LR B-Splines implementation in the Altair RadiossTM solver for explicit dynamics IsoGeometric Analysis

M. Occelli, Thomas Elguedj, S. Bouabdallah, L. Morançay

Version of Record: https://www.sciencedirect.com/science/article/pii/S0965997818311189
Manuscript_b89ec2814a2ee66243f6051055c37f00

LR B-Splines implementation in the Altair Radioss^{TM} solver for explicit dynamics IsoGeometric Analysis

M. Occelli[a,b,*], T. Elguedj[a], S. Bouabdallah[b], L. Morançay[b]

[a] *Univ Lyon, INSA-Lyon, CNRS UMR5259, LaMCoS, F-69621, France*
[b] *Altair Engineering, WTC bât.M, 1300 Route des Crêtes, 06560 Sophia Antipolis, France*

Abstract

IsoGeometric Analysis has shown to be a very promising tool for an integrated design and analysis process. A challenging task is still to move IGA from a proof of concept to a convenient design tool for industry and this work contributes to this endeavor. This paper deals with the implementation of the IGA concept into Altair Radioss finite element solver in order to address crash and stamping simulation applications. To this end, the necessary ingredients to a smooth integration of IGA in a traditional finite element code have been identified and adapted to the existing code architecture. A solid B-Spline element has been developed in Altair Radioss and then, an existing contact interface has been extended in order to work seamlessly with both NURBS and Lagrange finite elements. As local refinement is needed for solution approximation, an analysis is made in terms of analysis suitability and implementation aspects for several Spline technologies. The Locally Refined B-Spline (LRBS) technology is implemented and is validated on industrial benchmarks, for validation cases conventionally used for industrial codes like stamping and drop test.

Keywords: Industrial solver, IsoGeometric Analysis, Volumetric approach, Local refinement, Explicit dynamic solution, Analysis-suitability

1. Introduction

Since its introduction by Hughes et al. [1] and Cottrell et al. [2], IsoGeometric Analysis demonstrates a high potential gain with respect to efficiency, quality and accuracy in analysis by replacing traditional finite elements by B-Spline and Non-Uniform Rational B-Spline (NURBS) elements, see, e.g., [3–8]. NURBS functions were initially chosen as a basis, due to their relative simplicity and their use in the graphics industry: Computer Aided Design (CAD) and Animation. However, from the analysis standpoint, the tensor product structure of NURBS can be seen as a drawback as it prevents local refinement that is commonly used with classical FEA.

Many different Spline technologies have addressed the problem of local refinement, in particular Hierarchical B-Splines [9–12], Truncated Hierarchical B-Splines [13–15], T-Splines [16–19] and Locally Refined B-Splines [20–22]. They were introduced and subsequently improved in the past decade focusing on stability [23], linear independence [24–26] and partition of unity [13] properties. Research work is still ongoing for most of these Spline technologies to bring a technology with the desired analysis properties and the ability to locally refine meshes.

In the last years, IGA software were introduced in the research domain, in Matlab with GeoPDEs [27], in OOFEM [28], in C++ with Igatools [29], in C with PetIGA [30], or in Java [31], trying to constitute

*Corresponding author
Email addresses: matthieu.occelli@insa-lyon.fr,moccelli@altair.com (M. Occelli), thomas.elguedj@insa-lyon.fr (T. Elguedj), sbouabdallah@altair.com (S. Bouabdallah), lionel@altair.com (L. Morançay)

teaching and prototyping tools. Several IGA implementations in industrial software were also done, e.g. in LS-Dyna [32, 33] and in Abaqus [34, 35]. In LS-Dyna, the first IGA implementation was done for shells, providing good results and accuracy, benefiting in the same time of less complexity compared to a solid formulation. Attempts to implement solid formulation have more recently been done [36].

1.1. Outline of the article

This paper presents a locally refined solid IGA formulation using LR B-Splines for transient simulations and its implementation in Altair Radioss explicit dynamics solver, a leading structural analysis solver for nonlinear problems under transient loadings. The paper is organized as follows. The theoretical basis of B-Splines and NURBS will be quickly laid, with the initial isogeometric technologies. Then, several Spline technologies which allow to obtain a locally refined mesh will be introduced, highlighting similarities and differences between them. The refinement strategies using LR B-Splines will be more precisely discussed and an improved refinement scheme will be given to ensure the analysis-suitability for the trivariate cases in the Altair Radioss explicit solver. Some modifications of existing procedures inside Altair Radioss were done in order to fully integrate IGA inside the solver, with the formerly existing elements. Finally, we will present isogeometric analysis capabilities showing numerical results on benchmark examples.

2. B-Splines, NURBS and IGA review

We briefly review the IGA concept based on B-Splines and NURBS presented by Hughes et al. [1], Cottrell et al. [2].

B-Splines are piecewise polynomial functions with a prescribed degree of continuity. Univariate B-Spline basis functions are constructed from a knot vector, a set of coordinates in parametric space, $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the i^{th} knot and $\xi_{i+1} \geq \xi_i \ \forall i$. i is the knot index, p is the polynomial order, and n is the number of basis functions. B-Spline basis functions for a given order p are defined recursively in the parametric space, given a knot vector Ξ, beginning with piecewise constants ($p = 0$):

$$N_{i,0}(\xi) = \begin{cases} 1 \text{ if } \xi_i \leq \xi \leq \xi_{i+1}, \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

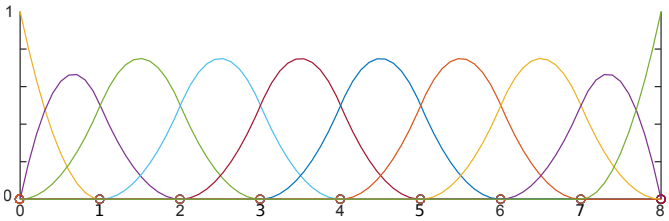
For $p \geq 1$, the basis is defined by the Cox-de Boor recursion formula [37]:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{2}$$

A knot vector of size $n + p + 1$ will generate n linearly independent basis functions of degree p. Each of them is non-zero on a span of $p + 2$ knots, which defines a local knot interval. A given $\Xi = \{0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 8, 8\}$ knot vector will define 10 quadratic B-Spline functions with 10 local knots intervals, see Figure 1. In these knot vectors, none of the intermediate knots is repeated, except for the edge functions of the patch. The multiplicity of the external knot is then maximal and the functions are interpolatory, i.e., have C^0 continuity.

B-Spline curves can be built in $\mathbb{R}^d$ by linear combination of basis functions. Coefficients $\boldsymbol{B}_i \in \mathbb{R}^{\mathrm{d}}, i = 1, 2, \ldots, n$ are called control points. They are similar to nodes for finite element analysis and define the control polygon. Given n functions $N_{i,p}, i = 1, 2, \ldots, n$, and the corresponding control points $\boldsymbol{B}_i \in \mathbb{R}^{\mathrm{d}}, i = 1, 2, \ldots, n$, a B-Spline curve is defined by:

$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) \boldsymbol{B}_i. \tag{3}$$



$$
\begin{aligned}
\Xi &= \{0,0,0,1,2,3,4,5,6,7,8,8,8\} \\
\Xi_1 &= \{0,0,0,1 \qquad\qquad\qquad\qquad\qquad\quad\}, \\
\Xi_2 &= \{\ \ 0,0,1,2 \qquad\qquad\qquad\qquad\quad\}, \\
\Xi_3 &= \{\ \ \ \ 0,1,2,3 \qquad\qquad\qquad\quad\}, \\
\Xi_4 &= \{\ \ \ \ \ \ 1,2,3,4 \qquad\qquad\qquad\}, \\
\Xi_5 &= \{\ \ \ \ \ \ \ \ 2,3,4,5 \qquad\qquad\}, \\
\Xi_6 &= \{\ \ \ \ \ \ \ \ \ \ 3,4,5,6 \qquad\quad\}, \\
\Xi_7 &= \{\ \ \ \ \ \ \ \ \ \ \ \ 4,5,6,7 \qquad\}, \\
\Xi_8 &= \{\ \ \ \ \ \ \ \ \ \ \ \ \ \ 5,6,7,8 \qquad\}, \\
\Xi_9 &= \{\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 6,7,8,8 \quad\}, \\
\Xi_{10} &= \{\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ 7,8,8,8\}.
\end{aligned}
$$

Fig. 1: All quadratic basis functions generated by the knot $\Xi = \{0,0,0,1,2,3,4,5,6,7,8,8,8\}$. Each individual basis function can be described using a local knot vector of $p+2$ knots each.

We can consider local mappings defined over one individual knot span, i.e., $[\xi_i, \xi_{i+1}]$ which is covered by $p+1$ B-Spline basis functions. This knot interval can be interpreted as a finite element, and n^e_{cp} can be introduced as the number of non zero local basis functions on the knot interval, where $n^e_{cp} = p+1$ in 1D. Equation (3) can therefore be rewritten as:

$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n^e_{cp}} N_{i,p}(\xi)\boldsymbol{B}_i. \tag{4}$$

The global tensor product generalization of univariate B-Splines defines multivariate B-Splines and allows to build B-Spline surfaces and volumes in two or three parametric directions, with p, q and r as the polynomial degree for each direction:

$$\boldsymbol{V}(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{o} N_{i,p}(\xi)M_{j,q}(\eta)R_{k,r}(\zeta)\boldsymbol{B}_{ijk}. \tag{5}$$

The knot spans for 2D and 3D cases, i.e. $[\xi_i,\xi_{i+1}]\times[\eta_j,\eta_{j+1}]$ and $[\xi_i,\xi_{i+1}]\times[\eta_j,\eta_{j+1}]\times[\zeta_k,\zeta_{k+1}]$, define in a similar way 2D and 3D elements. n^e_{cp} is then respectively equal to $(p+1)(q+1)$ and $(p+1)(q+1)(r+1)$, and Equation (5) can be re-written:

$$\boldsymbol{V}(\xi,\eta,\zeta) = \sum_{i=1}^{n^e_{cp}} N_i(\xi,\eta,\zeta)\boldsymbol{B}_i \quad \text{with} \quad N_i(\xi,\eta,\zeta) = N_{i,p}(\xi)M_{j,q}(\eta)R_{k,r}(\zeta). \tag{6}$$

2.1. Non-uniform rational B-Splines

NURBS basis functions can be obtained by weighting the B-Spline functions. More specifically, NURBS functions in $\mathbb{R}^d$ are built from projective transformations of B-Splines in $\mathbb{R}^{d+1}$. They are defined as:

$$R^p_i(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^{n} N_{i,p}(\xi)w_i}, \tag{7}$$

where $N_{i,p}(\xi)$ are polynomial B-Spline basis functions and w_i are weights.

Analogously, NURBS curves are built from linear combination of NURBS functions and the global tensor product is used to obtain NURBS surfaces and volumes:

$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n} R_{i,p}(\xi)\boldsymbol{B}_i, \tag{8}$$

$$\boldsymbol{V}(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{o} N_{i,p}(\xi)M_{j,q}(\eta)R_{k,r}(\zeta)\boldsymbol{B}_{ijk}. \tag{9}$$

2.2. h-, p- and k-refinement

B-Splines can be refined by three different approches : h-, p- and k- refinement. All these methods preserve initial geometry. h-refinement consists of a knot insertion in the global knot vector. It is equivalent to mesh refinement for FE. p-refinement is order elevation and raises the degree of the basis functions preserving the continuity. k-refinement is a combination of the previous two methods. It increases the degree and the regularity of the basis functions. Details of these refinements are given by Hughes et al. [1], Cottrell et al. [2]. Because of the global tensor product, it is not possible to obtain local refinement in 2D or 3D, as the tensor product structure propagates it along all the other parametric directions. Other methods or B-Spline technologies are then needed to preserve the locality of the refinement.

3. Local refinement approaches

In the following, we briefly review some Spline technologies developed to set up local refinement in IGA. For illustration purpose, we consider in 1D a coarse B-Spline basis functions with the global knot vector $\Xi = [0,0,0,1,2,3,4,5,6,7,8,8,8]$. We suppose that the area to be refined is set from knot $\xi = 4$ to $\xi = 7$, see Figure 2.

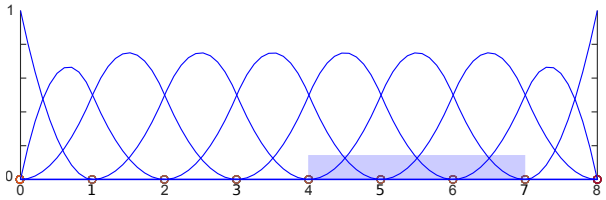


Fig. 2: Quadratic B-Spline basis functions, defined from $\Xi = [0,0,0,1,2,3,4,5,6,7,8,8,8]$ as the coarse level of functions. The area to be refined is highlighted in blue, between knots $\xi = 4$ and $\xi = 7$.

3.1. Hierarchical B-Splines

Hierarchical B-Splines were introduced and used for IGA by Vuong et al. [9], Giannelli and Jüttler [10] and by Schillinger et al. [11, 12]. A finer basis is generated by subdividing the coarse knot vector, adding each intermediate knot values, see Figure 3. The refinement consists in replacing the coarse basis function which is entirely included in the span to be refined by all non-zero finer functions on this span. The subdivision of a coarse function will generate $p+2$ new finer functions on the refined knot interval. In Figure 3, the coarse function is represented by a dotted blue line and the finer functions by red lines, for the corresponding refined area.

The refinement procedure starts from the coarser level and generates one or several nested finer levels, depending on the needed refinement level. The substitution is done without modifying the nearest neighbour coarse functions. It leaves some overloaded elements. The overloading of element or knot interval is defined

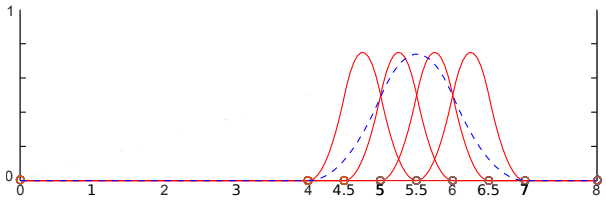


Fig. 3: Quadratic B-Spline basis function defined with a knot vector composed with the same knot than Ξ plus the intermediate knot values. Non-zero finer functions on the area to be refined are colored in dark red.

as an overlap of more than $p+1$ basis functions on an element or knot interval in 1D [20]. For Hierarchical B-Splines, some elements located on the boundary of the refinement area are overloaded, with more than $p+1$ non-zero functions on their interval. These elements are colored in light red in Figure 4, where the basis functions are colored in blue for the coarse one and in red for the finer one. The black ones are coarse basis functions which are outside the refinement area. We can also note that partition of unity is lost with Hierarchical B-Splines. Normalizing the basis, as introduced by Temizer and Hesch [38], allows to recover this property.

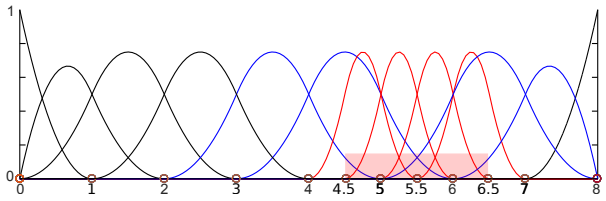


Fig. 4: Quadratic Hierarchical B-Spline basis function. Coarse B-Spline functions are colored in black and blue and finer ones in red. The overlap of more than $p+1$ functions from knots $\xi = 4.5$ to $\xi = 6.5$ is highlighted in red.

3.2. Truncated Hierarchical B-Splines

Truncated Hierarchical B-Splines were introduced by Giannelli et al. [13], Kiss et al. [14], Giannelli et al. [15]. Truncating the coarse basis functions reduces their support over refined area by applying a scaling factor called truncature on the coarse functions that are in the neighbourhood of the refined area. Partition of unity is in the same time restored and the number of overloaded knot intervals decreases significantly. However some overloaded intervals still exist, caused by the too large extent of the intermediate coarse basis functions on each side of the refinement area. These are represented by dotted lines in Figure 5.

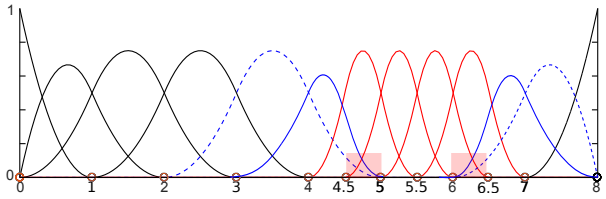


Fig. 5: Quadratic Truncated Hierarchical B-Spline basis functions. Coarse B-Spline functions are colored in black and blue and finer ones in red. The overlap area was reduced to a knot intervals from $\xi = 4.5$ to $\xi = 5$ and from $\xi = 6$ to $\xi = 6.5$: The dotted coarse functions are still too extended.

Another Spline technology named Polynomial Splines over Hierarchical T-meshes (PHT-Splines) was also introduced by Deng et al. [39], Nguyen-Thanh et al. [40], Wang et al. [41]. It shares some similarities with THB-Splines. For conciseness, they will not be detailed here, the interested reader can refer to the previously cited references.

3.3. Locally-Refined B-Splines

LR B-Splines were introduced by Dokken [42] and applied to IGA by Johannessen et al. [21, 22]. This refinement method does not use a subdivision procedure like (T)HB-Splines but a local knot insertion, inserting one knot at a time and splitting old B-Spline basis functions into two new ones, generating few new functions, see Figure 6. LR B-Spline basis functions are then defined on local knot vectors. Each refinement using the local knot insertion creates or modifies two functions. The algorithm is quite similar to the classical knot insertion in a standard tensor-product B-Spline. Considering a new knot $\hat{\xi}$ to be inserted between the knots ξ_{i-1} and ξ_i in a knot vector Ξ for a 1D case, Ξ will be split in two knot vectors Ξ_1 and Ξ_2, with an equivalent length, depending on the B-Spline degree:

$$\begin{aligned} \Xi &= [\xi_1, \xi_2, ..., \xi_{i-1}, \quad \xi_i, ..., \xi_{p+1}, \xi_{p+2}], \\ \Xi_1 &= [\xi_1, \xi_2, ..., \xi_{i-1}, \hat{\xi}, \xi_i, ..., \xi_{p+1} \quad\quad], \\ \Xi_2 &= [\quad \xi_2, ..., \xi_{i-1}, \hat{\xi}, \xi_i, ..., \xi_{p+1}, \xi_{p+2}]. \end{aligned} \tag{10}$$

The enriched B-Spline function B_Ξ is then a linear combination of the two B-Spline functions B_{Ξ_1} and B_{Ξ_2}:

$$B_\Xi(\xi) = \alpha_1 B_{\Xi_1}(\xi) + \alpha_2 B_{\Xi_2}(\xi), \tag{11}$$

where

$$\begin{aligned} \alpha_1 &= \begin{cases} \frac{\hat{\xi}-\xi_1}{\xi_{p+1}-\xi_1} & \text{if} \quad \xi_1 \leq \hat{\xi} \leq \xi_{p+1}, \\ 1 & \text{if} \quad \xi_{p+1} \leq \hat{\xi} \leq \xi_{p+2}, \end{cases} \\ \alpha_2 &= \begin{cases} 1 & \text{if} \quad \xi_1 \leq \hat{\xi} \leq \xi_2, \\ \frac{\xi_{p+2}-\hat{\xi}}{\xi_{p+2}-\xi_2} & \text{if} \quad \xi_2 \leq \hat{\xi} \leq \xi_{p+2}. \end{cases} \end{aligned} \tag{12}$$

The weights α_1 and α_2 are necessary to ensure the partition of unity. Several local knot insertions give the enriched Spline space, see Figure 6.

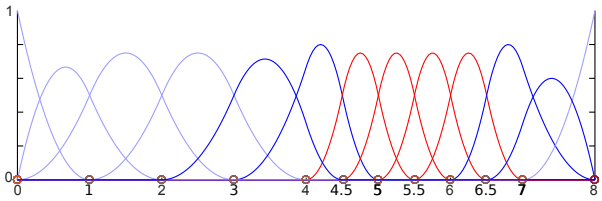


Fig. 6: Locally Refined B-Spline basis functions. Notice that there are no more than $p+1$ non-zero functions on each knot interval.

3.4. T-Splines

T-Splines were introduced by Churchhouse [16], Pottmann [17], Sederberg et al. [18] and used for IGA by Dörfel et al. [19] and Bazilevs et al. [43]. Like LR B-Splines, one of the main features lies on the fact that T-Spline basis functions are defined on local knot vectors. T-Splines allow to build meshes where the global tensor product is not a feature and where local refinement goes through T-junctions, analogous to hanging nodes for classical FE. Several T-Spline technologies are avalable including T-Splines [18, 44], analysis-suitable T-Splines [25], modified T-Splines [45], weighted T-Splines [46, 47], hierarchical analysis-suitable T-Splines [48], or truncated T-Splines [49]. Each approach corrects the shortcomings of the previous one, like the lack of linear independence or partition of unity.

3.5. Summary of the analysis for implementation in a commercial solver

The required properties of Spline basis functions to implement IGA in a FE architecture can be summarized as follows:

- Partition of unity

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1, \qquad \forall \xi \in [0,1]\,. \tag{13}$$

- Linear independence

$$\sum_{i=1}^{n} c_i N_{i,p}(\xi) = 0 \Leftrightarrow c_i = 0, i = 1, ..., n, \qquad \forall \xi \in [0,1]\,. \tag{14}$$

- Compact support, i.e., the support of $N_{i,p}$ is included in $[\xi_i, \xi_{i+p+1}]$.
- No overloaded element, i.e. no more than $p+1$, $(p+1)(q+1)$ or $(p+1)(q+1)(r+1)$ basis functions on an element (or knot interval) respectively for a 1D, 2D or 3D basis.

We will consider in this work that a sufficient condition to have a linear independent basis is to have no overloading, see Bressan [26], Bressan and Jüttler [50]. The non-overloading is an important aspect in terms of the locally refined IGA implementation in a traditionnal FE code like Radioss.

Hierarchical B-Splines are linearly independent by construction but do not ensure the partition of unity. The overloading of constructed elements is a specificity which makes complex the dimensioning of a classical commercial FE solver data structures. Truncated Hierarchical B-Splines are also linearly independent by construction and reduce this overlapping. However, they do not remove it entirely. Moreover, this method is somewhat costly because the truncation has to be done at any point of the parametric space. Finally, for these two approaches, the link between parent levels must be maintained throughout the calculation and makes their implementation complex.

T-Splines were often presented as a solution to local refinement. Their implementation has been done in LS-Dyna for shells and showed interesting results [36]. To the best of our knowledge, there has been no native implementation in a commercial solver and the full generalization to 3D, i.e., with T-junctions in all directions, is unclear.

LR B-Splines have several advantages, in particular there is no need of a link between coarse and fine levels after the construction of the refinement. After this construction, the functions are defined with their local basis knot vector independently of the other functions. The partition of unity is guaranteed and there is no overloaded element in 1D by construction. We will see later that for 2D and 3D cases, methods exist to ensure this property. Linear independence is guaranteed when there is no overloading [26].

4. LR B-Splines in 3D and overloaded elements

LR B-Spline refinement is constructed in 1D by inserting one singular knot or a mesh line in 2D. Mesh lines are defined by a knot value in one of the parametric direction and by an extent in the knot vector in the other parametric direction. In 3D, refinements are constructed using what we call mesh surfaces. Mesh surfaces are defined analogously with the specific inserted knot value in one of the three parametric directions which define the normal direction, and by a rectangular extent defined by two knot intervals in other parametric directions. All basis functions that are completely cut by a mesh line or a mesh surface in

2D and 3D will be split, creating or modifying at least two basis functions.

Figure 7 illustrates a hexahedral mesh with a set of quadratic B-Spline basis functions. The black dots are the control points associated with some specific B-Spline basis functions. Their global extent is colored in light pink, and is also given on each projected plane (ξ, η), (η, ζ) and (ζ, ξ). The blue and red lines define the mesh surface contour in the (η, ζ) plane at the ξ specific knot value. The light pink basis functions will be split, giving the interior element refinement in the ξ direction.

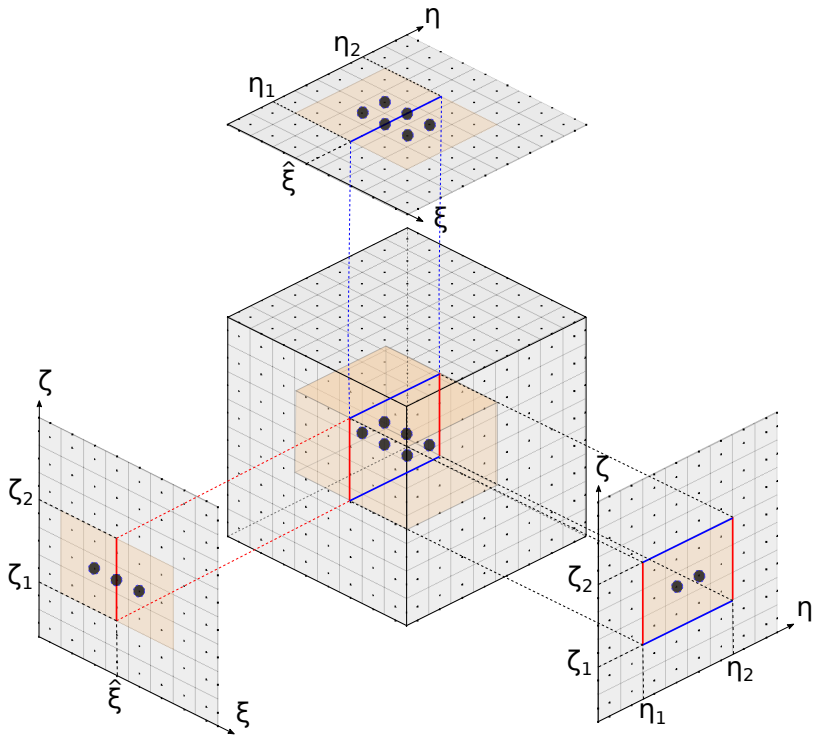


Fig. 7: 3D mesh surface: a specific knot value $\hat{\xi}$ defines the normal direction and the $[\eta_1, \eta_2] \times [\zeta_1, \zeta_2]$ intervals give the rectangular extent on the other directions. Black colored B-Spline basis functions are completely traversed by the mesh surface and will be split. Their support is highlighted in light pink. Mesh surface and basis functions support are projected on (ξ, η), (η, ζ) and (ζ, ξ) planes for a better visualization.

One of the contribution of this work is to set up a 3D formulation of LR B-Splines. However, to better explain the key points and issues of refinement for more complex cases than Figure 7, in particular non rectangular domains, the following points will be described on 2D cases. The refinements will be shown in the (ξ, η) plane. Next recommandations will have to be applied on each plane for each corresponding 3D refinement.

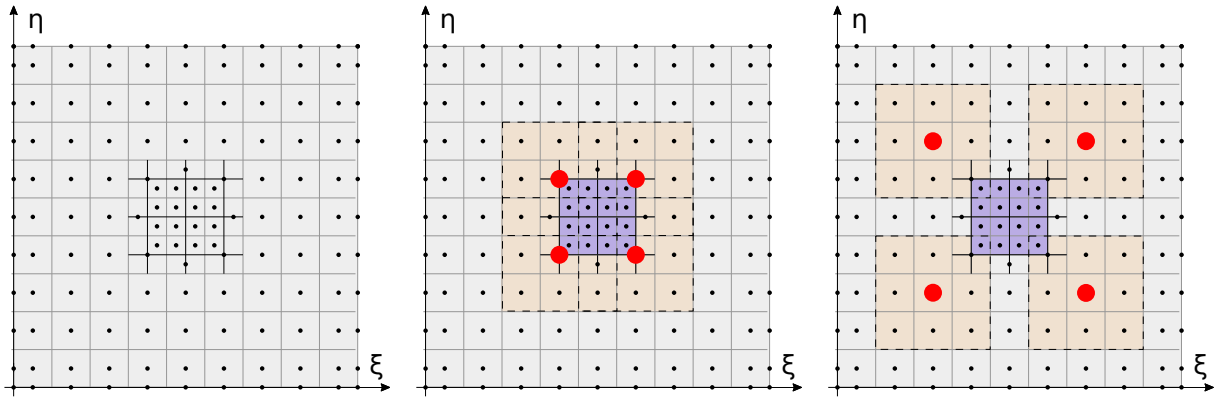
4.1. Overloaded elements in 3D

Due to the shift of the ends of local knot, LR B-Spline refinement does not generate overloaded elements in 1D. However, it is possible to have overloaded elements in multiple dimensions and the obtained basis functions are not guaranteed to be linearly independent and to be suited for analysis. Johannessen et al. [21] deal with linear independence recovery methods, based on overloading elimination, and construction of LR B-Spline basis according to several diagrams for 2D cases of rectangular zone. The overloading elimination

is an a posteriori method which will deactivate several basis functions on an overloaded element to recover the linear independence. This method will not be detailed. We focus on existing a priori methods to obtain refinements with non overloaded elements. Two of these methods will be discussed in the following.

4.1.1. Structured Mesh

Refinement according to the structured mesh method [21], see Figure 8(a), is described as a bi-directional refinement of all B-Spline functions fully contained in the rectangular refinement area. This scheme generates overloading which is due to the functions that are in the diagonals of the refined area and are illustrated in Figure 8(b) and Figure 8(c). The overloaded elements are colored in blue and the support for two of the extended functions is colored in orange in Figure 8. The extents of these functions were not modified and are too large. This refinement scheme in 2D and its generalization to 3D thus leads to a high number of overloading.



(a) (b) (c)

Fig. 8: Refinement according to structured mesh scheme for a quadratic 2D basis. Some elements, colored in blue, are overlodaded. There is still too many B-Spline functions on the elements. Some of them are dotted in red and their extent are colored in orange.

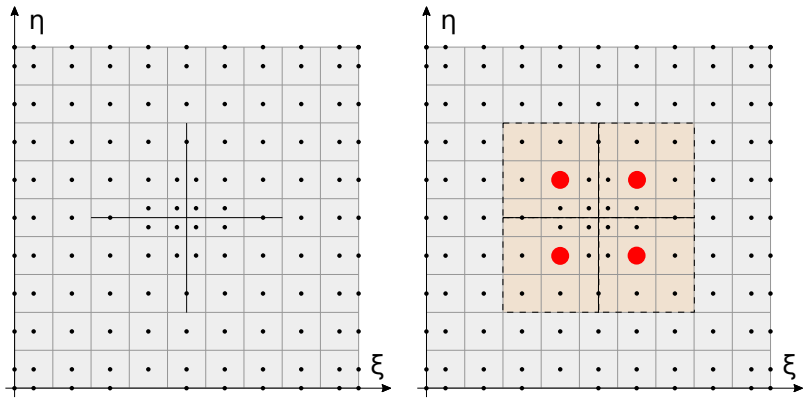
4.1.2. Full Span

The full span refinement [21], consists in extending the refinement lines, see Figure 9(a). Taking one or several elements to be refined, all the functions that have these elements on their support have to be split. These functions therefore no longer cause overloading, see Figure 9(b). The full span scheme ensure the linear independence for refinements that are rectangular shaped in 2D or hexahedral shaped in 3D, with only outward corners.

The full span scheme is sufficient for linear and cubic polynomial degree [21]. This is also valid for quadratic 3D cases that have outgoing corners. The example of Figure 10 is an illustration of the use of a full span scheme to establish a gradual refinement of a corner, for a quadratic 3D mesh.

4.1.3. Improved full span scheme

Despite this indication, there are still cases where overloading can be oberserved in 3D cases for refinements more complex than a rectangle in 2D or a hexahedron in 3D. This is for example the case for the quadratic case of Figure 11. This figure illustrates the refinement of the left and the lower part of a 2D mesh, which thus forms an L shape. In this case, the corner of the L is not an outgoing corner but a re-entrant



(a) (b)

Fig. 9: Full span refinement scheme for a quadratic 2D basis: extension of refinement lines on p elements. There are no more overloaded elements.

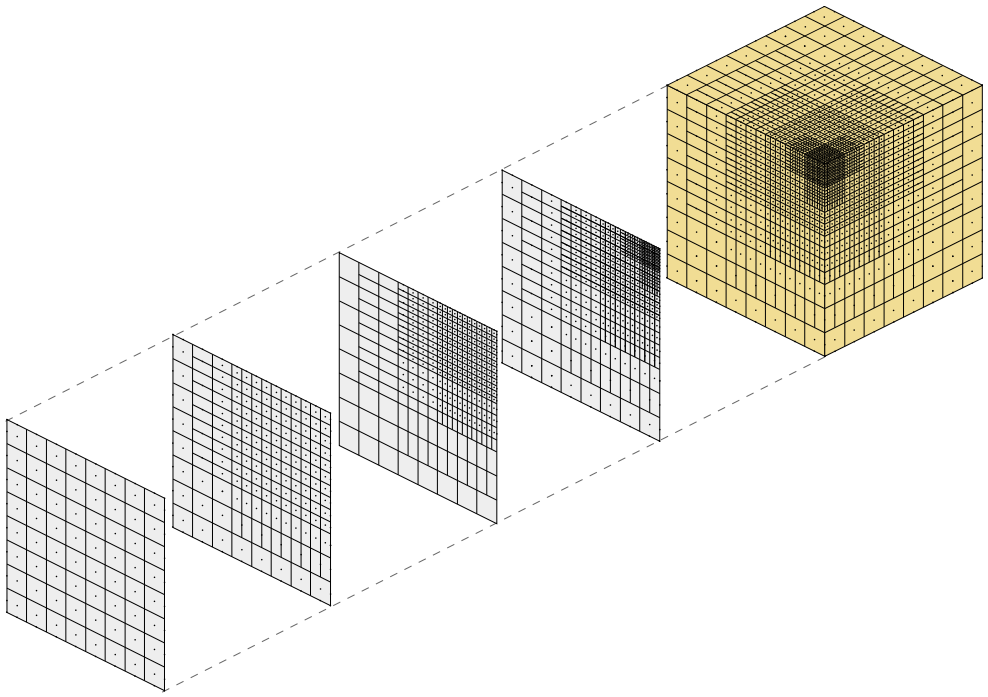


Fig. 10: Refinement applied following the pattern of full span on all three sides and inside a cube. There are no overloaded elements. The gray planes are the different sections illustrating the different levels of refinement in one of the three directions.

corner. The full span scheme does not allow here to avoid the presence of overloaded elements in this corner, as can be seen on Figure 11. The set of overloaded elements is highlighted in blue. This overloading is caused by the too large extent of the quadratic function that is highlighted.

To obtain a set of non overloaded elements, the p elements by direction in the corner have to be refined in only one direction. It is therefore necessary, as one can see in Figure 12, to have a refined model with a horizontal or vertical refinement in the re-entrant corner. In this way, one comes to reduce the number of re-

fined elements in the re-entrant corner but it is no longer overloaded. There are several working refinements that can be obtained with this procedure, and the linear independence of the basis is ensured. This case, presented here for a 2D patch, can be found very often in 3D cases for cross-shaped refinements, L-shaped refinement, stairs or any shape that has re-entrant corners.

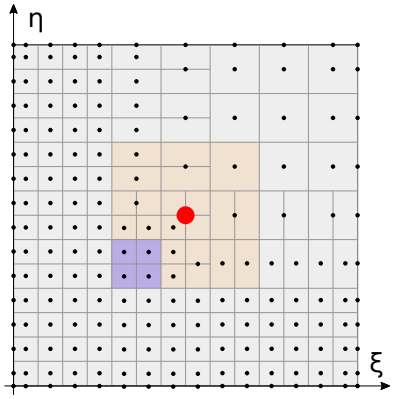
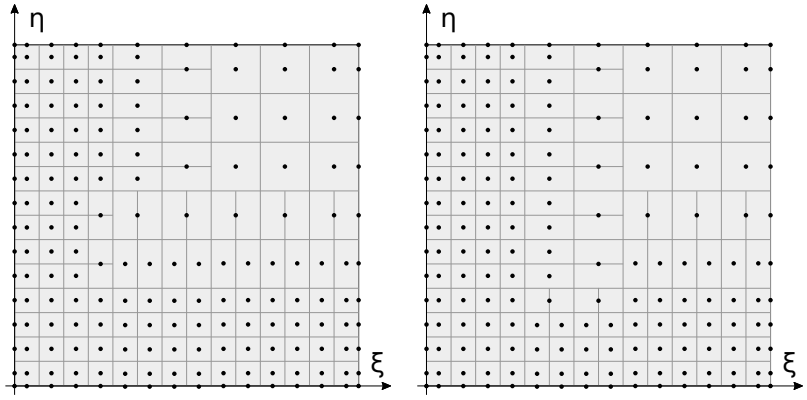


Fig. 11: Refinement of a re-entrant corner of a function base losing linear independence property. Overloading blue elements caused by the function that is highlighted.



(a) (b)

Fig. 12: Modification of the refinement of p elements by direction of the re-entrant corner, see Figure 11, to keep a non overloaded mesh. These elements are now refined in one direction. The number of introduced functions is reduced and avoids their overlapping. Illustration of the vertical refinement for Figure 12(a) and horizontal for Figure 12(b).

This method allows us to consider refinements that are no longer squares or rectangles. Figure 13 illustrates the use of the improved full span scheme for a quadratic 3D mesh. This refinement does not lead to overloading and ensures linear independence. For 3D refined models, this scheme has to be respected on each plane of the three directions, which is the case for Figure 13.

4.2. 3D Locally Refined B-Splines : algorithm

The refinement procedure provided by Johannessen et al. [21] is modified to be used in 3D. This procedure consists of 3 steps:

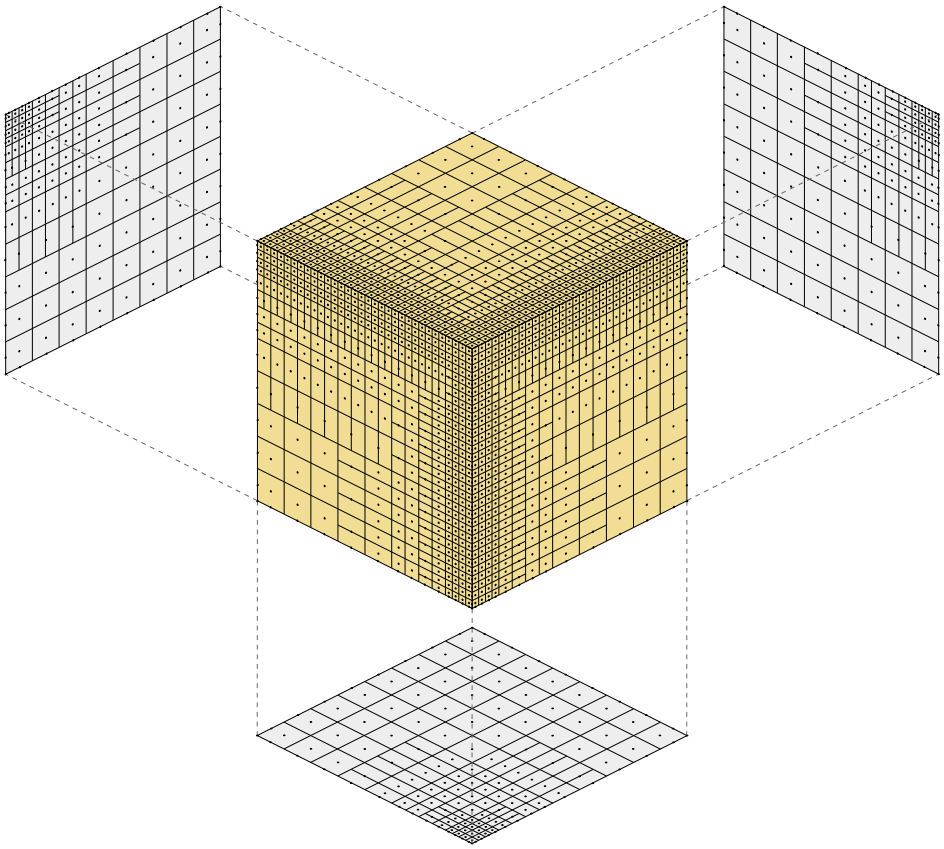


Fig. 13: L-shaped refinement taking into account the remark on the re-entrant corners refinement. There are no overloaded elements in this mesh. Gray planes show hidden faces where can be retrieved a full span pattern on all three sides.

- Considering the element standpoint, group the information of the elements to be refined, i.e. parametric direction and number of refinements. Group these elements and construct the mesh surface that corresponds to the refinement of this element or group of elements.
- Depending on the local knot vectors that are stored in memory, find and split any B-Spline whose support is completely cut by the new mesh surface.
- For all new basis function splits, check if their support is completely traversed by any existing mesh surface. In this case, split it again according to this existing mesh surface.

The split of a function by a mesh surface is described in Algorithm (1) for the 3D case.

At any time in the refinement of the initial mesh, the support of the functions which have been modified must be analyzed with respect to the elements which are in their neighborhood. In this way, we can ensure the absence of overloaded elements and thus the linear independence of the basis. This analysis is also done at the end of the refinement, when all the mesh surfaces have been applied, as an a posteriori test which validates the refinement.

5. LR B-Splines implementation in the Radioss Solver

3D IGA is implemented in Radioss by introducing a new part property which is an element type. It allows a user to define a 3D unrefined global model consisting of one or more patches. Each patch has a polynomial

Algorithm 1 Local ξ-split for the 3D case, from [21]

parameters: $\widehat{\xi}$ {new knot}; B_i {B-Spline to be split ($B_i \in S$)}; S {Spline space}; S_{new} {Functions not present in S}

calculate (α_1, α_2) from Equation (12)
$\Xi \leftarrow SORT(\Xi \cup \widehat{\xi})$
$\Xi_1 \leftarrow [\xi_1, ..., \xi_{p+2}]$
$\Xi_2 \leftarrow [\xi_2, ..., \xi_{p+3}]$
$\Psi_1 \leftarrow \Psi_i$
$\Psi_2 \leftarrow \Psi_i$
$Z_1 \leftarrow Z_i$
$Z_2 \leftarrow Z_i$
if $(\Xi_1, \Psi_1, Z_1) \in S$ **then**
    $\boldsymbol{c_1} \leftarrow (\boldsymbol{c_1}\gamma_1 + \boldsymbol{c_i}\gamma_i\alpha_1)/(\gamma_1 + \alpha_1\gamma_i)$
    $\gamma_1 \leftarrow \gamma_1 + \alpha_1\gamma_i$
else
    $\boldsymbol{c_1} \leftarrow \boldsymbol{c_i}$
    $\gamma_1 \leftarrow \alpha_1\gamma_i$
    add B_1 to S_{new}
end if
if $(\Xi_2, \Psi_2, Z_2) \in S$ **then**
    $\boldsymbol{c_2} \leftarrow (\boldsymbol{c_2}\gamma_2 + \boldsymbol{c_i}\gamma_i\alpha_2)/(\gamma_2 + \alpha_2\gamma_i)$
    $\gamma_2 \leftarrow \gamma_2 + \alpha_2\gamma_i$
else
    $\boldsymbol{c_2} \leftarrow \boldsymbol{c_i}$
    $\gamma_2 \leftarrow \alpha_2\gamma_i$
    add B_2 to S_{new}
end if
remove B_i from S.

degree and corresponding knot vector for each of the three parametric directions. Each knot interval in the global knot vector defines an isogeometric element. Within one patch, these elements are stored with their local indices in the corresponding global knot vectors and with all the non-zero basis functions on the knot interval.

5.1. Local knot vectors

Local knot construction is a fundamental part of the implementation for the refinement procedure but also in the calculation of stress, strain, displacement, outputs. These local knots are built as soon as the model is read from the global knots vectors and stored by patch in dynamic tables. These real values are thus attached to each control point and depend on the polynomial degree of each patch. It is important to have a suitable structure because some control points can be shared by several patches and have several local knot vectors, possibly of different lengths.

It may be noted that with all the refinement methods that have been presented we can not use the classical B-Spline basis functions calculation algorithm given by Piegl and Tiller [37] which uses global knot vectors. The calculation of the basis functions and their derivatives will have to be done one at a time using the local knots.

5.2. 3D Non locally-refined B-Splines

3D non-locally-refined B-Splines implementation in Radioss derives from the LR B-Splines implementation. It is similar in the patches and global knot vectors declaration. The B-Spline functions can be seen as a simplified version of LR B-Splines and the global knot vectors can directly be used for the estimation of the shape functions, using algorithms of Cottrell et al. [2] and Piegl and Tiller [37]. This has been used to implement IGA in commercial software such as Abaqus [51], or alternatively using Bezier extraction [52] as in LS-Dyna [32, 33, 53].

5.3. Contact formulations

The non interpolatory nature of control points in IGA prevents the direct use of the same geometrical approach than standard FE for contact. Unlike FE nodes, they generally do not lie on the physical surface. Several contact formulations have been developed in the last years for IGA, with penalty, Lagrange multiplier or mortar methods, with several discretization approaches. Temizer et al. [54] and Lu [55] introduced a Gauss point to segment contact formulation, whereas Matzen et al. [56], Matzen and Bischoff [57] use collocation points. Mortar contact formulation was used by Temizer et al. [58] for NURBS and another contact formulation for Hierarchical B-Splines using Mortar method was introduced by Temizer and Hesch [38].

For the IGA implementation in Radioss, an existing interface based on the node to surface approach was chosen to be used and to integrate the B-Spline and LR B-Spline models. This choice allows to benefit from the efficiency of the existing algorithms (sorting and contact candidates selection for example) by minimizing the programming effort required: the Type 7 interface, based on non linear penalty [59]. This node to surface interface works with a set of segments, initially established as the outer faces of the master finite elements, and a set of points, consisting of the nodes of the slave element. This formulation is similar to what was introduced by Matzen et al. [56]. The interface stiffness used for the penalty is estimated according to the type of element. It is therefore necessary to define a discretization of the surfaces and the isogeometric elements. Intermediate contact nodes, called ghost nodes, are defined and interpolate the B-Spline geometry. There are 4 ghost nodes per parametric direction and knot interval. They form 9 linear facets on each external face of the element. This discretization of the surfaces is similar to the discretization that is done for the post-processing. Figure 14 completes the description of the contact by the result of a simulation. Compared to a classical finite element formulation and for a similar element size, the B-Spline discretization is more robust because it allows a better representation of the surfaces as well as a better distribution of the contact forces thanks to the extent of the Spline basis functions.

5.4. Critical time step estimation

As classical finite element, the implementation of LR-B-Splines in an explicit solver requires to compute the critical time step associated to this new type of element. Several time step estimations are given in the litterature and can be used in isogeometric analysis. Nodal time step [60–64], or time step defined by the characteristic length [64, 65] constitute one example, also used for FEA. The power iteration method was introduced by Wilkinson [66], used in the FEA context by Benson [61] and later adapted to IGA by Benson et al. [32]. It gives an estimation of the highest natural frequency of the system. A critical time step can then be calculated from this value. All these methods have been implemented for IGA in Radioss.

6. Numerical examples

In this section, few numerical examples will be reviewed, illustrating the LR B-Splines and B-Splines implementation capabilities.

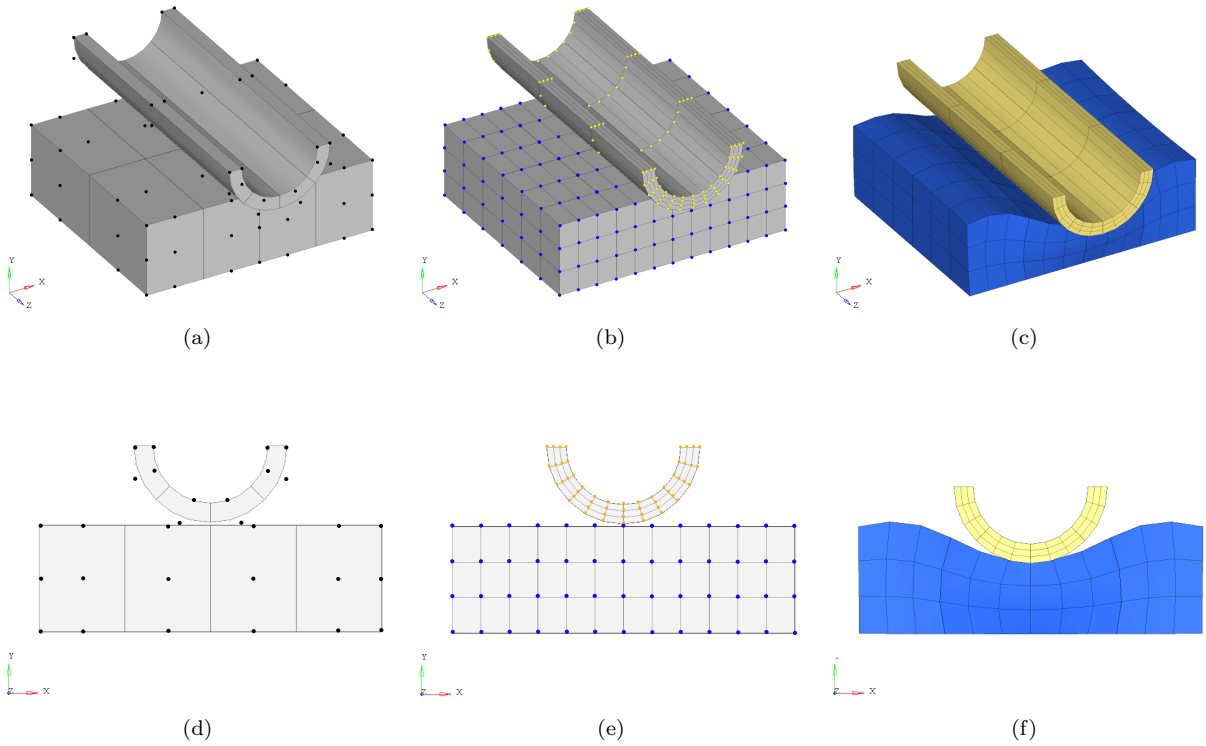


Fig. 14: 3D Contact case. (a,d) NURBS mesh and control points (b,e) Contact surfaces and ghost contact nodes (c,f) deformed NURBS mesh.

6.1. Square Taylor bar impact

This simulation consists in a square rod impacted on a rigid wall with an initial velocity of 227 m.s^{-1}. The rod is 32.4 mm long and has a side of 3.2 mm, see Figure 15(a). This simulation can be refered to experimental studies done for cylindrical rods cases by Rakvåg et al. [67]. Several materials and impact velocities are used in theses experimentations and allow to see an extended range of behavior, see Figure 15(b). Experimentations show that the high velocity impact causes severe deformation at the rod's end. It will cause mesh distortion in the corresponding simulations for elements near the front end of the rod, dramatically reducing the stable time increment during the solution. Numerical simulations have also been done by Kamoulakos [68] and Rakvåg et al. [69] with several softwares. We have used the square version of the Taylor bar for comparison with the reference example available in the Radioss database.

In this case, an isotropic elastic plastic with isotropic hardening material is used to represent the copper behavior. The Young's modulus is set to 117 GPa, the yield stress is defined as 400 MPa, the tangent modulus is 0.1 GPa, the density is 8930 kg.m^{-3} and Poisson's ratio is 0.35. The boundary conditions and dimensions are showed in Figure 15(a).

Three models are used here and represent one quadrant of the rod, due to the inherent symmetry: C^1 quadratic B-Splines, C^1 quadratic LR B-Splines, both with only one patch, and one finite element model with 20 nodes quadratic FE (S20 in Radioss element library). Element densities are identical for the first and third models, that is 256 elements. The LR B-Spline model has the same element density in the bottom-part of the bar, but 50% coarser on the upper-part of the bar, which results in 124 elements in total. These meshes are given in Figure 16. The appropriate boundary conditions prescribed on each of the two symmetry

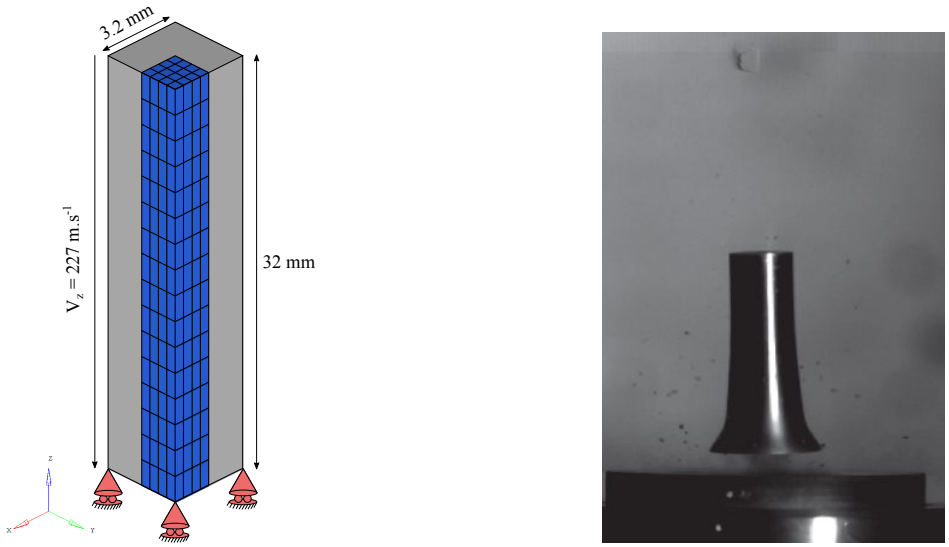


Fig. 15: a) Dimensions, boundary conditions and one of the mesh for one quadrant of the bar. b) High-speed video image from Taylor bar impact tests with nominal impact velocities of 300 m.s^{-1}. The image shows typical impact for an unhardened cylindrical projectile with mushroom deformation, from [69].

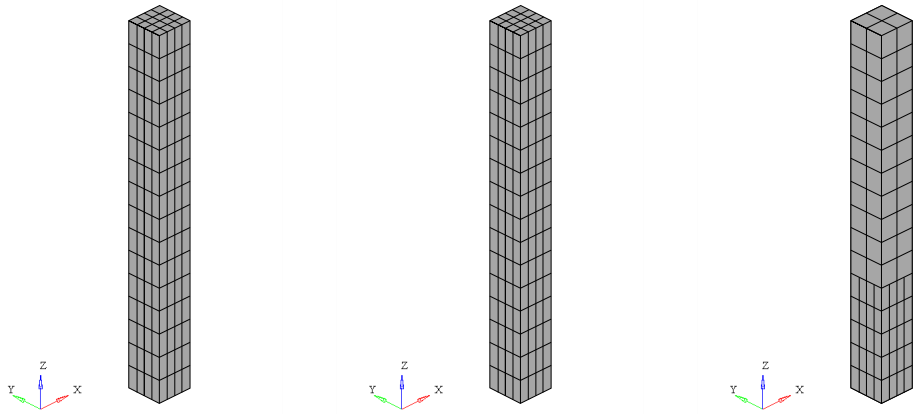
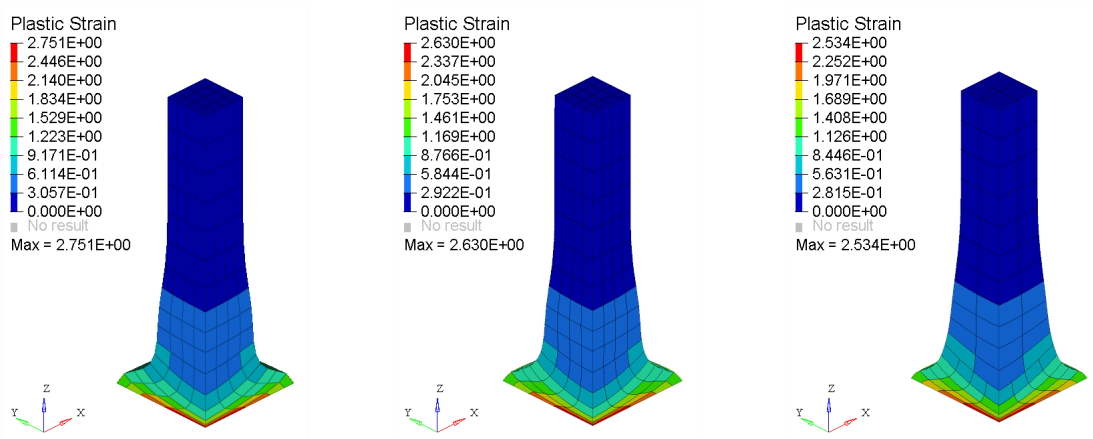


Fig. 16: Discretization used for the square Taylor bar case. a) FE 20-nodes quadratic model, b) C^1 quadratic B-Spline model and c) C^1 quadratic LR B-Spline model with local refinement.

planes for the problem are defined. On the bottom face of the bar, the vertical component of the kinematic fields is fixed. The other components are free. An initial vertical velocity of 227 m.s^{-1} is imposed on all the node/control points.

Results are shown in Figure 17 and reproduce the mushrooming behavior observed in [69]. From these figures it is clear that extremely high plastic strains develop at the crushed extremity of the rod, close to the center of the bar, resulting in severe local mesh distortion. Kamoulakos [68] gives the total aquivalent



(a) (b) (c)

Fig. 17: Plastic strain on all three models at the end of the calculation. a) FE 20-nodes quadratic model, b): C^1 quadratic B-Spline model and c) C^1 quadratic LR B-Spline model with local refinement.

plastic strain expression at the center of the rod for the cylindrical rod case:

$$\epsilon_p = \frac{C_E^2 - C_p^2}{C_E^2 C_p}\left(V - \frac{Y}{\rho C_E}\right) = 2.02, \tag{15}$$

with C_E and C_p are the elastic and plastic waves speeds respectively and V the initial velocity of the rod. This value will be considered as a reference value in the square rod case.

	Quadratic FE S20	B-Splines C^1 quad	LR B-Splines C^1 quad
Number of elements	256	256	4×11+16×5=124
Number of nodes	1505	648	388
Relative time step	1	6.25	6.25
Relative number of cycles	1	0.28	0.28
Relative total CPU time	1	0.56	0.66
Maximum plastic strain	2.751	2.630	2.534

Table 1: Results comparison for all three models. FE ones are used as a reference.

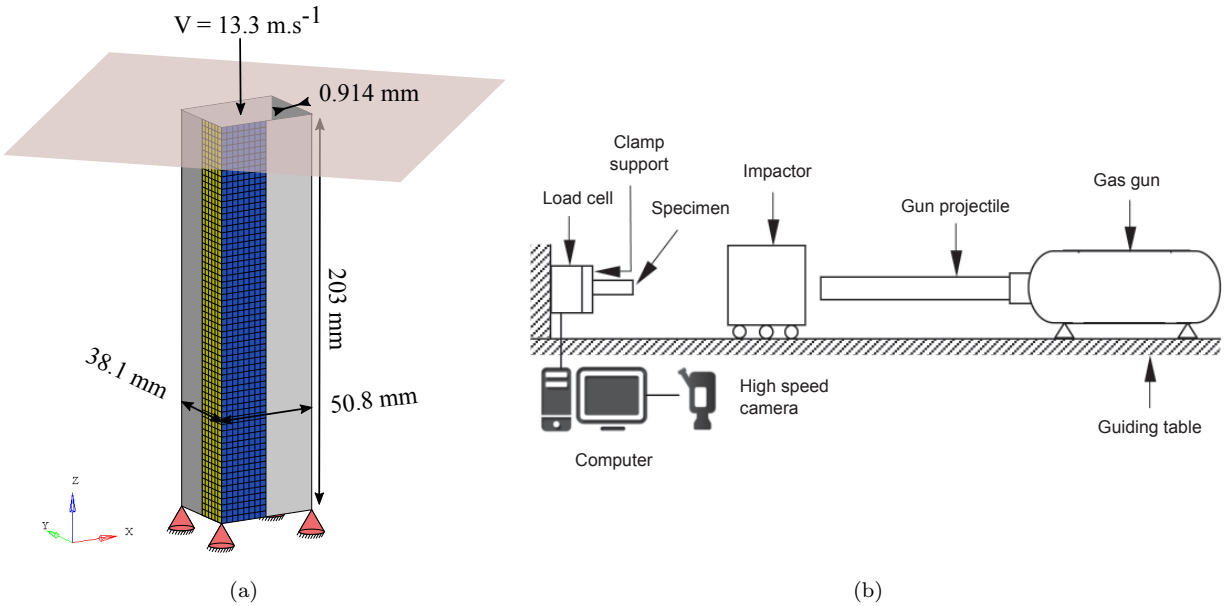
Table 1 groups informations about all meshes in terms of number of elements and number of nodes. It also gives minimal time step during the run, computational time and total number of time steps. For comparison purposes, we have considered as a reference the results obtained with the finite element model, values for other models are normalized with respect to these ones. This table shows that Spline based models are at least 40% faster. This is mainly due to a much higher time step for those models, about 6 times higher, which greatly reduces the total number of time steps. The number of elements is the same between the B-Spline model and the FE one, however the C^1 continuity inside the patch allows us to have fewer control points. The LR B-Spline model emphasizes this point even more because of the coarser discretization on the upper-part of the rod. Only the lower part is strongly deformed. The quality of the obtained solution is also better for the Spline models, with a maximum plastic strain closer to the reference value.

6.2. Axial crushing of a thin-walled beam

The dynamic axial crushing of a thin-walled rectangular box beam against a rigid wall is a typical benchmark in transient simulation. It is a conventional energy absorption device that is applied especially for frame structure of land and railway vehicles. Several experimental and numerical studies were done on this industrial case, see, e.g. [70] and Figure 18(b). The purpose of this example is to study the discretization influence on simulation results, comparing a classical FE model and a B-Spline one, for geometric non-linearities with self-contact and buckling.

The model we consider is taken from Radioss example library. It represents an aluminium hollow beam, leaned on its lower face and crushed by a rigid wall on the upper face with a descending vertical initial velocity of 13.3 m.s^{-1}. The boxbeam has a thickness of 0.914 mm, a hollow square section of 50.8 mm by 38.1 mm and a height of 203 mm, see Figure 18(a). A 0.1% random noise is introduced on the entire structure, on control points/nodes coordinates to trigger the buckling. This perturbation is different from the one used in Benson et al. [32], Belytschko et al. [63] where it is positioned at a specific height from the base and not on the entire box beam. As will be seen in the results, imposing such a perturbation triggers the buckling on the upper-part of the beam and not where the perturbation is imposed. Only a quarter of the beam is modeled due to the inherent symmetry of the problem. The appropriate symmetry boundary conditions are prescribed. The B-Spline parametrization of the quarter boxbeam is done with two patches, using one cubic element through the thickness and C^1 quadratic elements in the other directions. The FE model is composed of piecewise linear 8-nodes elements (HA8 in the Radioss element library), with a similar element density. A Type 7 interface is used to capture the self-contact during the crushing.

An isotropic Von Mises elastic plastic material is used. Young's modulus is set to 60.4 GPa and the yield stress is 90 MPa. The initial density is 2.7×10^{-3} g.mm^{-3} and Poisson's ratio is 0.33. Boundary conditions and dimensions are given in Figure 18(a).



(a) (b)

Fig. 18: a) Dimensions, boundary conditions and mesh for the B-Spline model. b) High speed impact testing machine, from Jusuf et al. [70].

The final deformed shapes are given in Figure 19. The computed models have been symmetrized for visualization purpose. We can see that both Radioss models have the same buckling mode: same number and same height of plies. The crushing response (resultant force) on the rigid wall is plotted in Figure 20. It shows at the very beginning of the computation a similar normal force intensity for both models. This peak corresponds to the first contact between the hollow beam and the rigid wall, which triggers the buckling.

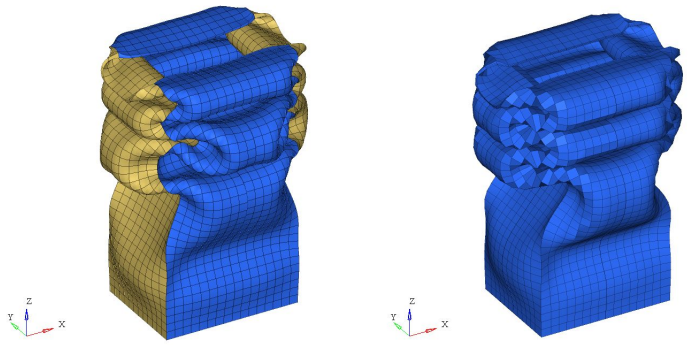


Fig. 19: Boxbeam shape deformations : deformed structure reconstruction using symmetry projection for B-Spline model and FE model.

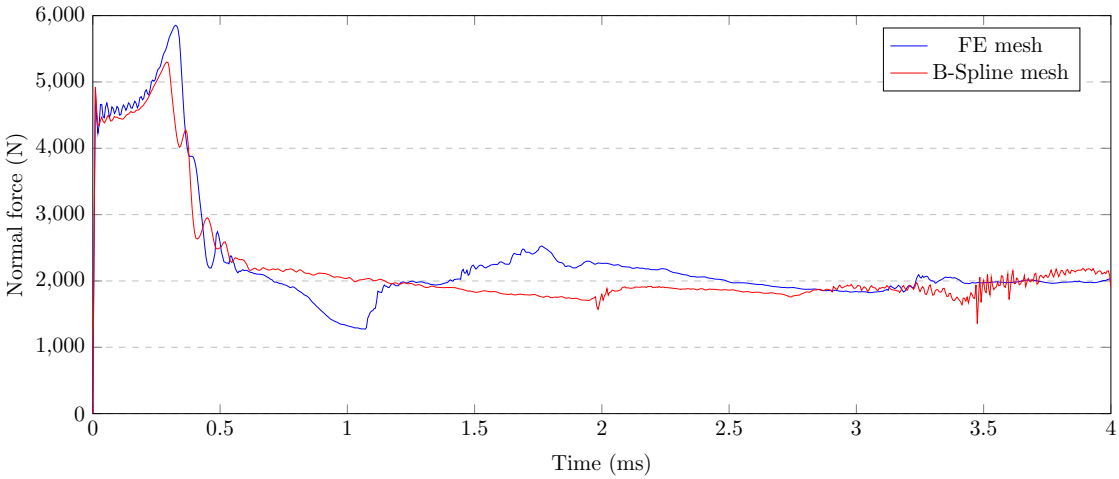


Fig. 20: Normal force on the impactor.

6.3. Came-Valve system

This example illustrates the dynamic behavior and the kinematic motion of a cam-valve system. The cam is rotating with an angular velocity of 314 rad.s^{-1}, which interacts and triggers the translation of a valve tied to two springs. The superposed springs have different stiffness, 30000 N.m^{-1} for the first and 15000 N.m^{-1} for the second one. The springs control the higher and lower angular frequencies. The valve has a radius of 44 mm, the cam has a height of 36 mm and a width of 18 mm. Boundary conditions and dimensions of this model are given in Figure 21.

The cam and the valve are both discretized with 5 patches. For the LR B-Spline model, local refinement has been performed on the outer row to better represent the contact surface with the valve. The FE model is meshed with piecewise linear bricks (HA8 in the Radioss element library).

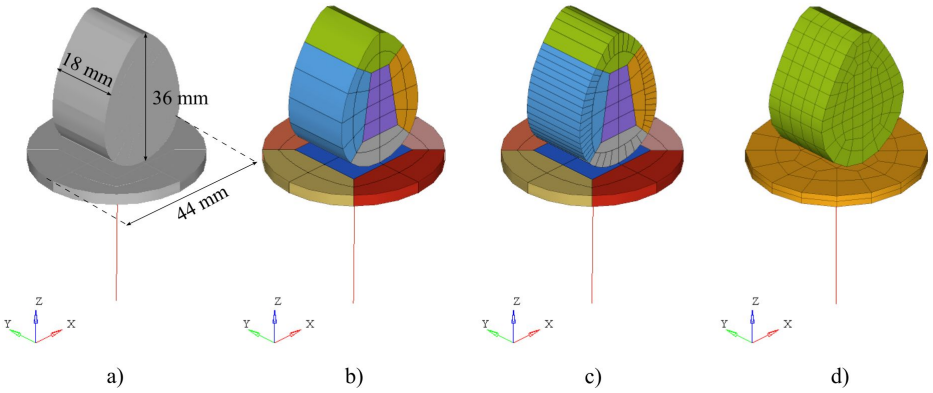


Fig. 21: Dimensions, boundary conditions and meshes. From left to right : a) Cam and valve's dimensions, b) C^1 quadratic B-Spline mesh, c) C^1 quadratic LR B-Spline mesh and d) FE quadratic 20-nodes mesh.

The material used for the cam and the valve is steel with an initial density of 7.8×10^{-3} g.mm^{-3}, a Young's modulus of 210 GPa, and Poisson's ratio of 0.3. The kinematics of the simulation is given in Figure 22.

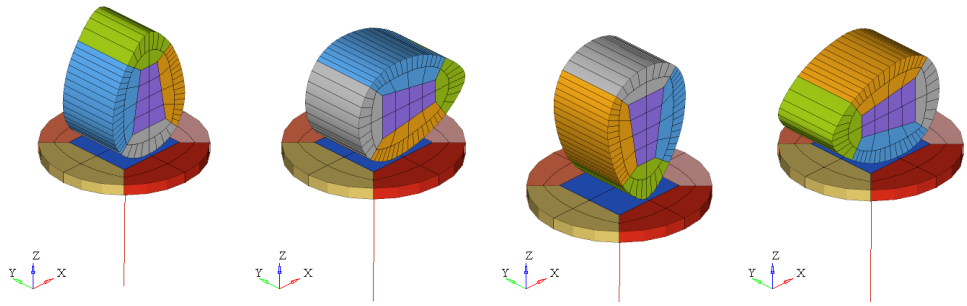


Fig. 22: Cam and valve positions during the simulation, for the LR B-Spline model.

Figure 23 provides the velocity of the valve's master node as a function of time. The raw results obtained are noisy for all models. This is due to the faceted description of the geometry in the contact algorithm, as explained in section 5.3.

The oscillations correspond to each application of a contact force to the valve. Note that for the FE mesh, they have a larger amplitude, even compared to the coarse B-Spline mesh. Although the FE mesh is finer than the B-Spline one, the contact surface discretization of the latter is finer which results in a slightly smoother response. Due to the local refinement, the contact surface discretization of the LR model is much finer which drastically reduces the oscillations.

A smooth velocity curve can be obtained by using a low pass filter. The filtering quality depends on the number of samples which in this case is the number of points computed by Radioss for each curve. The smooth velocity curves are shown on Figure 24. Results of all the models are nearly identical.

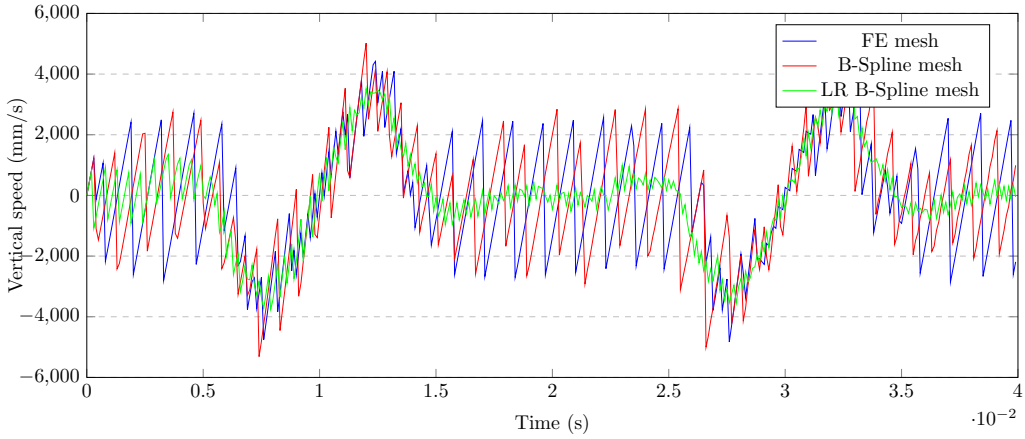


Fig. 23: Valve's master node vertical velocity.

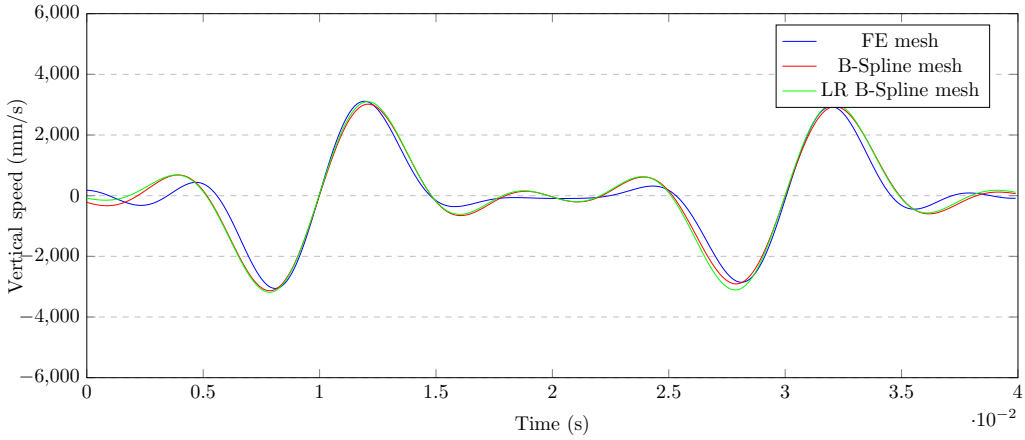


Fig. 24: Filtered valve's master node vertical velocity.

6.4. Stamping simulation

This example consists in the stamping process of a steel sheet. The stamping tools include a punch, a die and a blank holder. The initial thickness of the metal sheet is 1 mm, the length is 483 mm and the width is 281.4 mm. A resultant force of 200 kN is vertically applied on the blank holder in order to flatten the sheet against the die. We apply an evolving vertical velocity on the punch with a maximum value of 5 m.s^{-1}. Taking symmetry into account, only one half of the structure is modeled. Parts, boundary conditions and loads are shown in Figure 25. The punch is shown in yellow, the blank holder in transparent green and the die in blue. The sheet is colored in orange.

An isotropic elastic plastic material is used. The material parameters are 210 GPa for Young's modulus and 270 MPa for the yield stress. The initial density is 7.8×10^{-3} g.mm^{-3} and Poisson's ratio is 0.33. We consider four different models to represent the sheet: an adaptative piecewise linear 4-nodes shell model (QEPH in the Radioss element library), a piecewise linear 8-nodes model (HA8 in the Radioss element library), a C^1 quadratic B-Spline model, and a C^1 quadratic LR B-Spline one. LR B-Spline model allows to reduce the number of elements with B-Spline model from 2048 to 1326. The number of control points is reduced to 2574. All the three solid models use one element through the thickness. The other parts, the punch, the blank holder and the die are meshed with FE triangular rigid shell elements. Final shape

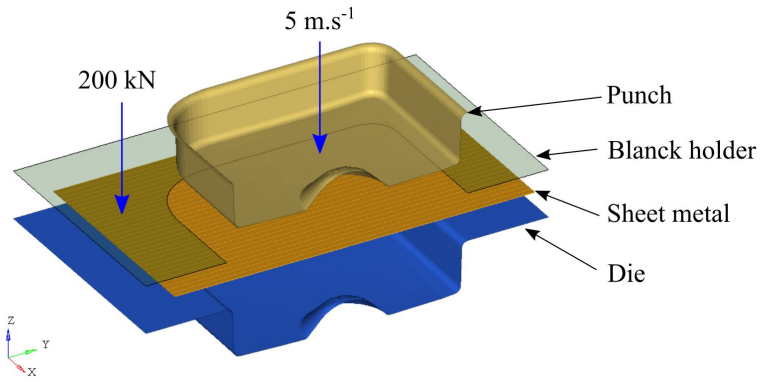


Fig. 25: Half-model exploded and boundary conditions.

deformations are given on Figure 26.

The final shape of the sheet is studied. Table 27(b) lists the amounts of draw-in in the X, Y and diagonal directions named by DX, DY1, DY2, DXY1 and DXY2 at punch travels of 69 mm, see Figure 27(a). In comparison with the two values obtained with FE model, B-Spline and LR B-Spline models seem to be a little stiffer than the FE shell one but a little more flexible than the FE solid one. Figure 28 shows that the most stressed zones are located at the edges of the punch. The Von Mises stress is up to 584.9 MPa for all four models, well above the elastic limit of the material. We can see in Figure 29 the maximum plastic strain for all four models, corresponding to the stressed areas.

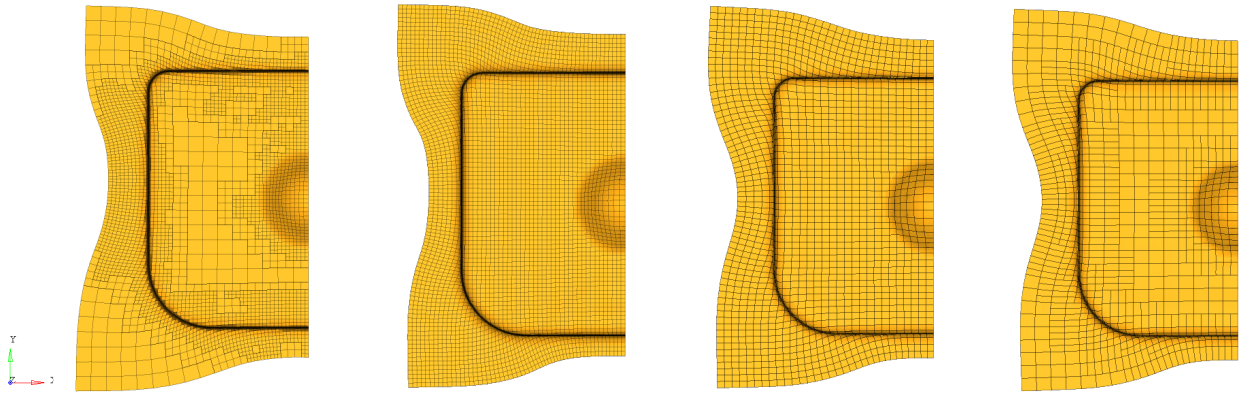
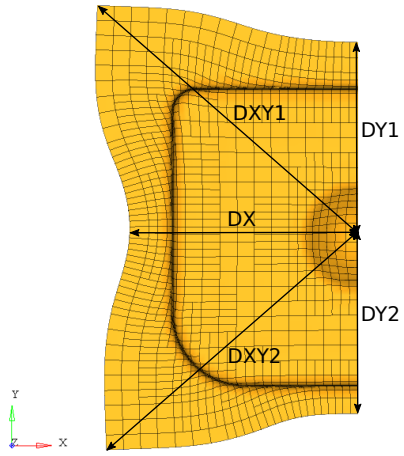


Fig. 26: Half-model shape deformation for all models, for a 69 mm punch travel. From left to right: the adaptative FE Shell model, the FE S20 model, the C^1 quadratic B-Spline model and the C^1 quadratic LR B-Spline model.

6.5. Cylindrical tube under pressure

This example illustrates the nonlinear buckling of an aluminum cylindrical tube under external pressure, with self-contact. Experimental tests have been performed by Farhat et al. [71], see Figure 30. Numerical simulations were performed with isogeometric shells by Benson et al. [32].

The cylindrical tube has an internal radius of 19 mm and an external radius of 19.7 mm, a total length of 177.8 mm. Two plugs at the ends of the tube apply an axial compression, which is correlated with the



(a)

	DX	DY1	DY2	DXY1	DXY2
Adaptative FE Shell	238.0	192.8	187.4	348.3	357.5
FE S20	227.8	196.6	175.6	350.1	328.9
B-Spline C^1 quad	233.5	197.1	180.4	356.3	336.3
LR B-Spline C^1 quad	233.9	199.7	180.4	357.1	337.2

(b)

Fig. 27: Distances between corners and edges and the center of the sheet for a half-model shape deformation. b) Draw-in for the sheet (unit: mm) for a 69 mm punch travel. Results comparison for all four models.

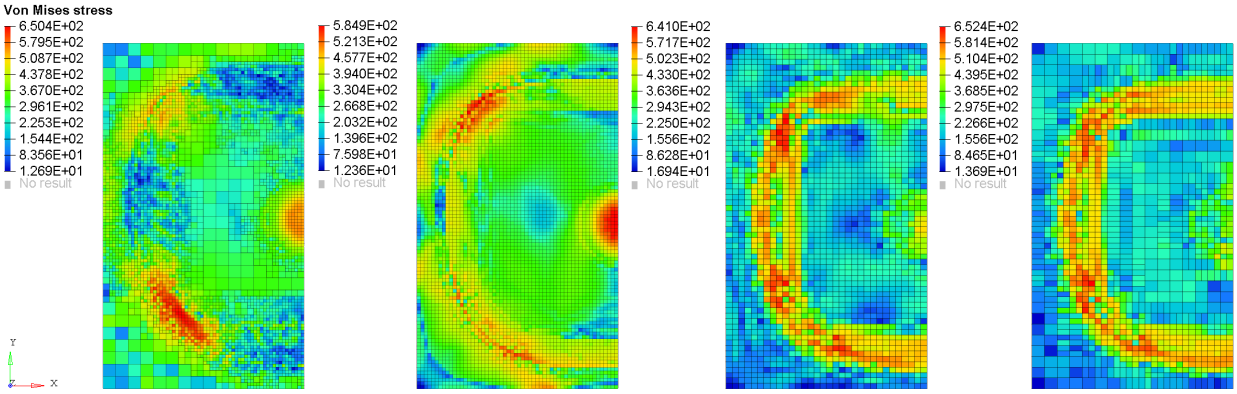


Fig. 28: Von Mises stress on all half-models, for a 69 mm punch travel. From left to right: the adaptative FE Shell model, the FE S20 model, the C^1 quadratic B-Spline model and the C^1 quadratic LR B-Spline model.

imposed pressure, see Figure 30. These plugs are inserted over a length of 25.4 mm. An external pressure is applied to the cylinder, reproducing the pressure difference between the inside and the outside. This pressure is carried out in a ramp of 1 ms then is held constant until the equilibrium of the structure.

An isotropic elastic plastic material with a user-defined stress-strain curve is used. For this simulation,

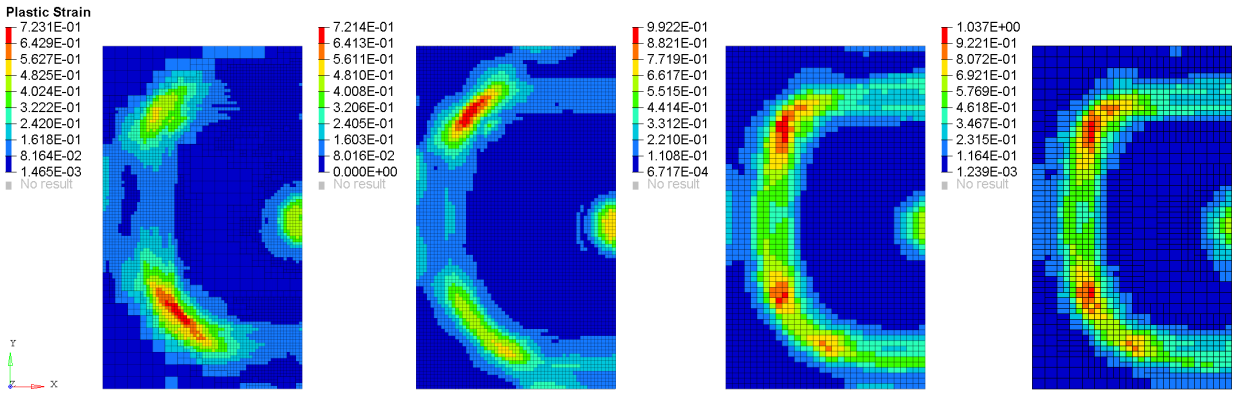


Fig. 29: Plastic strain on all half-models, for a 69 mm punch travel From left to right: the adaptative FE Shell model, the FE S20 model, the C^1 quadratic B-Spline model and the C^1 quadratic LR B-Spline model.

the alloy has a Young's modulus of 69.5 GPa, a yield stress of 227 MPa. The initial density is 1073 kg.m^{-3} and Poisson's ratio is 0.3.

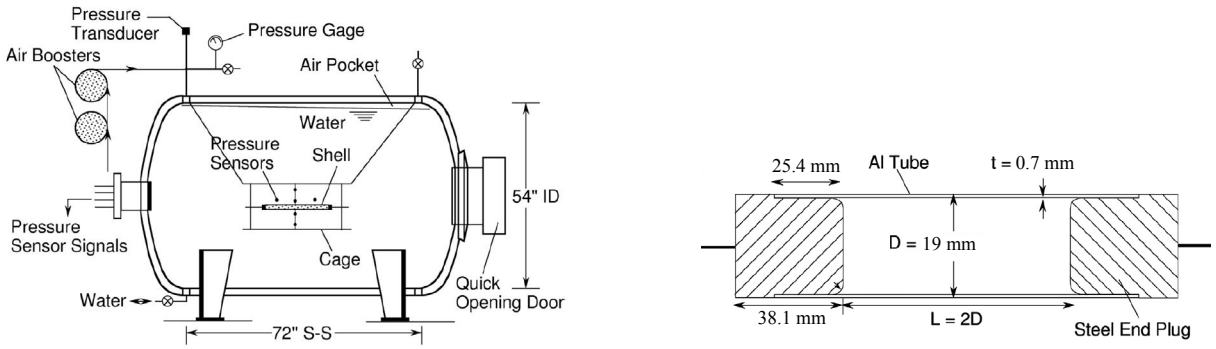


Fig. 30: Kyriakides' experimental protocol and boundary conditions. Pictures taken from Farhat et al. [71]. Note that dimensions on the right picture were converted to the metric system for consistency.

Figure 31(a) illustrates the boundary conditions and the dimensions. Only one half is modeled due to the symmetry. The compressive loading imposed by the fluid pressure on the end plug is modeled by a distributed force on the planar end surface. The kinematic boundary conditions corresponding to the rigid contact between the tube and the plug are modeled by imposing zero radial displacement on the corresponding surface. The cylinder is parametrized using three C^1 quadratic NURBS patches, with C^0 interfaces between each of them. A C^0 circumferential line is added to represent the boundary between the constrained and free portions of the tube. The imposed external pressure is applied with a ramp of 1 ms then is held constant until the equilibrium of the structure is reached.

We use a mesh composed of 960 quadratic elements, keeping the fact that only one element is used through the tube's thickness. Two kinds of geometric perturbation are introduced, distributed on the cylinder section in order to trigger the buckling in the third mode. These distributed noises are defined scaling the x and y coordinates of the control points by $1 + \frac{1}{100}cos(3\theta)$ and $1 + \frac{1}{100}cos(3 * (\theta - \frac{\pi}{3}))$. The first noise configuration will push outwards the C^0 patch interfaces, pushing inwards the middle of each patch. The second configuration does the opposite. A Type 7 interface is used to capture the self contact of the inner surface of the cylinder. Cross sections of the two perturbed geometries are shown on Figure 31(b).

Simulation results and final deformed configurations are presented in Figure 33 and show relatively good

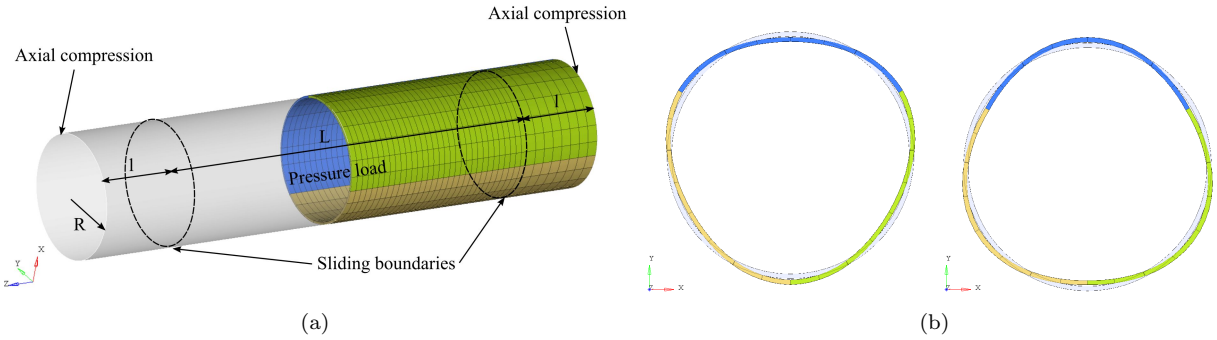


Fig. 31: a) Boundary conditions and mesh. b) Noise repartitions (highlighted at 5% for better visualization). The original shape is also given.

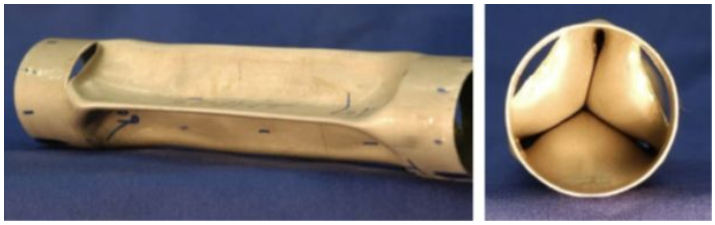


Fig. 32: Kyriakides' experimental result, from [32].

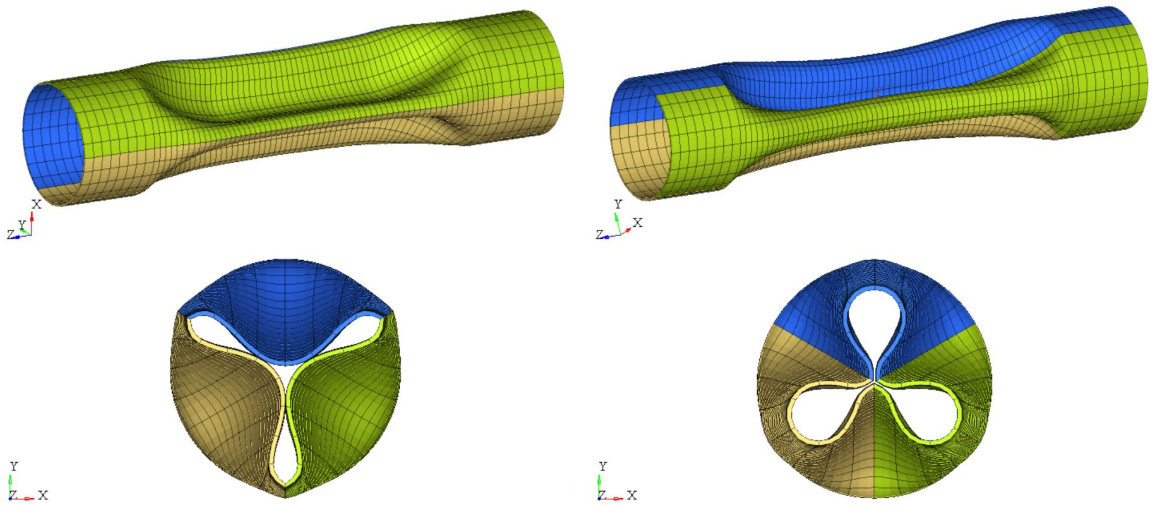
agreement. The experimental results showed a critical pressure of 2.827 MPa. This experimental pressure is used as a reference. For the first perturbation, the critical pressure needed to trigger the buckling on the third mode is 2.930 MPa, i.e. a corresponding error of 3.6% with the analytic reference. The final shape shows the apparitions of three kinks at the lobes, certainly due to the C^0 continuity accross patch interfaces. The second perturbation leads to a different final shape. The kinks are still located at the patch interfaces. The deformed shape is closer to the experimental one in this case, since the experimental deformed configuration has a kink at the triple contact point on the cylinder axis, see Figure 33.

The buckling pressure with the second configuration is relatively higher. We obtain a value of 3.448 MPa, which corresponds to an error of 22%. This difference in the buckling pressure could be an indication that the mesh isn't fine enough to obtain a converged solution. This situation might be amplified by the relative alignment of the geometric perturbation and the C^0 patch interfaces.

A twice finer mesh in the circumferential direction has been used with both perturbations and show identical results. Both simulation results show the same final deformed shape, which is different from the previous ones, but much closer to the experimental one. We can note that with such discretization the geometric perturbation is not mandatory anymore to trigger the buckling in the third mode. The critical pressure required to cause the buckling is lowered to 2.41 MPa, which coresponds to an an error of 14.7%. This is lower than the experimental buckling pressure. Nevertheless, we can remark that the cylinder bounces back outward after its implosion, and creates the secondary lobes near the axis, see on Figure 34. These secondary lobes are missing from the experimental results, and have been reported by Benson et al. [32].

6.6. Cardboard box drop test

This example models the drop test of a simplified cardboard box on its corner and edge. The interest of this example is that it combines in the same model several contact interfaces and a hybrid discretization



(a) (b)

Fig. 33: Buckling of the meshes with the two noise configuration. a) Kinks appear on the lobes with a buckling pressure of 2.930 MPa. b) Kinks appear at the cylinder's center with a buckling pressure 2.827 MPa. Both figures are extracted when the equilibrium of the structure is reached.

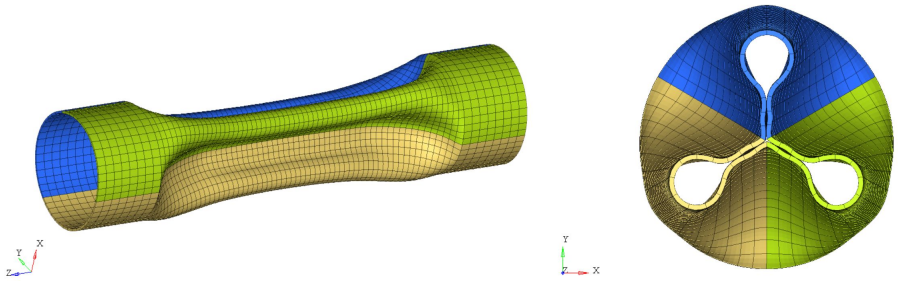


Fig. 34: Same deformed shape for both finer meshes. The appearance of sidelobes buckling can be noticed. Buckling pressure of 2.41 MPa, when the equilibrium of the structure is reached.

with FE elements and B-Splines. The cardboard is composed of several parts with different materials. The box inners, colored in pink on Figure 35, are represented by aluminium blocks and are connected at their sides by rubber pads (in blue) to a steel housing (in light red). This housing rests at its corners on eight styrofoam cushions (in green). Finally, a cardboard (in orange) packs all the parts. The purpose of this example is to check that the packing of the inners is efficient: energy absorption...

The cardboard box and the styrofoam cushions are discretized with C^1 quadratic B-Splines. The housing is discretized with piecewize linear shell elements. For all the other parts, piecewize linear FE bricks are used. Type 7 contact interfaces are defined between the cardboard box, the styrofoam cushions and the housing. All other part connections (e.g. between inners and rubber pads) are imposed by merging nodes with compatible meshes. An initial vertical velocity of 5 $\mathrm{m.s^{-1}}$ is set as well as a gravity field, which represents the drop. A rigid wall is fixed in front of the corner or the edge of the cardboard, depending on the case considered.

Material models used in this example are elastic for inners, elastic perfectly plastic for the cardboard and for the housing, foam-plastic for all the styrofoams, and hyperelastic for the rubber. The cardboard is

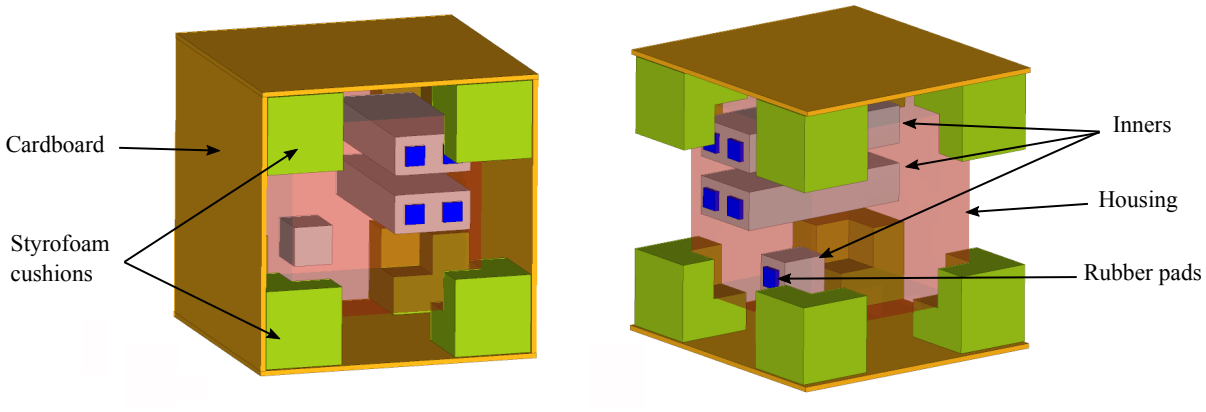


Fig. 35: The cardboard box and its internal parts. Some parts of the cardboard were removed for a better visualization.

considered as isotropic for simplicity.

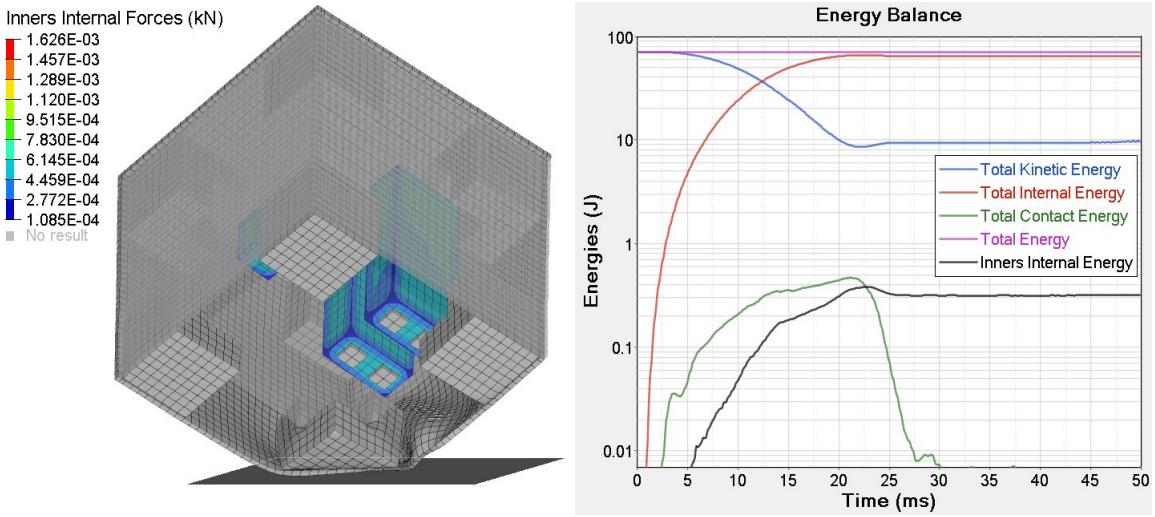


Fig. 36: Deformation and energy balance for a corner drop test.

Simulation results for the corner and the egde drop tests are given on Figure 36 and Figure 37. As said previously, the analysis of those results is focused on the energy balance of the inner parts. Energy balances for each test can be plotted to compare the total kinetic energy, the total internal energy, the contact energy and the inners internal energy, in a logarithmic scale. Results show a conversion between kinetic and internal energy during the drop test. Most of the energy is absorbed by the styrofoams and the cardboard and let the housing and the inners safe. Left figures show the internal force magnitude on the inners. We observe that the highest value is at least six times lower than that of the other parts of the model.

7. Conclusion

We have introduced in the present paper an explicit dynamics 3D locally refined isogeometric formulation using LR B-Spline technology. Based on the existing LR B-Spline refinement algorithms, we have introduced

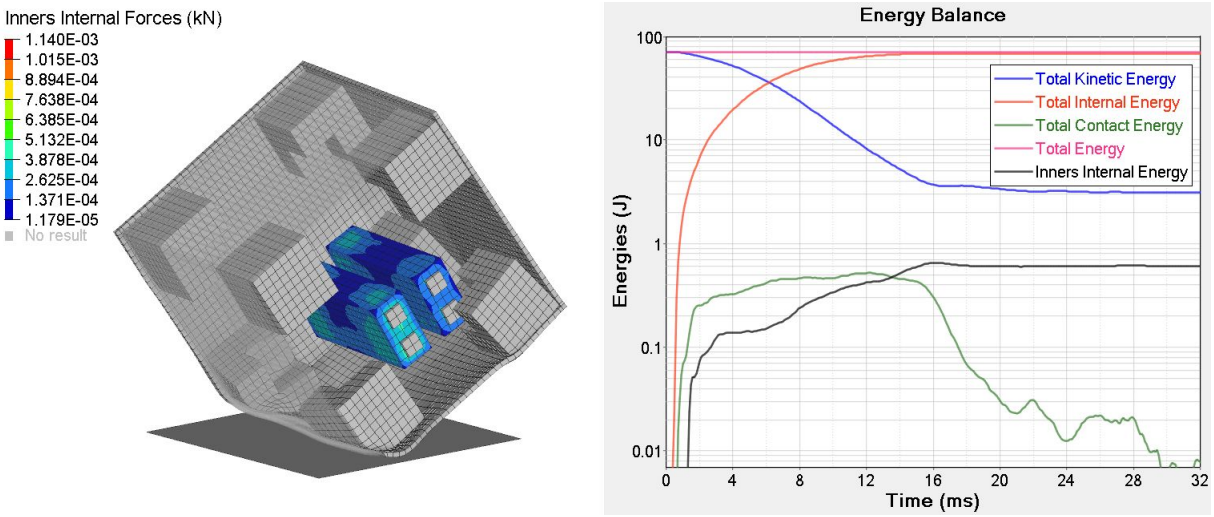


Fig. 37: Deformation and energy balance for an edge drop test.

an improved full span scheme suitable for 2D and 3D cases that prevents any overloading and thus guarantees linear independence of the basis. Specific implementation aspects in Altair Radioss are also addressed, such as time step estimation and contact formulation. The developed tool allows FE-IGA hybrid calculations, with contact, local refinement, for all material behavior models available in Altair Radioss. Several examples illustrate the efficiency of the approach for crash, stamping and dynamic buckling applications.

Appendix A. Pre/Post-processing

Preprocessing is still a delicate step in IGA, especially for 3D models. Efforts have been made to develop robust and exploitable solutions [72–75]. These methods bring the transition from CAD to analysis suitable isogeometric models.

Post processing is based on the principle of projection on a finite element mesh by global least squares methods on each of the B-Spline patches, see for example Elguedj and Hughes [76], Nguyen and Bordas [77]. This projection mesh is defined so that each B-Spline volume element is represented by 27 linear bricks, see Figure A.38. The fields defined on control points are directly interpolated on a linear brick mesh with the basis functions. The ones defined at the Gauss points are first projected using least squares.

References

[1] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (2005) 4135–4195.

[2] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, 1st ed., Wiley Publishing, 2009.

[3] Y. Bazilevs, V. M. Calo, Y. Zhang, T. J. Hughes, Isogeometric fluid–structure interaction analysis with applications to arterial blood flow, Computational Mechanics 38 (2006) 310–322.

[4] J. A. Cottrell, A. Reali, Y. Bazilevs, T. J. Hughes, Isogeometric analysis of structural vibrations, Computer methods in applied mechanics and engineering 195 (2006) 5257–5296.

[5] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Computer Methods in Applied Mechanics and Engineering 197 (2007) 173–201.

[6] Y. Bazilevs, V. M. Calo, T. J. Hughes, Y. Zhang, Isogeometric fluid-structure interaction: theory, algorithms, and computations, Computational mechanics 43 (2008) 3–37.

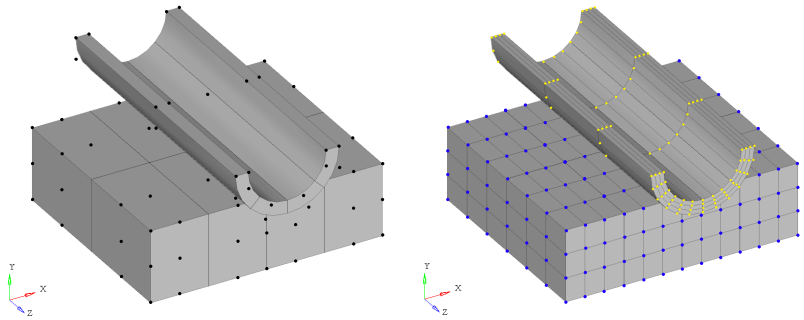


Fig. A.38: Post-processing projection illustration: 2 patches model. On the left, the control points are colored in black on the left and the displayed elements are the isogeometric elements as they have been defined. On the right, definition of the FE projection mesh for the post-treatment: 27 linear bricks per LR B-Spline or B-Spline element. The ghost nodes of these elemental bricks are blue and yellow.

[7] T. J. Hughes, A. Reali, G. Sangalli, Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method nurbs, Computer methods in applied mechanics and engineering 197 (2008) 4104–4124.

[8] W. A. Wall, M. A. Frenzel, C. Cyron, Isogeometric structural shape optimization, Computer methods in applied mechanics and engineering 197 (2008) 2976–2988.

[9] A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 200 (2011) 3554–3567.

[10] C. Giannelli, B. Jüttler, Bases and dimensions of bivariate hierarchical tensor-product splines, Journal of Computational and Applied Mathematics 239 (2013) 162–178.

[11] D. Schillinger, L. Dede, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, T. J. 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 (2012) 116–150.

[12] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, T. J. Hughes, Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations, Computer Methods in Applied Mechanics and Engineering 267 (2013) 170–232.

[13] C. Giannelli, B. Jüttler, H. Speleers, Thb-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (2012) 485–498.

[14] G. Kiss, C. Giannelli, B. Jüttler, Algorithms and data structures for truncated hierarchical b–splines, in: International Conference on Mathematical Methods for Curves and Surfaces, Springer, 2012, pp. 304–323.

[15] C. Giannelli, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon, J. Špeh, Thb-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 299 (2016) 337–365.

[16] R. Churchhouse, Spline functions: Basic theory, 1982.

[17] H. Pottmann, The geometry of tchebycheffian splines, Computer Aided Geometric Design 10 (1993) 181–210.

[18] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and t-nurccs, in: ACM transactions on graphics (TOG), volume 22, ACM, 2003, pp. 477–484.

[19] M. R. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with t-splines, Computer methods in applied mechanics and engineering 199 (2010) 264–275.

[20] T. Dokken, T. Lyche, K. F. Pettersen, Polynomial splines over locally refined box-partitions, Computer Aided Geometric Design 30 (2013) 331–356.

[21] K. A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using lr b-splines, Computer Methods in Applied Mechanics and Engineering 269 (2014) 471–514.

[22] K. A. Johannessen, F. Remonato, T. Kvamsdal, On the similarities and differences between classical hierarchical, truncated hierarchical and lr b-splines, Computer Methods in Applied Mechanics and Engineering 291 (2015) 64–101.

[23] C. Giannelli, B. Jüttler, H. Speleers, Strongly stable bases for adaptively refined multilevel spline spaces, Advances in Computational Mathematics 40 (2014) 459–490.

[24] A. Buffa, D. Cho, G. Sangalli, Linear independence of the t-spline blending functions associated with some particular t-meshes, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1437–1445.

[25] M. A. Scott, X. Li, T. W. Sederberg, T. J. Hughes, Local refinement of analysis-suitable t-splines, Computer Methods in Applied Mechanics and Engineering 213 (2012) 206–222.

[26] A. Bressan, Some properties of lr-splines, Computer Aided Geometric Design 30 (2013) 778–794.

[27] C. de Falco, A. Reali, R. Vázquez, Geopdes: a research tool for isogeometric analysis of pdes, Advances in Engineering Software 42 (2011) 1020–1034.

[28] D. Rypl, B. Patzák, From the finite element analysis to the isogeometric analysis in an object oriented computing environment, Advances in Engineering Software 44 (2012) 116–125.

[29] M. S. Pauletti, M. Martinelli, N. Cavallini, P. Antolin, Igatools: An isogeometric analysis library, SIAM Journal on Scientific Computing 37 (2015) C465–C496.
[30] N. Collier, L. Dalcín, V. M. Calo, Petiga: High-performance isogeometric analysis, arXiv preprint arXiv:1305.4452 (2013).
[31] L. Zhang, D. Eyheramendy, Une approche isogéométrique pour la thermo-mécanique en grandes transformations dans un contexte multiphysique à objets, in: 12e colloque national en calcul des structures, 2015.
[32] D. Benson, Y. Bazilevs, M.-C. Hsu, T. Hughes, Isogeometric shell analysis: the reissner–mindlin shell, Computer Methods in Applied Mechanics and Engineering 199 (2010) 276–289.
[33] D. Benson, Y. Bazilevs, M.-C. Hsu, T. Hughes, A large deformation, rotation-free, isogeometric shell, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1367–1378.
[34] A. Duval, T. Elguedj, H. Al-Akhras, 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, 2015.
[35] 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 (2017) 1648–1660.
[36] D. Benson, D. Bhalsod, S. Hartmann, P. Ho, L. Li, W. Li, A. Nagy, I. Yeh, Isogeometric analysis in ls-dyna: Using cad-geometry for numerical simulation, in: 10th European LS-DYNA Conference. Würzburg, Germany, June, 2015.
[37] L. Piegl, W. Tiller, The NURBS Book (2Nd Ed.), Springer-Verlag New York, Inc., New York, NY, USA, 1997.
[38] I. Temizer, C. Hesch, Hierarchical nurbs in frictionless contact, Computer Methods in Applied Mechanics and Engineering 299 (2016) 161–186.
[39] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, Y. Feng, Polynomial splines over hierarchical t-meshes, Graphical models 70 (2008) 76–86.
[40] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K. Bletzinger, Y. Bazilevs, T. Rabczuk, Rotation free isogeometric thin shell analysis using pht-splines, Computer Methods in Applied Mechanics and Engineering 200 (2011) 3410–3424.
[41] P. Wang, J. Xu, J. Deng, F. Chen, Adaptive isogeometric analysis using rational pht-splines, Computer-Aided Design 43 (2011) 1438–1448.
[42] T. Dokken, Locally refined splines (2010).
[43] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. Hughes, S. Lipton, M. A. Scott, T. W. Sederberg, Isogeometric analysis using t-splines, Computer Methods in Applied Mechanics and Engineering 199 (2010) 229–263.
[44] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, in: ACM transactions on graphics (TOG), volume 23, ACM, 2004, pp. 276–283.
[45] H. Kang, F. Chen, J. Deng, Modified t-splines, Computer Aided Geometric Design 30 (2013) 827–843.
[46] L. Liu, Y. J. Zhang, X. Wei, Weighted t-splines with application in reparameterizing trimmed nurbs surfaces, Computer Methods in Applied Mechanics and Engineering 295 (2015) 108–126.
[47] L. Liu, Y. J. Zhang, X. Wei, Handling extraordinary nodes with weighted t-spline basis functions, Procedia Engineering 124 (2015) 161–173.
[48] E. Evans, M. Scott, X. Li, D. 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 (2015) 1–20.
[49] X. Wei, Y. Zhang, L. Liu, T. J. Hughes, Truncated t-splines: Fundamentals and methods, Computer Methods in Applied Mechanics and Engineering 316 (2017) 349–372.
[50] A. Bressan, B. Jüttler, A hierarchical construction of lr meshes in 2d, Computer Aided Geometric Design 37 (2015) 9–24.
[51] 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.
[52] 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 (2015) 55–105.
[53] D. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, T. Hughes, Blended isogeometric shells, Computer Methods in Applied Mechanics and Engineering 255 (2013) 133–146.
[54] I. Temizer, P. Wriggers, T. Hughes, Contact treatment in isogeometric analysis with nurbs, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1100–1112.
[55] J. Lu, Isogeometric contact analysis: Geometric basis and formulation for frictionless contact, Computer Methods in Applied Mechanics and Engineering 200 (2011) 726–741.
[56] M. Matzen, T. Cichosz, M. Bischoff, A point to segment contact formulation for isogeometric, nurbs based finite elements, Computer Methods in Applied Mechanics and Engineering 255 (2013) 27–39.
[57] M. Matzen, M. Bischoff, A weighted point-based formulation for isogeometric contact, Computer Methods in Applied Mechanics and Engineering 308 (2016) 73–95.
[58] I. Temizer, P. Wriggers, T. Hughes, Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with nurbs, Computer Methods in Applied Mechanics and Engineering 209 (2012) 115–128.
[59] G. Willkenmuller, K. Kayvantash, Radioss theoretical manual, MECALOG archives (1999).
[60] D. Flanagan, T. Belytschko, Eigenvalues and stable time steps for the uniform strain hexahedron and quadrilateral, Journal of applied mechanics 51 (1984) 35–40.
[61] D. J. Benson, Stable time step estimation for multi-material eulerian hydrocodes, Computer methods in applied mechanics and engineering 167 (1998) 191–205.
[62] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Dover Publications, 2012.

[63] T. Belytschko, W. K. Liu, B. Moran, K. Elkhodary, Nonlinear finite elements for continua and structures, John wiley & sons, 2013.
[64] S. Hartmann, D. J. Benson, Mass scaling and stable time step estimates for isogeometric analysis, International Journal for Numerical Methods in Engineering 102 (2015) 671–687.
[65] J. O. Hallquist, et al., Ls-dyna theory manual, Livermore software Technology corporation 3 (2006) 25–31.
[66] J. Wilkinson, The algebraic eigenvalue problem, in: Handbook for Automatic Computation, Volume II, Linear Algebra, Springer-Verlag New York, 1971.
[67] K. G. Rakvåg, T. Børvik, I. Westermann, O. S. Hopperstad, An experimental study on the deformation and fracture modes of steel projectiles during impact, Materials & Design 51 (2013) 242–256.
[68] A. Kamoulakos, A simple benchmark for impact, Bench Mark (1990) 31–35.
[69] K. G. Rakvåg, T. Børvik, O. S. Hopperstad, A numerical study on the deformation and fracture modes of steel projectiles during taylor bar impact tests, International Journal of Solids and Structures 51 (2014) 808–821.
[70] A. Jusuf, T. Dirgantara, L. Gunawan, I. S. Putra, Numerical and experimental study of single-walled and double-walled columns under dynamic axial crushing, Journal of Mechanical Engineering 9 (2012) 53–72.
[71] C. Farhat, K. Wang, A. Main, S. Kyriakides, L.-H. Lee, K. Ravi-Chandar, T. Belytschko, Dynamic implosion of underwater cylindrical shells: experiments and computations, International Journal of Solids and Structures 50 (2013) 2943–2961.
[72] H. Al Akhras, T. Elguedj, A. Gravouil, M. Rochette, Isogeometric analysis-suitable trivariate nurbs models from standard b-rep models, Computer Methods in Applied Mechanics and Engineering 307 (2016) 256–274.
[73] H. Al Akhras, T. Elguedj, A. Gravouil, M. Rochette, Towards an automatic isogeometric analysis suitable trivariate models generation—application to geometric parametric analysis, Computer Methods in Applied Mechanics and Engineering 316 (2017) 623–645.
[74] G. Xu, M. Li, B. Mourrain, T. Rabczuk, J. Xu, S. P. Bordas, Constructing iga-suitable planar parameterization from complex cad boundary by domain partition and global/local optimization, Computer Methods in Applied Mechanics and Engineering 328 (2018) 175–200.
[75] X. Wang, X. Qian, An optimization approach for constructing trivariate b-spline solids, Computer-Aided Design 46 (2014) 179–191.
[76] T. Elguedj, T. J. Hughes, Isogeometric analysis of nearly incompressible large strain plasticity, Computer Methods in Applied Mechanics and Engineering 268 (2014) 388–416.
[77] V. P. Nguyen, S. Bordas, Extended isogeometric analysis for strong and weak discontinuities, in: Isogeometric methods for numerical simulation, Springer, 2015, pp. 21–120.