated above, no modification of the whole FE routines is envisaged in this work (*i.e.*, no modification into the shape functions subroutine (point 1 above) and concerning the assembly (point 2 above)). In order to do so, a global view of the extraction is adopted, as depicted in Figure 4. The path starting with a B-Spline mesh is exact and mainly consists of applying procedure exposed in [Schillinger *et al.* 2016], but in a global way. The path related to NURBS however requires the construction of an additional operator to go from polynomials to rational functions. Such a transformation cannot be exact since this is the space of the rational functions that includes the associated polynomials and not the other way around. A projection thus needs to be performed. For simplicity, the projection is performed on the Lagrange side, *i.e.*, the idea is to project the rational Lagrange discretization onto the polynomial Lagrange space. Operator $\mathbf{D}_{LL}$ is introduced as:

$$\mathbf{R}^{LAG} = \mathbf{D}_{LL}\mathbf{L}, \tag{14}$$

and its construction is explained in the next paragraphs.

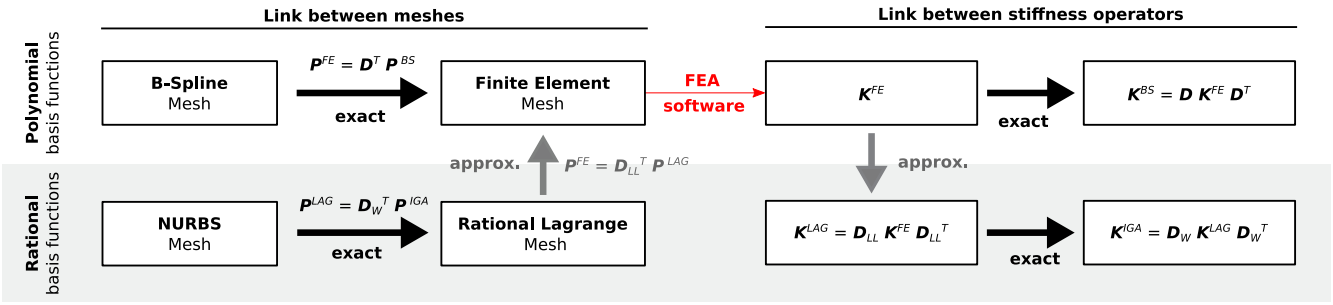


Figure 4: Approach to link a NURBS mesh to a FE mesh using different global operators. Those operators are then used to recover the NURBS stiffness matrix from the FE one computed using a classical FE software, taken as a black-box. Black arrows represent an exact link, gray arrows an approximation.

Once the finite element mesh is created from the IGA mesh in a pre-processing step, it is used as an input for the classical FE code in order to compute the FE stiffness matrix. With the different elaborated operators, successive transformations are then performed to obtain the final isogeometric stiffness matrix, as follows:

$$\mathbf{K}^{IGA} = \mathbf{D}_W\mathbf{D}_{LL}\mathbf{K}^{FE}\mathbf{D}_{LL}^T\mathbf{D}_W^T. \tag{15}$$

This transformation is also applied on the right-hand side:

$$\mathbf{F}^{IGA} = \mathbf{D}_W\mathbf{D}_{LL}\mathbf{F}^{FE}. \tag{16}$$

Consequently, the system $\mathbf{K}^{IGA}\mathbf{U}^{IGA} = \mathbf{F}^{IGA}$ can be solved to obtain the displacement $\mathbf{U}^{IGA}$. Finally, note that the resulting IGA displacement can be back-converted in terms of nodal displacements:

$$\mathbf{U}^{FE} = \mathbf{D}_{LL}^T\mathbf{D}_W^T\mathbf{U}^{IGA}, \tag{17}$$

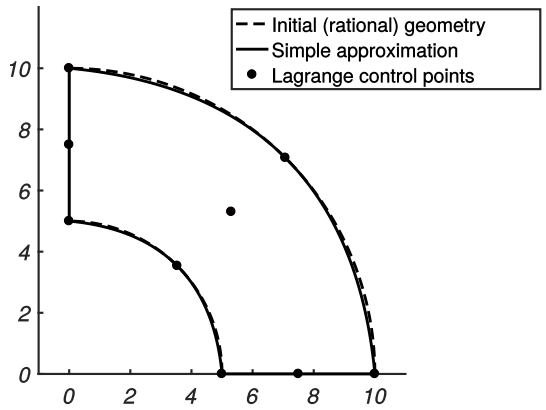
so that existing subroutines of the FE code can be used for post-processing.

Remark 2 *Notice that Eqs. (15) and (16) can be interpreted as projecting the FE system onto the isogeometric basis. As a result, the developed algebraic bridge* $(\mathbf{D}_W\mathbf{D}_{LL})$ *provides a new point of view on the relation between IGA and FEA: IGA can be interpreted as the projection of FEA on a specific regular reduced basis.*

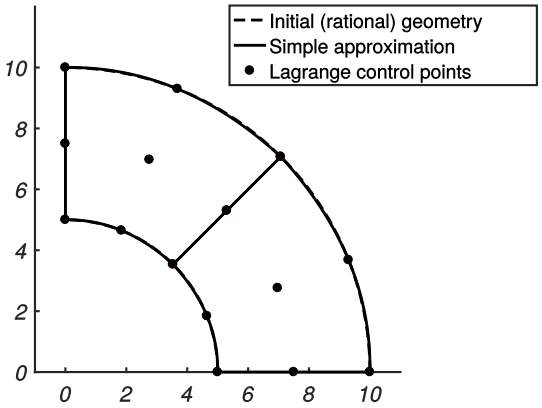
3.2.2 Simple approximation.

Acting at the Lagrange level for performing the projection between the rational and associated polynomial spaces offers the opportunity to follow a pragmatic yet accurate strategy. Indeed, it has to be emphasized that the

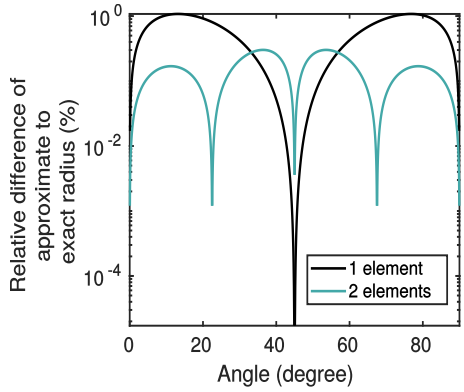
control points of the rational Lagrange discretization interpolate the geometry. As a result, it is possible to simply consider that the position of the FE nodes $\mathbf{P}^{FE}$ is exactly the same as the position of the rational Lagrange control points $\mathbf{P}^{LAG}$. In this case, $\mathbf{D}_{LL}$ formally reads as the identity operator (denoted by $\mathbf{I}_D$ in the following), so that the derived procedure does not add any extra-computational effort from the current practice. As an illustration, Figure 5 investigates the difference in terms of geometry in case of a quarter circular beam. It can be observed that the approximation is already very accurate for a single element (Figure 5(a)) and, obviously, it is improved through the refinement of the mesh since more interpolated control points are added (Figure 5(b)). The accuracy of the approximation technique can also be appreciated through Figure 5(d) that displays the geometry error in terms of the relative difference in radius from the exact value evaluated along the circular annulus. Given this high accuracy, it is expected that the error related to the mapping $\mathbf{D}_{LL} = \mathbf{I}_D$ appears largely insignificant compared to that associated to the finite element resolution of an underlying mechanical problem.



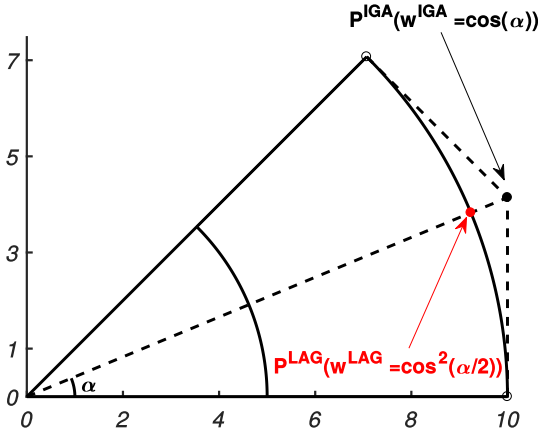
(a) Simple approximation for one element.



(b) Simple approximation for two elements (in the arc direction).



(c) Geometry error in terms of the relative difference in radius between the simple approximation and the exact circle.



(d) Weight evolution with refinement. The weight of the rational Lagrange control point P^{LAG} is equal to $\cos^2(\alpha/2)$, which quickly converges to 1 as the mesh is refined.

Figure 5: The simple approximation in case of a quarter circular beam. The difference in terms of geometry quickly vanishes with the refinement of the mesh.

An alternative interpretation can be given: the strategy simply consists of considering all the weights of the rational Lagrange discretization equal to one. Taking back the example of a quarter circular beam, the approximation can be illustrated as follows (see Figure 5(d)). As explained in the NURBS technology [Cottrel *et al.* 2009, Piegl and Tiller 1997], the weight of the middle point of a quadratic circular one-element arc is equal to the cosine of half of the angle subtended by the arc (the weights of the two boundary control points being one). From here on, α denotes the angle of interest (see Figure 5(d) again). Making use of previous equality (10), then expressing

operator $\mathbf{D}$ using (7), and finally noting that $\mathbf{C} = \mathbf{I}_D$ in case of a mesh composed of a single NURBS element, the weights associated to the rational Lagrange discretization of this circular arc can be computed as follows:

$$\mathbf{w}^{LAG} = \left(\mathbf{D}_{LB}^{e^{1D}}\right)^T \mathbf{w}^{IGA}, \tag{18}$$

with $\mathbf{w}^{IGA} = \begin{bmatrix} 1 \\ \cos(\alpha) \\ 1 \end{bmatrix}$ and $\mathbf{D}_{LB}^{e^{1D}}$ the 1D version of $\mathbf{D}_{LB}^{e}$ (6), which ends up with the weight:

$$w^{LAG} = \frac{1}{2}(1 + \cos(\alpha)) = \cos^2\left(\frac{\alpha}{2}\right) \tag{19}$$

for the middle point of the rational Lagrange parametrization. Therefore, the good accuracy of the simple approximation is accounted for: when α decreases $\cos^2(\alpha/2)$ quickly tends towards one. Note finally that going from Bernstein polynomials to rational Bézier functions with the same treatment would lead to a convergence speed of $\cos(\alpha)$ instead, and going from B-Spline polynomials to NURBS functions similarly would converge even less quickly.

3.2.3 Local least-square approximation.

As a more rigorous mathematical approach, a strategy based on a projection using a L^2-norm is also proposed to approximate the rational Lagrange functions by means of linear combinations of Lagrange polynomials. The presentation of the method is restricted to the two-dimensional setting. Given the local structure of rational Lagrange functions as well as Lagrange polynomials, the methodology proceeds element-by-element for the construction of operator $\mathbf{D}_{LL}$, thus providing simplicity and limiting the computational cost. Note that similar works have been made in quite related contexts, such as to project C^0 discretizations to smoother basis (see, *e.g.*, [Thomas *et al.* 2014] and [Schillinger *et al.* 2016] that build Bézier and Lagrange projection, respectively). Here, the following simple treatment can be performed. To start with, the vertex Lagrange nodes of the considered element are taken equal to the corresponding vertex rational Lagrange control points. Then, in order to be directly compatible with the neighboring elements, the projection is formulated by writing a least-square problem on each element edge. As a result, the developed strategy does not require the assembly step encountered in common local least-square procedures (see, *e.g.*, [Govindjee *et al.* 2012]) in order to combine in some way the different values obtained from the different elements for the same point.

More precisely, let us consider an elementary rational Lagrange geometry expressed similarly as in Eqs (1) and (4):

$$\mathcal{S}^{e^{LAG}} = \mathbf{P}^{e^{LAG}{}^T} \mathbf{R}^{e^{LAG}}, \tag{20}$$

where $\mathbf{P}^{e^{LAG}}$ is a matrix that collects the positions of the control points related to the considered rational Lagrange element and $\mathbf{R}^{e^{LAG}}$ is the vector of the elementary rational Lagrange functions. The objective is to determine the positions of the equivalent Lagrange nodes $\mathbf{P}^{e^{FE}}$ that give an accurate approximation of the initial rational Lagrange geometry $\mathcal{S}^{e^{LAG}}$. Considering an element edge defined by the same second coordinate $\tilde{\eta}$ in the parent space, the quadratic functional J is minimized such that:

$$J = \frac{1}{2} \int_{\tilde{\xi}=-1}^{1} \left(\mathcal{C}^{e^{LAG}}(\tilde{\xi}) - \mathcal{C}^{e^{FE}}(\tilde{\xi})\right)^2, \tag{21}$$

to compute the equivalent Lagrange nodes related to the considered edge. Let us emphasize that the curves $\mathcal{C}^{e^{LAG}}(\tilde{\xi})$ and $\mathcal{C}^{e^{FE}}(\tilde{\xi})$ only involve the one-dimensional rational Lagrange functions and Lagrange polynomials, respectively. After differentiation, we end up with the following linear system that relates each of the spacial coordinates of the edge Lagrange nodes ($X^{e^{FE}}$) with that of the edge rational Lagrange control points ($X^{e^{LAG}}$):

$$\mathbf{M}_{1D}^{e} X^{e^{FE}} = \mathbf{M}_{R}^{e} X^{e^{LAG}}, \tag{22}$$

where $\mathbf{M}^e_{1D}$ is simply the elementary mass matrix (with unit density) of the 1D Lagrange polynomials and $\mathbf{M}^e_R$ reads as follows:

$$\begin{bmatrix} \int_{\tilde{\xi}=-1}^{1} L_1(\tilde{\xi})R_1^{LAG}(\tilde{\xi}) & \int_{\tilde{\xi}=-1}^{1} L_1(\tilde{\xi})R_2^{LAG}(\tilde{\xi}) & \int_{\tilde{\xi}=-1}^{1} L_1(\tilde{\xi})R_3^{LAG}(\tilde{\xi}) \\ \int_{\tilde{\xi}=-1}^{1} L_2(\tilde{\xi})R_1^{LAG}(\tilde{\xi}) & \int_{\tilde{\xi}=-1}^{1} L_2(\tilde{\xi})R_2^{LAG}(\tilde{\xi}) & \int_{\tilde{\xi}=-1}^{1} L_2(\tilde{\xi})R_3^{LAG}(\tilde{\xi}) \\ \int_{\tilde{\xi}=-1}^{1} L_3(\tilde{\xi})R_1^{LAG}(\tilde{\xi}) & \int_{\tilde{\xi}=-1}^{1} L_3(\tilde{\xi})R_2^{LAG}(\tilde{\xi}) & \int_{\tilde{\xi}=-1}^{1} L_3(\tilde{\xi})R_3^{LAG}(\tilde{\xi}) \end{bmatrix}. \tag{23}$$

From system (22), the position of the middle Lagrange node alone is actually picked since, as stated above, the vertex nodes are taken equal to the vertex Lagrange control points. The procedure is then repeated to all element edges. Unlike operator $\mathbf{D}^e_{LB}$, note that $\mathbf{D}^e_{LL}$ is not the same for each element, as it depends on the weights of the control points of the considered rational Lagrange element. However, let us emphasize that those weights only appear on the right-hand side of the local least-square problem (through $\mathbf{M}^e_R$, see Eq. (23)). As a consequence, $\mathbf{M}^e_{1D}$ is analytically computed and inverted once and for all elements:

$$(\mathbf{M}^e_{1D})^{-1} = \begin{bmatrix} 9/2 & -3/4 & 3/2 \\ -3/4 & 9/8 & -3/4 \\ 3/2 & -3/4 & 9/2 \end{bmatrix}. \tag{24}$$

The only requirement of the developed procedure is actually to compute $\mathbf{M}^e_R$ on each element, that is, to perform a numerical integration in the parent space of an element edge. This is straightforward using standard numerical analysis packages.

Still on a quarter circular arc, Figure 6 shows the difference in terms of geometry encountered when applying the proposed local least-square procedure for a mesh composed of a single element. From a global point of view, this more rigorous alternative seems to produce equivalent results (at least in terms of geometry) as the previous pragmatic strategy that considers all the weights equal to one. This accounts for the validity of the previous simple approximation in practical (engineering) applications.

4 Examples with elasticity models

As a first assessment of the method, numerical experiments with two and three-dimensional linear elasticity models are carried out in this section. For validation purpose, the first test-case (for which a reference analytical solution is available) is computed using a *Matlab* homemade code. Then, for proof-of-concept, the open source package *Code_Aster*, which is developed by the EDF R&D company [Code_Aster 2014], is used as an industrial FE software for the second benchmark. It may be noticed that no unit is given for the following test-cases (as it is usually the case in the literature). The given values are those that are directly taken as input for the numerical process.

4.1 2D circular beam under end shear

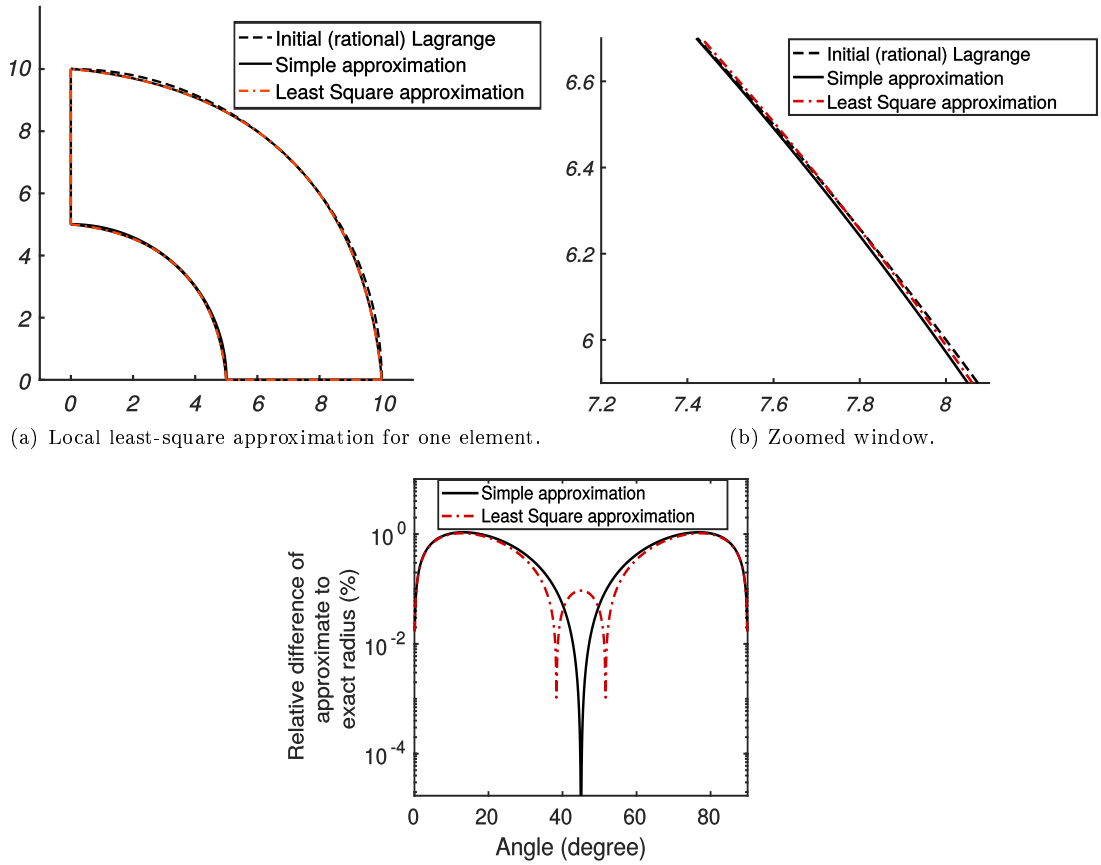
The first example consists of a 2D circular beam under plane stress subjected to end shear, as illustrated in Figure 7. The material properties are the following: Young Modulus, $E = 10000$ and Poisson ratio, $\nu = 0.25$. A radial constant displacement $u_0 = 0.01$ is prescribed on the bottom edge of the beam. The closed form solution in terms of strain energy can be found in [Zienkiewicz *et al.* 2003].

In Figure 8, the evolution of the relative error regarding the strain energy is given as a function of the number of degrees of freedom. The relative error is computed as:

$$\frac{|E_d^{ex} - E_d^{fe}|}{|E_d^{ex}|}, \tag{25}$$

where E_d^{ex} denotes the reference exact strain energy and E_d^{fe} the strain energy of the discrete model.

To start with, it can be noticed that the results for the 9-node element created with the simple or the least-square projections are similar. This behavior was expected because, as it has been previously illustrated from a



(a) Local least-square approximation for one element.

(b) Zoomed window.

(c) Geometry error in terms of the relative difference in radius.

Figure 6: The local least-square approximation in case of a quarter circular beam. It consists of an element-level projection technology which finds the best position of the Lagrange nodes that generates a geometry close to the Rational Lagrange one.

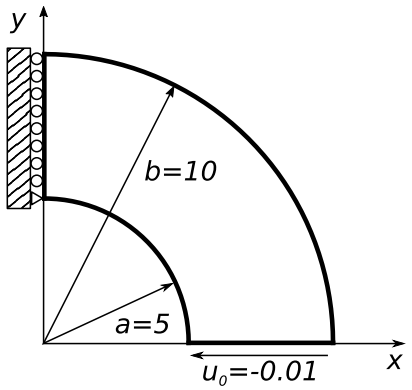


Figure 7: Problem description of the circular beam under end shear.

geometrical viewpoint, the simple approximation was already good for one element in the arc direction. Here, with the first refinement used, there are already 6 elements in this direction; therefore, the error related to the mapping $\mathbf{D}_{LL} = \mathbf{I}_D$ is largely insignificant compared to the associated FE error. The analysis performed with the FE mesh created by both of these approximation methods gives an error evolution similar to the FEM reference [Zienkiewicz *et al.* 2003]. Then, in order to obtain the IGA stiffness matrix, and thus to recover the isogeometric

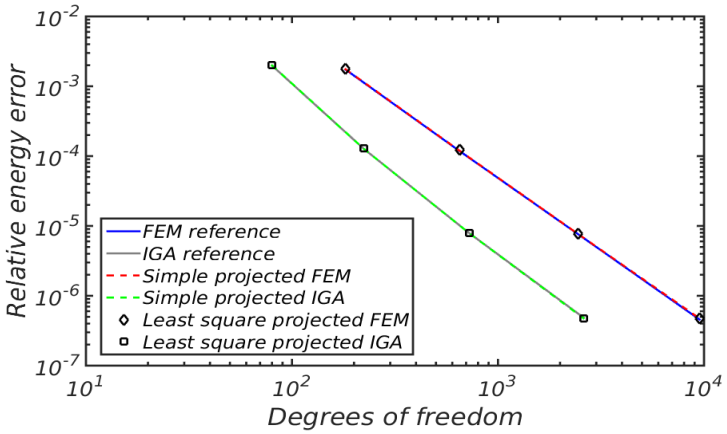
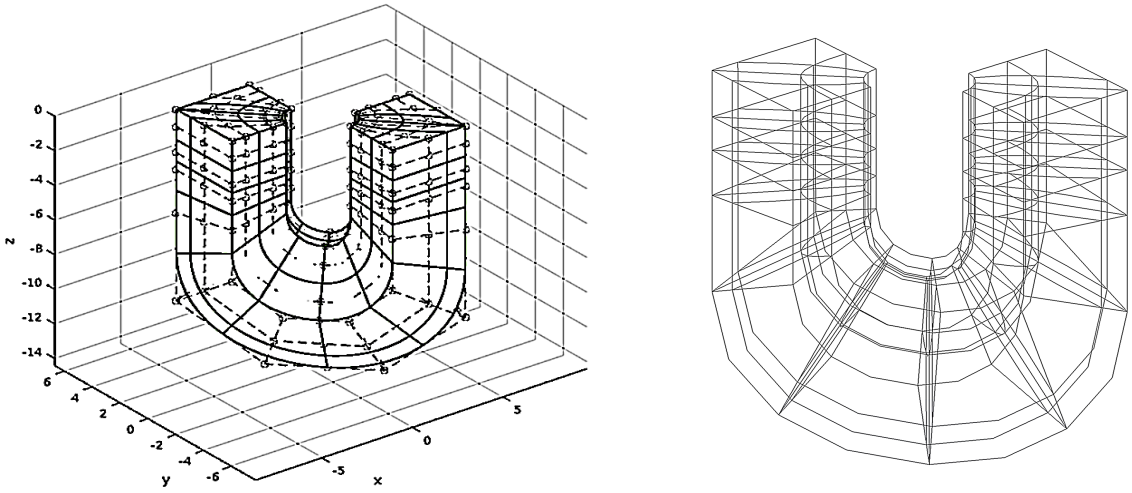


Figure 8: Evolution of the relative energy error of the circular beam. Full FEM [Zienkiewicz *et al.* 2003] and IGA resolutions are used as references. On the one hand, the error is calculated by using the simple approximation between rational Lagrange functions and Lagrange polynomials (see section 3.2.2) for the resulting FE problem (curve "Simple projected FEM") and back-projected IGA problem (curve "Simple projected IGA"). On the other hand, this error is computed for the FE and IGA problems with a least-square approximation between rational Lagrange functions and Lagrange polynomials (see section 3.2.3, associated curves "Least-square projected IGA and FEM").

results, the FE stiffness matrices are back-projected using the developed global bridge. The errors found with the simple and least-square projected IGA are similar to the error of a full IGA reference computation (the curves "IGA reference", "Simple projected IGA" and "Least-square projected IGA" are superimposed). This accounts for the accuracy of the developed global bridge in case of NURBS. Finally, one can notice that for a given mesh refinement, the errors are about the same for FEM and IGA. The main difference is the number of degrees of freedom which significantly decreases when IGA is performed (the two curves are simply horizontally translated). This gap between the two curves illustrates the increased per-degree-of-freedom accuracy of IGA (reduction by a factor of more than 3 here) which has been deeply demonstrated in the literature (see, *e.g.*, [Evans *et al.* 2009, Cottrell *et al.* 2006, Schillinger *et al.* 2013]).

4.2 Solid horseshoe

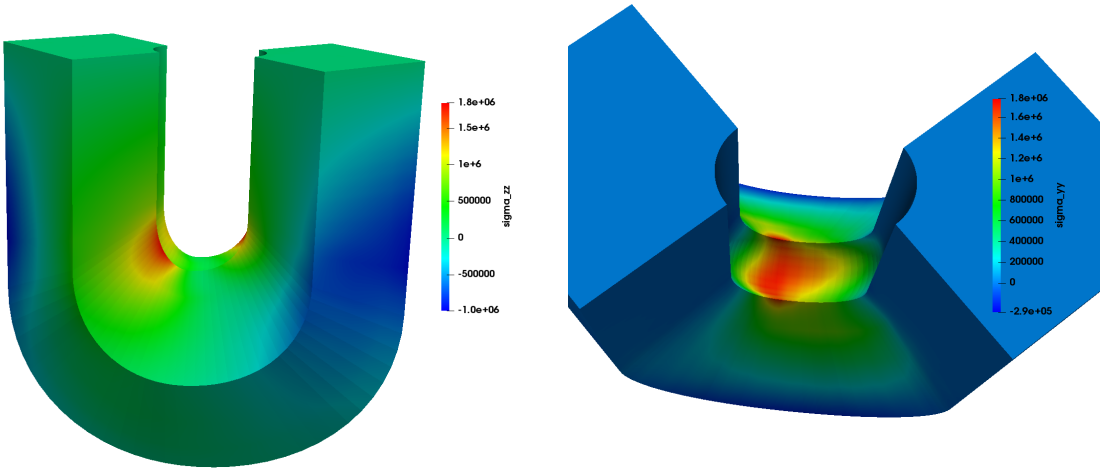
In this section, a solid horseshoe subjected to equal and opposite displacements on the top surfaces is used as a full 3D example (see, *e.g.*, [Hughes *et al.* 2005] for similar computations). The geometry and the NURBS mesh associated with the solid horseshoe are given in Figure 9(a). Let us emphasize that it involves a multi-patch model which can be easily handled using the proposed methodology. The NURBS mesh is composed of 1152 elements of degree 2 with 7020 degrees of freedom. The material properties are the following: $E = 3 \times 10^7$ and $\nu = 0.3$. An initial displacement $\mathbf{u}_0 = \mathbf{y}$ is applied to the top-left surface and a similar displacement $\mathbf{u}_0 = -\mathbf{y}$ is applied to the top-right surface. In addition, the top surface displacement across the x-direction is locked whereas for the z-displacement only the corner points on each top surface are locked. This choice of boundary conditions makes the loading asymmetric. Using the developed global bridge, a corresponding finite element mesh, composed of hexahedral elements with 27 nodes, is created (see Figure 9(b)). The FE mesh is composed of 33507 degrees of freedom. It is used as an input mesh in the industrial FE software. After recovering and solving the IGA linear system from the FE one constructed in *Code_Aster*, the obtained IGA displacement field can be back-converted in terms of nodal displacements (see Eq. (17)) and the pre-processing routines of the FE code can be used to plot the NURBS stress, see Figure 10(a). A zoom on the area of stress concentration of σ_{yy}, due to the asymmetric loading, is made in Figure 10(b). Results are smooth and in good agreement with reference IGA computations [Hughes *et al.* 2005], which means that the developed global bridge does not introduce significant errors for the projection.



(a) Initial NURBS definition of the horseshoe geometry

(b) Associated constructed FE mesh.

Figure 9: Example of coarse meshes of the horseshoe.



(a) Stress σ_{zz} over the global surface

(b) Detail on the localized stress σ_{yy} due to the asymmetric loading.

Figure 10: Contour plot of the stress for the solid horseshoe (simple projected IGA).

5 Extension to nonlinear analysis

Making IGA more accessible for industrial applications requires not to touch optimized routines which perform the integration of the nonlinear mechanical behavior. In this part the case of a nonlinear problem is considered with minimal invasiveness (no modification into the nonlinear solver). The only requirement of the method is to be able to extract the tangent stiffness and right-hand-side operators at each iteration of the nonlinear solver.

5.1 Implementation of the method

In the nonlinear case, the resolution of the tangent problem is performed within a Newton solver which is still realized by the commercial FE code. The method is thus non-invasive also with respect to the options of the nonlinear solver. Technically, it requires additional features of the FE code, which consist in being able to pause the nonlinear resolution in order to externalize only the resolution of the global tangent system; then, to re-inject the displacement field solution; and finally, to restart the nonlinear resolution without any other external treatments. The method is implemented here using *Code_Aster* software which offers this possibility thanks to Python subroutines and the so-called *STAT_NON_LINE in splitted commands* solver.

More precisely, Figure 11 shows how the method works in a nonlinear context. After the pre-processing that does not change with respect to the elastic case, the solver *STAT_NON_LINE in splitted commands* enters the Newton loop. The nonlinear behavior is integrated and the tangent problem assembled with the optimized Fortran

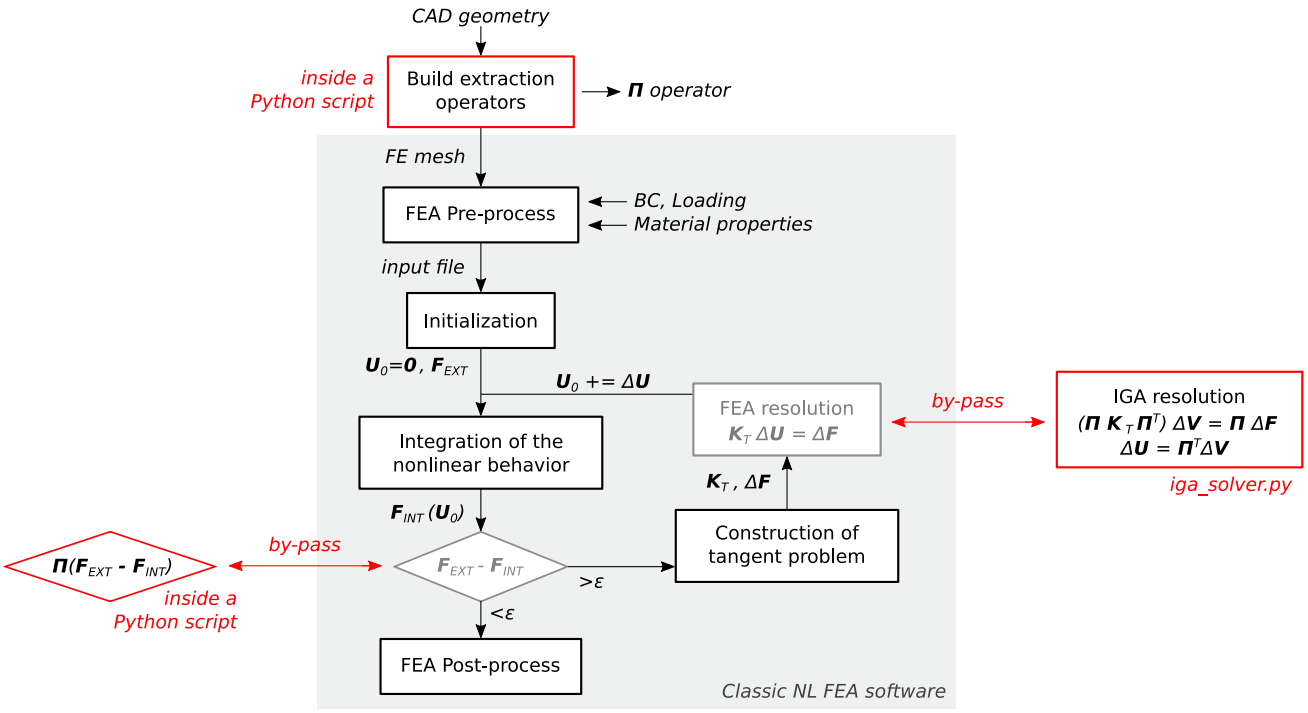


Figure 11: Flowchart of the non-invasive and nonlinear implementation in an existing FE software, taken as a black-box. Note that operator $\mathbf{\Pi}$ is introduced to define the full bridge between IGA and FEA (*i.e.*, $\mathbf{D}$ for B-Splines or $\mathbf{D}_W\mathbf{D}_{LL}$ for NURBS).

routines of *Code_Aster*. On the other hand, the resolution of the tangent linear system is replaced by a Python subroutine, which consists in solving its projection onto the subspace spanned by $\mathbf{\Pi}$. Once this resolution is made, the IGA displacement solution vector is transferred on the FE space so that it can be re-introduced in *Code_Aster* in order to update the behavior. It is worth noticing that a second very short Python subroutine is needed for the calculation of the IGA residual from the FE one.

Some elements of these subroutines are provided hereafter for the implementation in *Code_Aster*. For instance, the subroutine `iga_solver.py` is roughly organized as follows:

- get the Aster concept which corresponds to the matrix and the right-hand side and convert to numpy arrays:
  `Kef = sparse.csr_matrix(STIFF.EXTR_MATR())`
  `Fef = FORCE.EXTR_COMP().valeurs`
- projection of the global system and resolution:
  `Uiga = scipy.sparse.linalg.spsolve( Pi.T.dot(K.dot(Pi)) , Pi.T.dot(F) )`
- reconstruction of the FE solution vector
  `Uef = Pi.dot(Uiga)`

This Python subroutine is called in the COMM file of *Code_Aster* using the command below:

```
from iga_solver import *
Uef=iga_solver(STIFF,FORCE)
```

Finally, in the COMM file, an Aster concept has to be built from the displacement `Uef` using `CREA_TABLE(...)` and `CREA_CHAMP(...)` functions.

Remark 3 *The application programming interface of* Code_Aster *being* `Python`*, it is possible in this particular case to avoid reading/writing the linear operators on the drive, as it may be required with other industrial software. The operators are kept in memory in sparse format. The algorithm simply requires to project the FE tangent system and FE residual on an ad-hoc reduced basis. This operation is efficiently performed using a dedicated sparse Library* `Scipy.Sparse`*. Note that this kind of projection is classically used in many model order reduction techniques (Proper Orthogonal Decomposition, Reduced Basis, see,* e.g. *[Kerfriden et al. 2012, Quarteroni et al. 2016, Chinesta et al. 2017]).*

Remark 4 *It may also be noticed that our method appears compatible with previous works performed regarding the parallel computation of nonlinear problems using Code_Aster (see,* e.g, *[Duval et al. 2016, Oumaziz et al. 2019]). As a result, the use of our strategy in combination with such works should allow the parallel computation of nonlinear IGA using Code_Aster.*

5.2 Application to elastoplasticity

For illustration purpose, a 3D dog-bone sample in tension is considered in this section. Such an experimental test is often used to characterize the hardening of metallic materials [Mathieu *et al.* 2015]. The specimen is 100mm long, the ligament is 5 mm wide and the sample is 2.5 mm thick, as shown in Figure 12 (top). An elastoplastic constitutive behavior is considered with a Von Mises mixed hardening (see Figure 12 (bottom)). The parameters are the Young modulus E=22.13GPa and Poisson ratio ν=0.3. The Prager constant is set to 2200MPa and the nonlinear hardening is based on the tensile stress-strain curve given in Figure 12. The sample is subjected to a remote tension: the load goes from p = 0MPa to p = 80MPa within 10 (non-uniform) increments. With such parameters, the nonlinearity is easily computed using very few iterations of the Newton-based solver at each loading step.

Remark 5 *To prescribe the Dirichlet conditions, Code_Aster uses the double Lagrange multiplier method (see [Charras et al. 1993]) to avoid losing the positive-definiteness of the operator. The associated Lagrange multipliers are used in the nonlinear solver of Code_Aster to update the behavior. Although this does not represent any theoretical limitation, for the sake of simplicity, a Neumann problem is considered here, so that the multipliers vanish.*

The FE mesh is built from the CAD as described in Section 3. Even if, in this case, only one element was used along the sample thickness, the number of degrees of freedom is significantly reduced: 6561 for the IGA versus 23409 for the associated finite element mesh, see Fig. 13.

The longitudinal component of the displacement field solution is depicted in Figure 14 (left). The Von Mises stress field obtained with the non-invasive IGA method (*STAT_NON_LINE in splitted commands* + subroutine Python) is also presented in Figure 14 (center). It can be seen that, in the region of the ligament, this field has values above the yield stress ($\approx$ 200MPa) which illustrates that in its central part, the specimen has undergone plastic deformations. This IGA stress field is compared to the Von Mises stress field obtained using the input FE mesh and the nonlinear FE solver *STAT_NON_LINE* of *Code_Aster* with the same mesh. Let us notice that

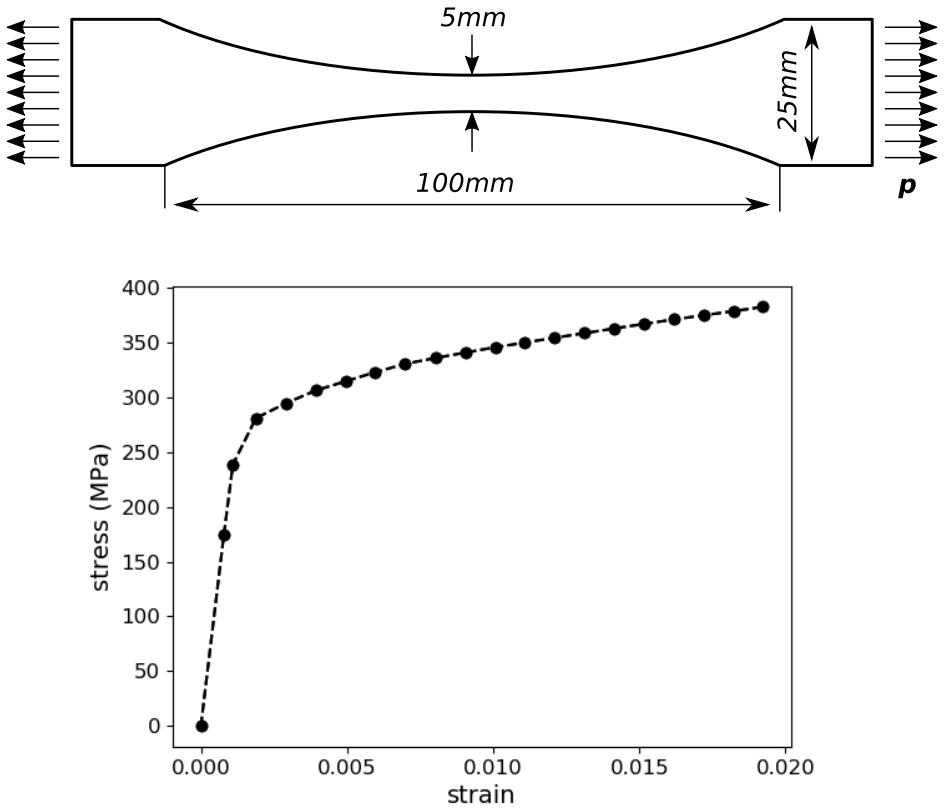


Figure 12: Problem description of the elastoplastic dog-bone sample in tension.

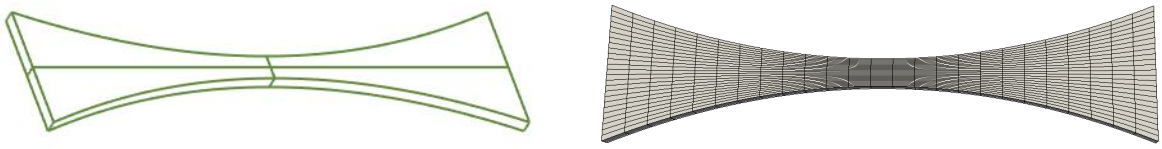


Figure 13: Initial IGA mesh (left) and refined automatically generated FE mesh (right).

no significant difference regarding the convergence speed of the nonlinear solver has been observed between the IGA and FE versions for this test-case. Since the converged solutions are very close, the relative discrepancy in Von Mises stress between the IGA and FE solutions is plotted in Figure 14 (right). Despite a reduction of more than 70% in the number of degrees of freedom, the solutions are very close (less than 2% of local mismatch while the two solutions come from different approximation subspaces), which confirms, in the nonlinear framework, the observations that had been made in elasticity.

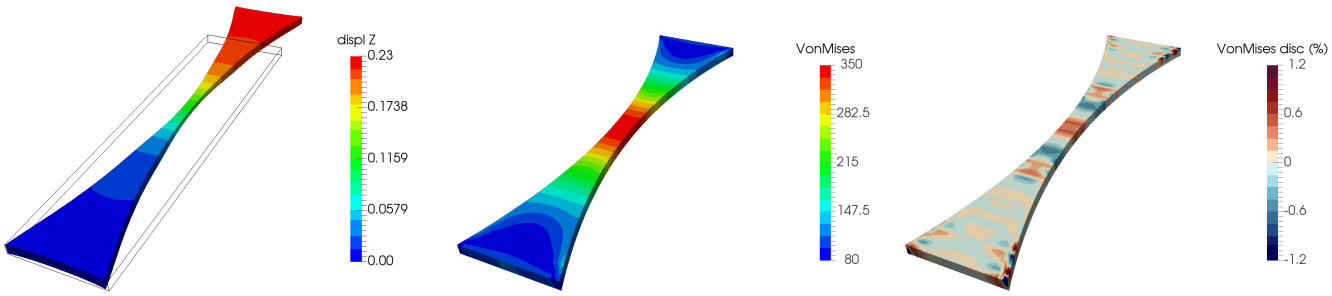


Figure 14: Accuracy of the proposed non-invasive IGA implementation: Longitudinal displacement field (left, amplification 100) ; Von Mises stress field (center); and relative discrepancy in Von Mises stress from the FE reference solution computed using *Code_Aster* (right). The seventh increment, which corresponds to $p = 70$MPa, is considered for the plot.

In order to quantitatively assess the proposed non-invasive IGA method with respect to the classical FE method, the tensile force-displacement curve is eventually plotted in Figure 15. Again, despite its much smaller dimension,

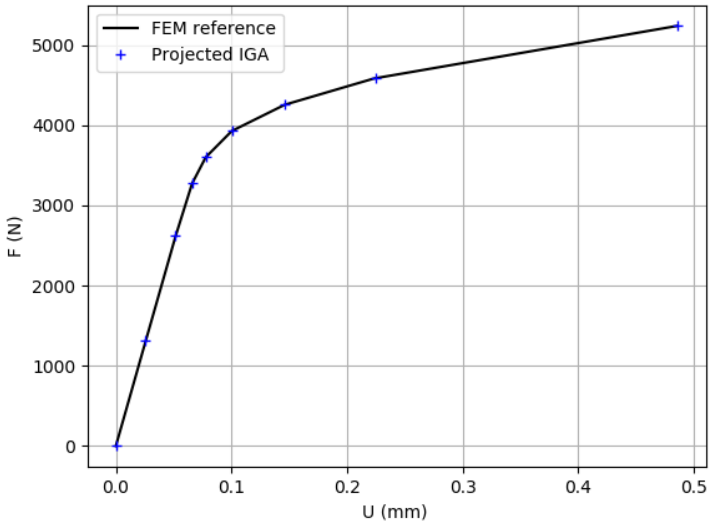


Figure 15: Force versus displacement curve. Comparison between *Code_Aster* FE reference (solid black line) and non-invasive IGA (blue crosses).

the IGA subspace is able to provide a solution almost identical to that generated by FEM. This result is in line with an interpretation of the IGA as the projection of FEA on a specific regular reduced basis (see Remark 2).

6 Conclusions

In this work, a novel implementation procedure was developed that further simplifies the integration of IGA in well-established FE environments, the ultimate goal being to make IGA more accessible for industrial applications. The attractive property of our approach is that it exhibits a reduced level of invasiveness. More precisely, in the proposed implementation, the whole FE routines remain completely untouched. As a result, the total FE code that may integrate complex, nonlinear numerical models and optimized routines can be used as a black box. The only requirements is that the commercial code (a) is able to output the tangent stiffness and right-hand side operators and (b) has full quadratic elements (the 9-node quadrilateral element in 2D, or the 27-node cubic element in 3D) so that it becomes possible to recover and solve, in an external script, the isogeometric system from the FE one. In that sense, the proposed procedure lies in the family of *non-invasive* methods whose interest for the transfer of advanced technologies to the industrial community has been proved these last years [Duval *et al.* 2016].

The key aspect of our approach was to adopt a global vision regarding the link between IGA and FEM. Starting with a global formulation of the Lagrange extraction operator [Schillinger *et al.* 2016], it was shown that it is possible to perform non-invasive B-Spline based IGA in a real industrial FE code, which already constitutes a progress from a practical point of view. Then, to meet a similar reduced level of invasiveness in the case of rational functions as well, a global approximate link to go from standard nodal Lagrange polynomials to NURBS functions was developed. An additional operator that enables to take into account the weights of the rational Lagrange control points was constructed, which enabled to perform non-invasive NURBS-based IGA. The performance of the implementation was first illustrated through a series of numerical experiments involving 2D and 3D elasticity problems and the use of the industrial FE software *Code_Aster* developed by the EDF R&D company [Code_Aster 2014]. According to the authors knowledge, this is the first time that IGA has been implemented in *Code_Aster*. Then, the potential of the method was assessed in the nonlinear regime. None of the routines concerning the nonlinearity has been touched: at each iteration of the nonlinear solver, the FE tangent system and the FE nonlinear residual were simply projected on an *ad-hoc* reduced basis to obtain their isogeometric counterparts.

From a general point of view, this work further contributes to bridging the gap between IGA and standard FEM. Indeed, the developed full algebraic bridge going from Lagrange polynomials to (possibly rational) isogeometric functions offers a new lighting on the relation between IGA and FEA. IGA can be interpreted as the projection of FEA on a specific regular reduced basis. Pushing this interpretation a step further, the implementation procedure may also appear of interest in the context of reduced-order modeling [Kerfriden *et al.* 2012, Quarteroni *et al.* 2016, Chinesta *et al.* 2017]: it could be used to implement any reduced order model in a non-invasive way from a standard FE code.

Acknowledgements

The authors would like to thank Paul Oumaziz, from Núcleo Científico Multidisciplinario-DI, Universidad de Talca, Chile, for his valuable help regarding the implementation of the method in *Code_Aster*.

References

[Hughes *et al.* 2005] 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.

[Cottrel *et al.* 2009] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. In D. Kroner, C. Rohde, and M. Ohlberger, editors, *Isogeometric Analysis: Toward Integration of CAD and FEA*, Lecture Notes in Computational Science and Engineering. Springer, Berlin, 2009. ISBN: 978-0-470-74873-2.

[Cohen *et al.* 1980] 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.

[Piegl and Tiller 1997] L. Piegl and W. Tiller. The NURBS Book (Monographs in Visual Communication), second ed., *Springer-Verlag*, New York, 1997.

[Bazilevs *et al.* 2010] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R Hughes, S. Lipton, M.A. Scott and T.W.Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229-263, 2010.

[Evans *et al.* 2009] 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.

[Lipton *et al.* 2010] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj and T.J.R Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357-373, 2010.

[Schillinger *et al.* 2013] D. Schillinger, J.A. Evans, A. Reali, M.A Scott and T.J.R Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering*, 267:170–232, 2013.

[Yin *et al.* 2015] S. Yin , T. Yu , T.Q. Bui , M.N. Nguyen. Geometrically nonlinear analysis of functionally graded plates using isogeometric analysis. *Engineering Computations*, 32(2), 2015.

[Rauen *et al.* 2017] M. Rauen, R. D. Machado, M. Arndt. Isogeometric analysis of free vibration of framed structures: comparative problems *Engineering Computations*, 34(2), 2017.

[Hartmann *et al.* 2011] S. Hartmann, D.J. Benson, and D. Lorenz. *About Isogeometric Analysis and the new NURBS-based Finite Elements in LS-DYNA*, In: 8th European LS-DYNA Users Conference, Strasbourg, France, 2011.

[Benson *et al.* 2013] D. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, T.J.R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133–146, 2013.

[Hartmann *et al.* 2016] S. Hartmann, D. Benson, A. Nagy. Isogeometric analysis with LS-DYNA. *J. Phys. Conf. Ser.* 734(3):032125, 2016.

[Chen *et al.* 2016] Y. Chen, S.P. Lin, O. Faruque, J. Alanoly, M. El-Essawi, and R. Baskaran. *Current status of ls-dyna isogeometric analysis in crash simulation*, In: 14th International LS-DYNA Users Conference, Detroit, 2016.

[Al-Akhras *et al.* 2016] H. Al-Akhras, T. Elguedj, A. Gravouil and M. Rochette. Isogeometric analysis-suitable trivariate NURBS models from standard B-Rep models, *Computer Methods in Applied Mechanics and Engineering*, 307:256–274, 2016.

[Elguedj *et al.* 2012] T. Elguedj, A. Duval, F. Maurin, H. Al-Akhras. *Abaqus user element implementation of NURBS based isogeometric analysis*, In: 6th European Congress on Computational Methods in Applied Sciences and Engineering, Vienna, Austria, 2012, pp. 10–14.

[Duval *et al.* 2015] A. Duval, T. Elguedj, H. Al-Akhras, and F. Maurin. *abqNURBS: implémentation d'éléments isogéométriques dans Abaqus et outils de pré - et post - traitement dédiés*, In: 12e Colloque national en calcul des structures, Giens, 2015.

[Lai *et al.* 2017] 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.

[Occelli *et al.* 2019] M. Occelli, T. Elguedj, S. Bouabdallah, L. Morancay. LR B-Splines implementation in the Altair RadiossTM solver for explicit dynamics IsoGeometric Analysis. *Advances in Engineering Software*, 131:166-185, 2019.

[Borden *et al.* 2011] 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.

[Scott *et al.* 2011] M.A. Scott, M. Borden, C. Verhoosel, T. Sederberg and T.J.R Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88:126–156, 2011.

[Scott *et al.* 2012] M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213–216:206–222, 2012.

[Schillinger *et al.* 2012] D. Schillinger, L. Dedé, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes. An isogeometric design-through- analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249–250:116–150, 2012.

[Hennig *et al.* 2016] P. Hennig, S. Müller, M. Kästner. Bézier extraction and adaptive refinement of truncated hierarchical NURBS. *Computer Methods in Applied Mechanics and Engineering* 305:316-339, 2016.

[Angella *et al.* 2018] D. D'Angella, S. Kollmannsberger, E. Rank, A. Reali. Multi-level Bézier extraction for hierarchical local refinement of Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 328:147-174, 2018.

[Evans *et al.* 2015] E.J. Evans, M.A. Scott, X. Li, D.C Thomas. Hierarchical T-splines: analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:1–20, 2015.

[Chen and de Borst 2018] L. Chen, R. de Borst Adaptive refinement of hierarchical T-splines. *Computer Methods in Applied Mechanics and Engineering*, 337:220–245, 2018.

[Dokken *et al.* 2013] T. Dokken, T. Lyche, K.F. Pettersen. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design*, 30:331–356, 2013.

[Schillinger *et al.* 2016] 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.

[Nguyen and Schillinger 2017] L.H. Nguyen and D. Schillinger. A collocated isogeometric finite element method based on Gauss–Lobatto Lagrange extraction of splines. *Computer Methods in Applied Mechanics and Engineering* 316:720–740, 2017.

[Dalcin *et al.* 2016] L. Dalcin, N. Collier, P. Vignal, A.M.A. Cortes and V.M. Calo. PetIGA: A framework for high-performance isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 308:151-181, 2016.

[Chang *et al.* 2016] F. Chang, W. Wang, Y. Liu and Y. Qu. A Fortran implementation of isogeometric analysis for thin plate problems with the penalty method. *Engineering Computations*, 33(7), 2016.

[Nguyen *et al.* 2015] V.P. Nguyen, C. Anitescu, S.P.A. Bordas and T. Rabczuk. Isogeometric analysis: an overview and computer implementation aspects. *Mathematics and Computers in Simulation*, 117:89–116, 2015.

[Calabro *et al.* 2017] F. Calabro, G. Sangalli and M. Tani. Fast formation of isogeometric Galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering*, 316:606-622, 2017.

[Gendre *et al.* 2009] 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.

[Bouclier *et al.* 2016] R Bouclier, J.C. Passieux and M Salaün. Local enrichment of NURBS patches using a non-intrusive coupling strategy: geometric details, local refinement, inclusion, fracture. *Computer Methods in Applied Mechanics and Engineering*, 300:1–26, 2016.

[Bouclier *et al.* 2017] 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.

[Duval *et al.* 2016] 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.

[Oumaziz *et al.* 2017] P. Oumaziz, P. Gosselet, P.A. Boucard and S. Guinard. A non-invasive implementation of a mixed domain decomposition method for frictional contact problems. *Computational Mechanics*, 60(5):797-812, 2017.

[Oumaziz *et al.* 2019] P. Oumaziz, P. Gosselet, P.A. Boucard and S. Guinard. A parallel non-invasive mixed domain decomposition - Implementation and applications to mechanical assemblies. *Finite Elements in Analysis and Design, 156:24-33, 2019.*

[Bettinotti et al. 2014] O. Bettinotti, O. Allix, U. Perego, V. Oancea and B. Malherbe, A fast weakly intrusive multi-scale method in explicit dynamics. International Journal for Numerical Methods in Engineering, 100(8):577-595, 2014.

[Chevreuil et al. 2013] M. Chevreuil, A. Nouy and E. Safatly. A multiscale method with patch for the solution of stochastic partial differential equations with localized uncertainties. Computer Methods in Applied Mechanics and Engineering, *255:255-274, 2013.*

[Guinard et al. 2018] 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).

[Kamensky and Bazilevs 2019] D. Kamensky and Y. Bazilevs. tIGAr: Automating isogeometric analysis with FEniCS. Computer Methods in Applied Mechanics and Engineering, *344:477-498, 2019.*

[Code_ Aster 2014] Électricité de France, Code_ Aster, http://www.code-aster.org, *2014.*

[Cottrell et al. 2007] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. Computer Methods in Applied Mechanics and Engineering, *196:4160–4183, 2007.*

[Bézier 1986] P. Bézier. In Courbes & Surfaces. *Hermès, 1986.*

[Thomas et al. 2014] D.C. Thomas, M.A. Scott, J.A. Evans, K. Tew, E.J. Evans. Bézier projection: a unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis. Computer Methods in Applied Mechanics and Engineering, *284:55–105, 2014.*

[Govindjee et al. 2012] S. Govindjee, J. Strain, T.J. Mitchell, R.L. Taylor. Convergence of an efficient local least squares fitting method for bases with compact support. Computer Methods in Applied Mechanics and Engineering, *213-216:84-92, 2012.*

[Zienkiewicz et al. 2003] O. Zienkiewicz, R. Taylor, and J. Zhu. In The Finite Element Method: Its Basis and Fundamentals, Sixth Edition, *Butterworth-Heinemann, 2005. ISBN: 978-0-7506-6431-8.*

[Cottrell et al. 2006] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. Computer Methods in Applied Mechanics and Engineering, *195(41–43):5257–5296, 2006.*

[Schillinger et al. 2013] D. Schillinger, J.A. Evans, A. Reali. M.A. Scott, T.J.R.Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. Computer Methods in Applied Mechanics and Engineering, *267:170–232, 2013.*

[Belytschko et al. 1985] T. Belytschko, H. Stolarski , W.K. Liu , N. Carpenter and J.S.J Ong. Stress projection for membrane ans shear locking in shell finite elements. Computer Methods in Applied Mechanics and Engineering, *51:221–258, 1985.*

[MacNeal et al. 1985] R.H. MacNeal and R.L. Harder. A proposed standard set of problems to test finite element accuracy. Finite Elements in Analysis and Design, *1:3-20, 1985.*

[Mathieu et al. 2015] F. Mathieu, H. Leclerc, F. Hild ans S. Roux. Estimation of Elastoplastic Parameters via Weighted FEMU and Integrated-DIC. Experimental Mechanics, *55:105–119, 2015.*

[Charras et al. 1993] T. Charras, A. Millard, and P. Verpeaux. Solution of two-dimensional and three dimensional contact problems by means of Lagrange multipliers in the CASTEM 2000 finite element program. in Proc. Contact Mechanics, Computer Mechanics Publication*, 183–194, 1993.*

[Kerfriden et al. 2012] P. Kerfriden, J.C. Passieux, S.P.A. Bordas. Local/global model order reduction strategy for the simulation of quasi-brittle fracture. International Journal for Numerical Methods in Engineering*, 89(2):154–179, 2012.*

[Quarteroni et al. 2016] A. Quarteroni, A. Manzoni, F. Negri. Reduced basis methods for partial differential equations - An introduction. Springer*, 2016.*

[Chinesta et al. 2017] F. Chinesta, A. Huerta, G. Rozza, K. Wilcox. Model reduction methods - Part 1: solids and structures. Encyclopedia of Computational Mechanics Second Edition *2017.*