



Politecnico di Torino

Corso di Laurea Magistrale in Automotive Engineering

Isogeometric analysis: LS-DYNA implementation and comparison with standard finite elements.

Candidato: Federico Ferrari

Relatori: Lorenzo Peroni, Martina Scapin

Anno accademico 2019-2020

RINGRAZIAMENTI

Prima di procedere con la trattazione, vorrei dedicare qualche riga a tutti coloro che mi sono stati vicini in questo percorso di crescita personale e professionale.

In primis, un ringraziamento speciale ai miei relatori, Lorenzo Peroni e Martina Scapin, che con la loro disponibilità mi hanno permesso di portare a conclusione questo percorso di studi, malgrado la situazione non propriamente favorevole.

Ringrazio infinitamente i miei familiari che mi hanno sempre sostenuto, appoggiando ogni mia decisione e riponendo fiducia in me.

Infine, un grazie per tutti gli amici, per aver aumentato in modo considerevole la qualità (e la quantità) degli anni di studi.

ABSTRACT

In the last few years, isogeometric analysis (IGA) has become the object of many scientific researches. It is a new numerical analysis method, based on the usage of the exact geometry representation in analysis environment instead of the common mesh discretization. In the NURBS based IGA, the linear Lagrange polynomials basis function, utilized in FEA, are substituted by non uniform rationals b-splines basis functions, the same used in CAD environment.

This new approach, in the future, will lead to a great saving of time and money, considering that the meshing phase can be computationally very expensive, hard to fully automate, error-prone, and it becomes exponentially more time consuming with the growth of the product complexity.

In this thesis, after a brief theoretical presentation, different types of NURBS based IGA are performed by means of the commercial software LS-DYNA, in order to assess the state of the art and the limits of the software with this method. Moreover, comparisons with FE analyses are made in order to understand the accuracy of the results and the cpu complexity of equivalent problems.

At the end, strong and weak points of the method are presented, based on the issues encountered and on the results produced by the analyses.

INDICE

RINGRAZIAMENTI.....	2
ABSTRACT.....	3
1. INTRODUCTION	1
1.1. Background	1
1.1.1 CAD and CAE	1
1.2. Isogeometric analysis	4
1.2.1 IGA with NURBS basis functions	5
1.3. Research objective	8
2. THEORETICAL BACKGROUND.....	9
2.1. Spline geometry	9
2.1.1 Introduction.....	9
2.2. Bézier curves	10
2.3. B-splines.....	12
2.3.1 Derivatives of B-spline basis functions	17
2.3.2 B-spline curves	19
2.3.3 B-spline surfaces.....	20
2.3.4 B-spline solids.....	22
2.4. Refinement techniques	22
2.4.1 Knot Insertion (h-refinement)	23
2.4.2 order elevation (p-refinement)	25
2.4.3 k-refinement.....	27
2.5. Non Uniform Rational B-Splines (NURBS).....	30

2.5.1 NURBS basis functions and derivatives	32
2.6. NURBS modelling softwares	34
3. ISOGEOMETRIC ANALYSIS WITH LS-DYNA	37
3.1. LS-DYNA	37
3.1.1 Pre-Post processors	39
3.2. IGA modelling	39
3.2.1 Shell elements	40
3.2.3 Available shell formulations	42
3.2.4 ELEMENT_SHELL_NURBS_PATCH KEYWORD	45
3.2.5 DEFINE_NURBS_CURVE keyword	50
3.2.6 Solid NURBS elements	54
3.3. Contacts and boundary conditions	57
3.3.1 Penalty based contacts	57
3.3.2 Tied NURB contacts	58
4. LS-PrePost Isogeometric analysis set up and capabilities	62
4.1. Shell NURBS elements set up.....	62
4.1.1 ELEMENT_SHELL_NURBS_PATCH creation	62
4.1.2 ELEMENT_SHELL_NURBS_PATCH refinement.....	66
4.1.3 Trimmed NURBS patch.....	67
4.2. Solid nurbs elements	68
4.2.1 NURBS solid elements creation with 3D NURBS editor.....	69
4.2.2 Refinement.....	71
4.3. Analysis performance.....	72
4.3.1 Model set up.....	72
4.3.2 Run.....	73
4.3.3 Post-processing	74

5.	IGA PERFORMANCE.....	79
5.1.	Multi-patch simple crashbox impact.....	79
5.1.1	Set up	79
5.1.2	Results.....	80
5.2.	Spot welded s-rail impact analysis	84
5.2.1	Set up	84
5.2.2	Results.....	85
5.2.3	Issues.....	88
5.3.	Analysis of complex models	89
5.4.	Implicit solver and eigenvalue analysis	93
5.4.1	Eigenvalue analysis of shell isogeometric elements.....	95
5.4.2	Issues.....	97
5.4.2	Eigenvalue analysis with solid NURBS elements	98
6.	Composite modelling.....	100
6.1.	Definition	100
6.1.1	Material definition	100
6.1.2	Section and composite layers definition	103
6.1.3	Through thickness definition of different materials	105
6.2.	Analysis.....	107
6.2.1	Impact on a carbon fiber plate	107
6.2.2	Three point bending of a sandwich panel	109
7.	Sheet metal forming.....	113
7.1.	State of the art	113
7.2.	Multi-stage simulations.....	116
8.	CONCLUSIONS	119
	LIST OF FIGURES	121

REFERENCES128

1. INTRODUCTION

1.1. Background

During the twentieth century, the increasing number and complexity of the engineering problems made it difficult to find the solutions by using hand calculations. At this point, engineers started to make use of computers, the most important factor behind the development of mankind over the last fifty years. Beginning in the 1960s, the design process of construction projects has been gradually digitized. With the availability of personal computers In the 1980s, the use of software tools became a common practice in engineering consultancies.

Nowadays all the industrial product development process is based on advanced computer calculations. The software packages used by engineers are generally organized in three main groups: Computer Aided Design (CAD), Computer Aided Engineering (CAE) and Computer Aided Manufacturing (CAM).

1.1.1 CAD and CAE

CAD technology is used, as its name implies, for design of structures and design process documentation. 3D models, Detailed engineering drawings, material information, dimensions and tolerances with specific conventions can be created by using CAD programs and such drawings are main input for the manufacturing process. It is widely recognized that modern CAD technology has its origins in the work of two French engineers: Pierre Bezier, from Renault, and Paul de Casteljau, from Citroen. Bezier [1, 2, 3] used Bernstein's polynomials [4] as the basis for his model of generating lines and surfaces, that he called Bezier curves. (de Casteljau did the same some years earlier, without however publishing his studies). the term spline was first introduced by Shoenberg [5], who studied them as interpolatory function, but his work hasn't been used in cad technologies until the 1960s [6]. In the 70's there have been a rapid development of these topics: Reisenfeld [7] and Vesprille [8] studied in their PhD thesis', respectively, the B-

splines and the Non-Uniform Rational B-splines (NURBS). NURBS made the usage of rational functions and exact representation of the conic sections possible which is not the case for B-Spline. Today, NURBS is in use by most of the commercial CAD software packages and data exchange standards due to its superior properties. Subsequently, many new techniques were introduced to improve the representation, In particular, the introduction of T-splines [9,10] in the CAD program in the early 2000s is noteworthy: these functions are an extension of the NURBS concept and are very efficient for what concerns local refinement.

On the other hand, CAE is used to conduct engineering analyses such as Structural Analysis, Computational Fluid Dynamics (CFD) and Multibody Dynamics (MBD). In CAE softwares, Engineering designs are evaluated in terms of their functions and the structures are analyzed under the working conditions with applied forces, pressures or temperatures and so on. For complex geometries and boundary conditions, it is not possible to solve the problem of a structure under working conditions in an analytical way. On this purpose, numerical methods have been developed and, nowadays, the most commonly used in structural problems is the Finite Element Method [FEM]. The origin of the FEM goes back to study of Richard Courant (1943) [11] where he proposed discretization of the whole domain into a set of finite triangular subregions in accordance with the philosophy of the finite element method. A few years later, in 1960, Dr. Ray Clough has used the term “finite elements” for the first time in his study [12]. At the same time, since digital computers were invented with capability of making hundreds of operations per second, first commercial FEA programs began to be developed.

The long-term use of these mathematical models, both for CAD and CAE, can be shown as a proof that analysis and design mathematical models worked well throughout years, even if different solution methods are being used in these two fields and this causes extra time consumption. Infact, the models created by using CAD software cannot be directly used by FEA technique, because, while the CAD

community uses geometry descriptions like e.g. NURBS, subdivision surfaces, T-splines or others, the FEA community generally uses linear Lagrange polynomials to approximate the geometry. So, one should make the design model suitable for analysis by transforming the data set, performing well-known method called as “meshing”. Although the geometric transformation can be easily achieved for many applications in solid mechanics, it constitutes a severe bottleneck for the analysis of complex geometries, that can be computationally very expensive, hard to fully automate, and often leads to error-prone meshes, which have to be manually improved by the user.

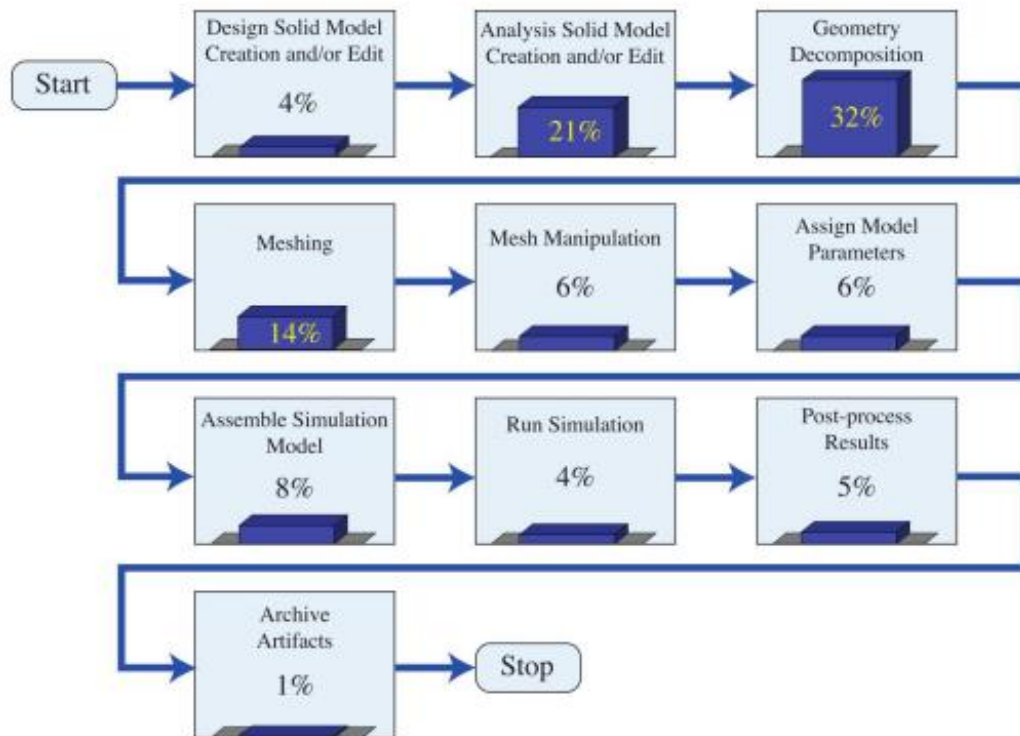


Figure 1-1: Estimation of the relative time costs of each component of the model generation and analysis process at Sandia National Laboratories. [13]

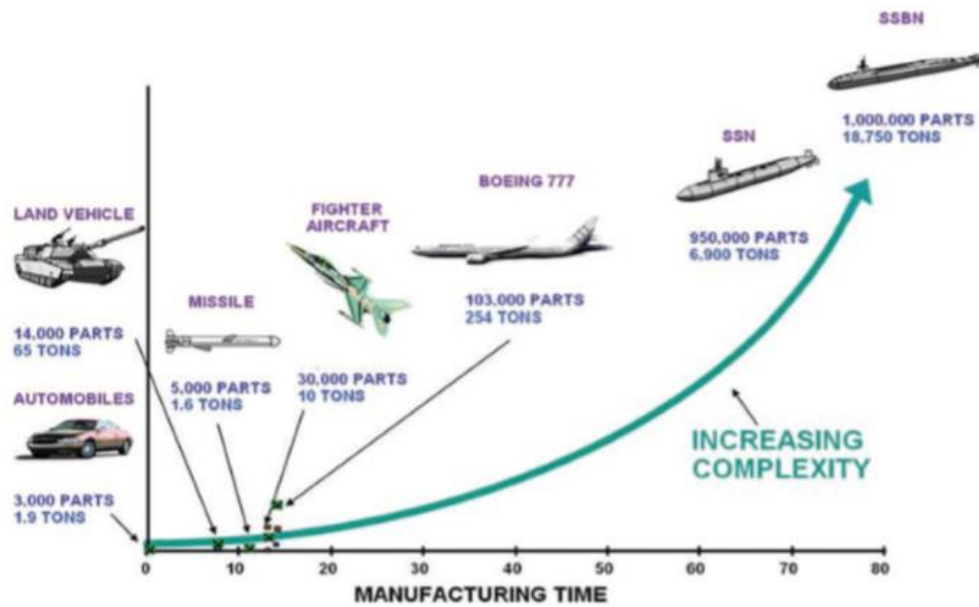


Figure 1-2: Manufacturing time in relation with the number of parts of a model [General Dynamics & Electric Boat Corporation]

The generation of a comprehensive structural model for different conceptual designs is time consuming. As can be seen from fig. 1-1, about 75% of overall engineering time is required to generate the final simulation model from the design input, moreover, the meshing phase becomes exponentially more time consuming with the growth of the product complexity (fig. 1-2).

1.2. Isogeometric analysis

This situation prompted the academy and industry to seek a new solution that could be used jointly for the two main disciplines, design and analysis. Isogeometric analysis emerged in accordance with these conditions. IGA is a recently born analysis method that combines Finite Element Analysis (FEA) and Computer Aided Design (CAD) by providing an appropriate algorithm for computerized solution. The main idea behind the emergence of isogeometric analysis is utilizing the same basis functions in both design and analysis [14]. It

focuses to use one geometric model that can be utilized for analysis directly or can be manipulated for analysis easily and automatically.

A comparison of meshing for standard FE and IGA is shown in Fig. 1-3. Where can be seen that the geometry representation based on linear lagrange polynomials will lead to a discretization error that can only be reduced to a tolerable value performing mesh refinement. On the other hand, IGA uses the exact geometry for Analysis, any necessary mesh refinements for enlarging the solution space won't change the geometry. [16]

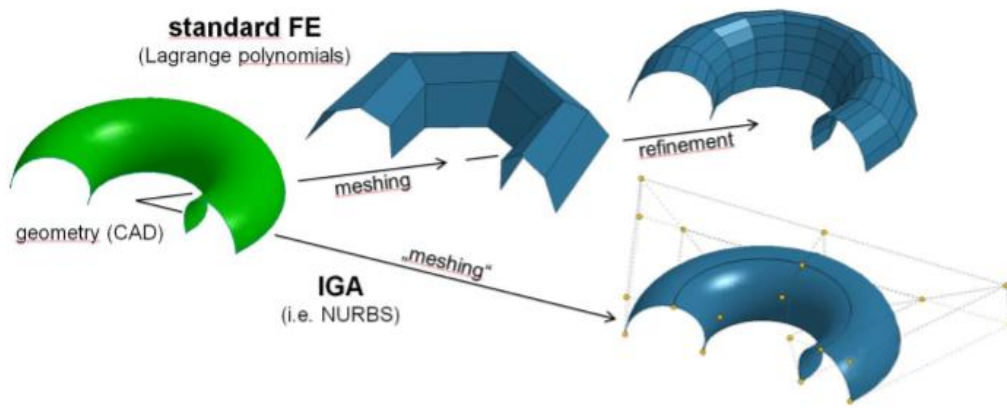


Figure 1-3: : Comparison of meshing for standard FE and IGA.[16]

It must be kept in mind that the term “isogeometric analysis” is not restricted to any special type of basis functions. It just indicates that the geometrical description that is used for FEA is the same than was used in CAD before. [15]

1.2.1 IGA with NURBS basis functions

In 2003 the research on isogeometric analysis started to focus on the question if finite element analysis could be done with non-uniform rational B-splines (NURBS), the most widely used geometry description in commercial CAD programs. The first promising results of these studies were presented in 2005 [14]. Since then, much research has been done on various topics of FEA (e.g. linear and non-linear static and dynamic analysis of thin-walled structures, fluid mechanics, fluid structureinteraction, shape and topology optimization, vibration analysis,

buckling and others) where many studies were performed using NURBS as basis functions.

This approach is also found in LS-DYNA, the software this thesis will be focusing on, where NURBS patch (shell and solid) elements can be created.

The main reasons behind the choice of NURBS is listed below:

- NURBS allows exact representation of geometries.
- It is successful in modeling free-form surfaces, conic sections, circular, cylindrical, spherical and ellipsoid shapes with great flexibility and precision.
- With the help of Cox de Boor formulation, efficient and stable algorithms for NURBS can be easily generated or already available algorithms can be found.
- NURBS enables users to easily apply geometry refinement without regeneration of geometry.
- NURBS has non-interpolatory nature and high continuity.

In the classical finite element method approach, the geometric approximation inherent in mesh can cause accuracy problems. Some of the structures as in the case of thin shells are very sensitive to geometric imperfections. Any deficiency in the representation of geometry may change the results tremendously. As can be seen in Figure 1.2, magnitude of allowable buckling load on the cylindrical shell decrease considerably with the introduced geometrical imperfections.

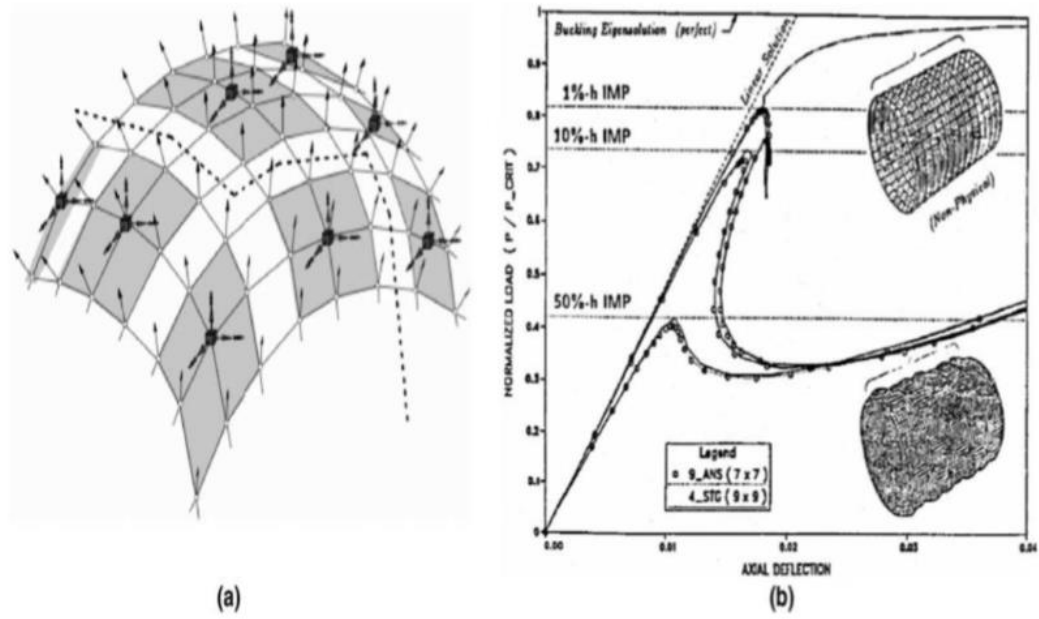


Figure 1-4: Thin shell structures exhibit significant imperfection sensitivity: (a) faceted geometry of typical finite element meshes introduces geometric imperfections and (b) buckling of cylindrical shell with random geometric imperfections [14]

On the other hand, since NURBS can define such cylindrical shapes free from imperfections, these problems can be analyzed with high accuracy. For example, in the field of contact mechanics, when finite elements are applied to geometry with curved surface, the result is a non smooth geometrical representation of interface surface which may lead to mesh interlocking, high jumps and oscillations in contact forces. To eliminate these issues, smoothening strategies are used in FEM, whereas, in IGA, these are not needed, thanks to the higher order continuity of the NURBS basis functions. Moreover, isogeometric analysis enables analysts to easily make mesh refinements without communicating and changing the geometry. On the other hand, for classical finite element method application, mesh refinement necessitates the regeneration of geometry and this means a lot of time consumption especially for assemblies with large number of parts.

1.3. Research objective

The main objective of this thesis is to conduct a study about isogeometric analysis (IGA) and its implementation in the commercial software LS-DYNA, to figure out which are the potentialities of this new method applied to structural mechanics problems.

IGA will be introduced with its theoretical background, Then there will be the evaluation of the state of the art of LS-DYNA for what concerns Isogeometric analyses capabilities, with an overview on the model set up and definition for different types of models and analyses.

Validation of IG analyses, will be then performed with the software, comparing them with analyses carried out using the finite element method.

2. THEORETICAL BACKGROUND

Upon giving the motivation for the use of isogeometric analysis in the introductory chapter 1, in this chapter spline geometry is introduced by initially discussing univariate Bézier, B-spline, and NURBS curves together with their respective underlying basis functions. Afterwards, the univariate curves are extended to multivariate formulations. With the definition of the different basis functions at hand, their incorporation in the finite element method is illustrated and the fundamental properties of the resultant isogeometric analysis concept are presented.

The information contained in the following sections serves as a basis for this work but does not provide a complete discussion on the individual topics. For this purpose, references to fundamental publications and monographs are given in the respective sections. The content provided here stems from studies presented in [17, 18, 19, 23, 24, 32, 33, 34], but due to its basic nature, is in general not cited explicitly.

2.1. Spline geometry

2.1.1 Introduction

As the geometry description is essential for isogeometric analyses, this section shall give a brief overview of the different formulations prior to incorporating them in the context of the finite element method. Since NURBS are best understood when explained by the steps of its evolution, Bézier curves and standard B-spline curves will be briefly addressed before turning the attention to NURBS curves. After discussing the one dimensional formulations, the extension to higher dimensions, i.e. to surface and volumes will be presented.

2.2. Bézier curves

The Bézier curves are parametric curves, used in computer graphics and related fields. Independently discovered by Pierre Bézier and Paul de Casteljaeu, it was a way to give a mathematical description to the car bodies design with the emerging methods of CAD and CAM. [25,26]

A Bézier curve is defined by a set of $p+1$ control points \mathbf{P} , where p is called its order ($p = 1$ for linear, 2 for quadratic, etc.). The curve is defined in the parameter domain $[0,1]$, and the first and last control points are always the end points of the curve; however, the intermediate control points (if any) generally do not lie on the curve.

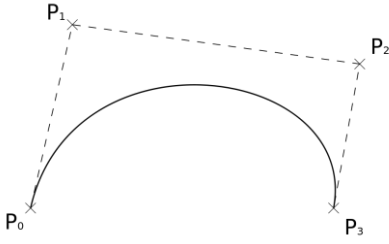


Figure 2-1: Example of cubic Bézier curve with its control polygon [wiki].

The dashed lines of the curve in fig 2-1 constitute the control polygon of that curve. Moving any of the control points will change the shape of the entire curve in an intuitive manner. This type of control established the curve's popularity in the world of computer aided design.

A Bézier curve as shown is defined as

$$C(\xi) = \sum_{i=0}^n N_p^{(i)}(\xi) \mathbf{P}^{(i)} \quad \forall \xi \in [0,1] \quad (2.2.1)$$

Where the i^{th} basis function $N_p^{(i)}$ is the Bernstein polynomial of degree p .

$$N_p^{(i)}(\xi) = \binom{p}{i} \xi^i (1 - \xi)^{p-i} \text{ with } 0^0 \equiv 1 \text{ and } \binom{p}{i} = \frac{p!}{i!(p-i)!} \quad (2.2.2)$$

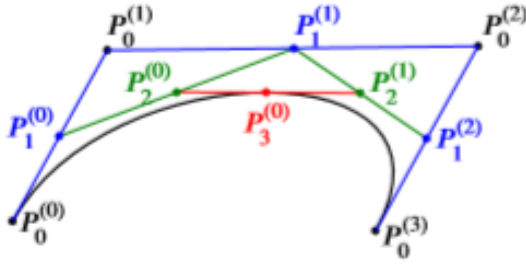


Figure 2-2: the evaluation of a point on that curve at $\xi = 0.5$ with the de Casteljau algorithm.

The geometric construction of a point on the curve with the help of the de Casteljau algorithm is depicted in fig. 2-2. The curve $C(\xi)$ is evaluated at $\xi = 0.5$ by consecutively interpolating the lines connecting the control points at 0.5, creating new lines connecting the interpolated points and starting over until at iteration p the point $C(0.5)$ is found.

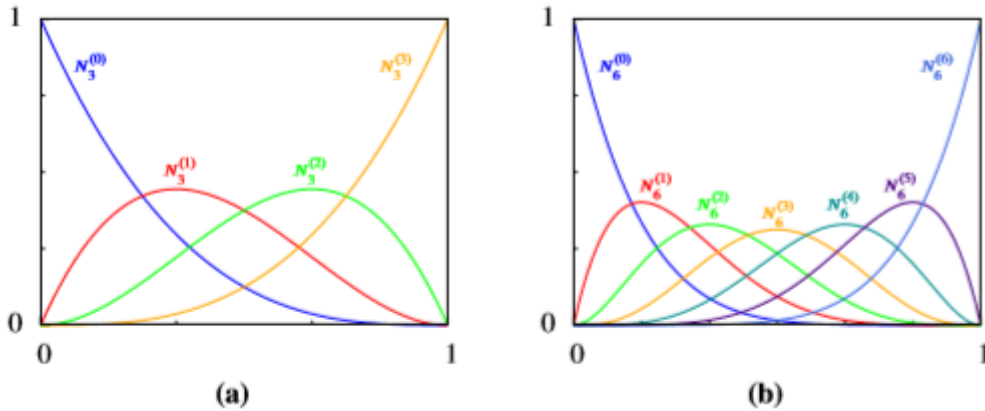


Figure 2-3: Bernstein polynomials of different degree plotted over the domain $[0, 1]$, i.e. the basis functions of a Bézier curve. (a) For the depicted curve in fig. 2-1 (a) with degree 3; (b) the functions of a curve with seven control points ($p = 6$) [24].

The basis functions of the curve in fig. 2-1 are plotted in fig. 2-3 (a), and for the reason of comparison, the respective functions of a Bézier curve with seven control points, i.e. a curve of polynomial degree 6, are shown in fig. 2-3(b). The functions are non-negative and fulfill the partition of unity property. It is to be noted, that all functions are non zero over the entire domain $]0, 1[$. This is the reason why, manipulating the position of a single control point influences the

shape of the entire curve, whatever the number of control points of the curve is. As none of the function values becomes one over $]0,1[$, the curve does not pass through any of the control points $P^{(1)}$ to $P^{(n-1)}$ and reversely, passes through the start and end point, where the respective basis functions are equal to one at $\xi = 0$ and $\xi = 1$.

2.3. B-splines

Owing to the use of the Bernstein polynomial as the basis functions of the Bézier curve, it is not possible to define a single curve of low degree with a higher number of control points, i.e. more control points than $p+1$. The basis also prohibits the application of local changes on a given single curve.

These short comings were overcome with B-spline curves that are built from piecewise polynomial functions, defined by knot vectors on non-overlapping connected intervals. Within these intervals B-splines are smooth, differentiable and continuous while at the boundaries of these intervals they are still continuous but not necessarily differentiable.

Instead of defining the curve over $\Omega = [0,1]$, the domain of a B-spline curve in parameter space is given as $\Omega = [\xi_{p+1}, \xi_{n+1}]$, where the variables ξ_{p+1} and ξ_{n+1} stem from a set of coordinates Ξ , called knot vector.

$$\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p}, \xi_{n+p+1}\} \quad (2.3.1)$$

where $\xi_i \in \mathbb{R}$ is the i^{th} knot, $i = 1, 2, \dots, n+p+1$, is the knot index, p is the polynomial order, and n is the number of basis functions used to constitute B-Spline curve, equal to the number of control points.

The coordinates in the knot vector, commonly referred to as knots, must not be decreasing, i.e:

$$\xi_{i+1} > \xi_i \quad (2.3.2)$$

Must hold for all entries.

Generally, knot values are normalized in the range between 0 and 1. The knots partition the parameter space into elements, usually referred as “knot spans”. Element boundaries in the physical space are simply the images of knot lines under the B-spline mapping.

Knot vectors can be classified as uniform or non-uniform and open or periodic knot vectors. If knot values in the knot vector are equally spaced in the parameter space such as [0 1 2 3 4] or [0 0,1 0,2 0,3 0,4], then knot vector is called as uniform. Otherwise it is named as non-uniform knot vector. A knot vector can be defined as open if its first and last knot values appear $p+1$ times. B-Spline basis that are constructed from open knot vectors interpolate to the control points at the ends of the parameter space interval, $[\xi_1, \xi_{n+p+1}]$, for one dimension. On the other hand, for multiple dimensions, they interpolate at the corners of patches. However, in general they are not interpolatory at interior knots. This is a distinctive property between knots in isogeometric analysis and nodes in finite element analysis.

In the parametric space more than one knot can be located at the same coordinate and thus, knot values can repeat in knot vector. The number of repetitive knots is called as knot multiplicity and this case has essential effects on the properties of basis functions. Knot repetition can decrease the continuity of the basis function to C^{p-m} where m is the number of multiplication. When the number of multiplication is equal to polynomial degree p , the basis will be C^0 continuous at the multiplied knot value [17]. This makes the basis function non-differentiable at that knot. This property makes it possible to create sharp corners in the spline curve by controlling the continuity to the associated basis functions.

These coordinates are used for the evaluation of the B-spline basis functions $N_p^{(i)}$ with the Cox-de Boor recursion formula, so:

$$N_0^{(i)} = \begin{cases} 1 & \text{for } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.3.3)$$

And for $p > 0$:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (2.3.4)$$

While working with open knot vectors or repeated knots, it is very crucial to take into account that one might encounter with zero denominator. This problem was solved by defining the result of such equations equal to zero. [16, 17]

During the calculation of basis functions, due to the recursive nature of formulation, results of higher order polynomials require the results of lower orders. This dependency is shown in Figure 2-4.

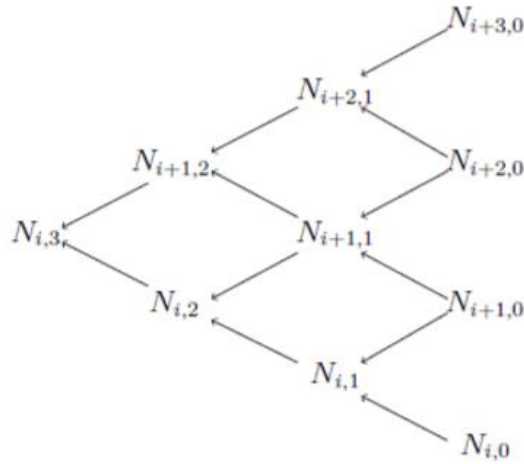


Figure 2-4: Dependencies between results of basis functions for computing a cubic basis function [17]

For constant and linear basis functions with a uniform knot vector $\Xi = \{0, 1, 2, 3, 4, 5\}$ the results are represented in the figure 2-5. Looking at the figure

shown, it can be said dynamic programming code is necessary to improve the efficiency of this recursive formula. Otherwise, the same values will be calculated several times. It should be noted that for $p = 0$ and $p = 1$, B-Spline basis functions have the same values as constant and linear shape functions of classical finite element method. However, by increasing the order, B-Spline basis functions differentiate from their finite element counterparts. This difference can be observed in figure 2-6, where the graphs of quadratic B-Spline basis functions and quadratic finite element shape functions are drawn. Quadratic B-spline basis functions are exactly same but shifted relative to each other with varying knot values. As we continue to higher-order basis functions this “homogeneous” pattern continues for the B-spline basis functions. On the other hand, quadratic finite element shape function differs according to the corresponding node position. This is a distinguishing feature between B-Spline basis and FEM shape functions that makes IGA superior to FEA.

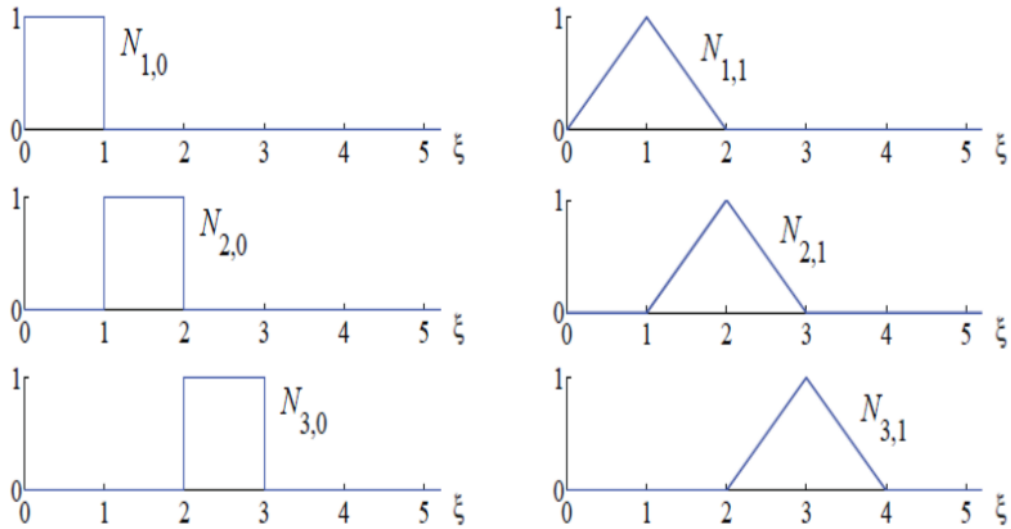


Figure 2-5: Basis functions for order 0 and 1 for uniform knot vector = $\{0,1,2,3,4,5\}$ [21]

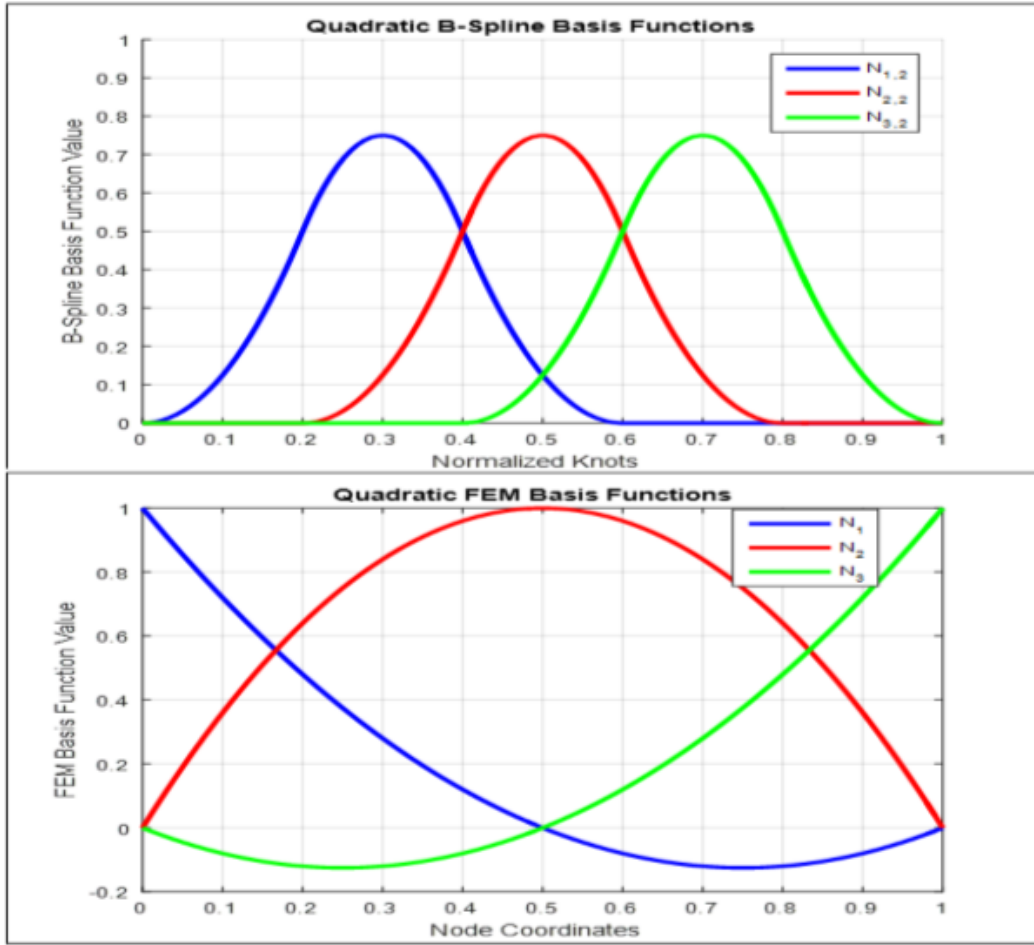


Figure 2-6: Comparison of quadratic finite element shape functions and B-spline basis functions [23].

In addition to mentioned homogeneity, B-Spline basis functions have several important properties as explained below [23]:

- B-Spline basis functions constitute a partition of unity $\sum_{i=1}^n N_{i,p}(\xi) = 1$
- Each basis function is non-negative over the entire domain $N_{i,p}(\xi) \geq 0, \forall \xi$
- B-Spline basis functions are linearly independent $\sum_{i=1}^n \alpha_i N_i(\xi) = 0$ only for $\alpha_i = 0, i = 1, 2, \dots, n$.
- The support of a B-Spline basis function of order p is $p+1$. $N_{i,p}$ is non-zero over $[\xi_i, \xi_{i+p+1}]$.
- Basis functions of order p have $p-m_i$ continuous derivatives across knot ξ_i where m_i is the multiplicity of knot ξ_i .
- Scaling or translating the knot vector does not alter the basis functions.

- B-Spline basis are generally only approximate to control points and not interpolate. Therefore, they do not satisfy the Kronecker delta property $N_{i,p}(\xi) \neq \delta_{ij}$. Only in the case $m_i = p$, then $N_{i,p}(\xi) = 1$.

Non-uniform knot vectors should be preferred to obtain richer behavior for basis functions rather than uniform knot vectors. An example created in [23] by using an open non-uniform knot vector $\Xi = [0 \ 0 \ 0 \ 0.2 \ 0.4 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1 \ 1]$ is shown in figure 2-7. Basis functions are interpolatory at the end points and additionally at the repeated knots where multiplicity is equal to polynomial degree p . At this repeated knot, only C^0 continuity is attained. Elsewhere the functions have C^1 continuity. When the multiplicity is $p+1$, the basis becomes discontinuous and the patch boundary is formed.

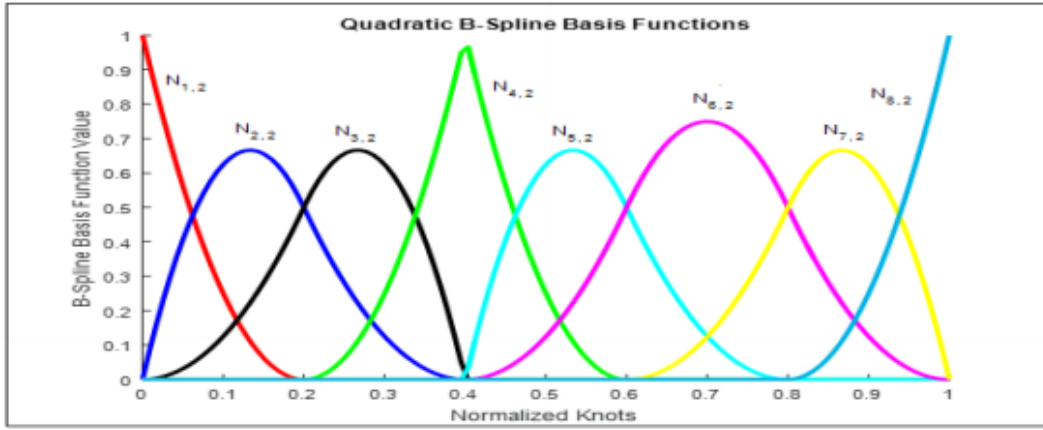


Figure 2-7: Quadratic basis functions drawn for non-uniform open knot vector $\Xi = [0 \ 0 \ 0 \ 0.2 \ 0.4 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1 \ 1]$. [23]

2.3.1 Derivatives of B-spline basis functions

By deriving the B-spline basis functions we obtain:

$$\begin{aligned}
 \frac{d}{d\xi} N_{i,p}(\xi) &= N'_{i,p}(\xi) \\
 &= \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) \\
 &\quad - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi)
 \end{aligned} \tag{2.3.5}$$

This formula defines the derivative of a basis function as a linear combination of 2 basis functions of degree $p-1$, this is due to the recursive nature of these functions.

For the higher order derivatives above formula can be generalized by simply taking the derivatives of each side to get:

$$N_{i,p}^{(k)}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} \left(\frac{d^{k-1}}{d^{k-1}\xi} N_{i,p-1}(\xi) \right) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} \left(\frac{d^{k-1}}{d^{k-1}\xi} N_{i+1,p-1}(\xi) \right) \quad (2.3.6)$$

Expanding (2.3.5) by means of (2.3.6) results in an expression purely in terms of lower order basis functions, $N_{i,p-k}, \dots, N_{i+k,p-k}$, are given below;

$$N_{i,p}^{(k)}(\xi) = \frac{p!}{(p-k)!} \sum_{j=0}^k \alpha_{k,j} N_{i+j,p-k}(\xi) \quad (2.3.7)$$

With:

$$\alpha_{0,0} = 1 \quad (2.3.8)$$

$$\alpha_{k,0} = \frac{\alpha_{k-1,0}}{\xi_{i+p-k+1} - \xi_i} \quad (2.3.9)$$

$$\alpha_{k,j} = \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+p+j-k+1} - \xi_{i+j}} \text{ where } j = 1, \dots, k-1, \quad (2.3.10)$$

$$\alpha_{k,k} = \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}} \quad (2.3.11)$$

2.3.2 B-spline curves

B-Spline curves in \mathbb{R}^d are created by taking linear combination of multiplication of B-Spline basis functions with coefficients called as “control points”. When the control points are linearly interpolated, the resultant polygon is referred as “control polygon”. Given n basis functions $N_{i,p}$ with specific order p , where $i = 1, 2, \dots, n$, and corresponding control points $\mathbf{B}_i \in \mathbb{R}^d$, $i = 1, 2, \dots, n$, then B-Spline curve is defined by:

$$\mathcal{C}(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i \quad (2.3.12)$$

The resulting B-Spline curve does not necessarily interpolate the control points. Nevertheless, when the interpolation is desired, by using the properties stated in the previous part, curve can interpolate to specific control points.

A B-Spline curve example is shown in figure 2-8 which is constructed by using quadratic basis functions given in figure 2-7 created from specified knot vector $\Xi = [0 \ 0 \ 0 \ 0.2 \ 0.4 \ 0.4 \ 0.6 \ 0.8 \ 1 \ 1 \ 1]$. The control points and control polygon is also seen in the figure.

Since the curve is built from an open knot vector, it interpolates to first and last control points. Moreover, curve is also interpolatory at the fourth control point due to the repetition of knot $\xi = 0.4$ as much as the polynomial order.

B-Spline curves carry many properties of their basis functions. For instance, in the absence of repeated knots or control points, B-Spline curves of degree p have $p-1$ continuous derivatives. In the light of this information, sample curve is $C^{p-1} = C^1$ continuous everywhere except at the location of the repeated knot, $\xi = 0.4$, where it is $C^{p-2} = C^0$ continuous.

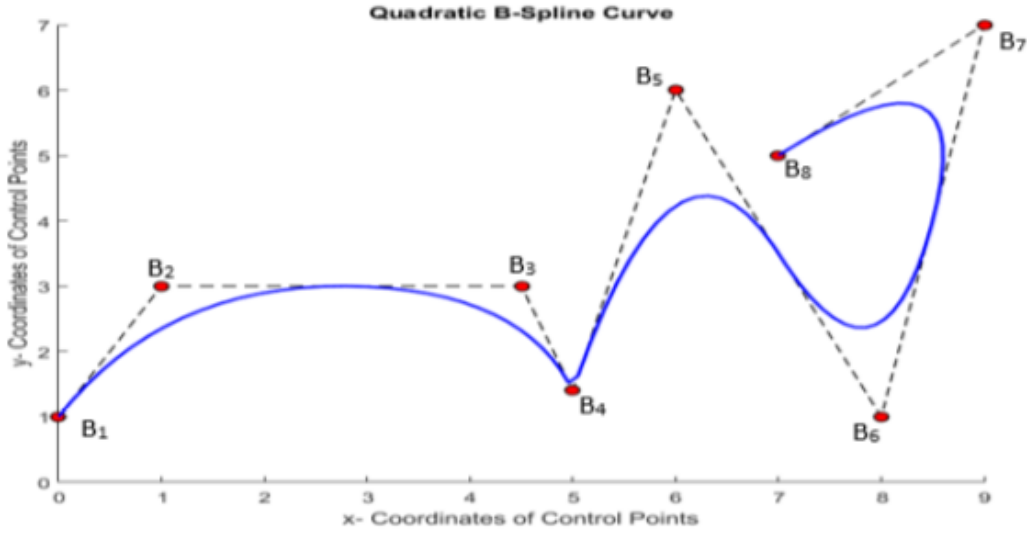


Figure 2-8: A Quadratic B-Spline curve example [23].

Another property the curves inherit from their basis is “locality”. Due to the compact support of the B-spline basis functions, moving a single control point can affect the geometry of curve by affecting $p + 1$ elements of the curve.

2.3.3 B-spline surfaces

in order to obtain a B-spline surface, it is necessary to take a bidirectional net of control points, $\{\mathbf{B}_{i,j}\}$, $i = 1, 2, \dots, n$, and $j = 1, 2, \dots, m$ and two knot vectors $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, $\mathcal{H} = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$ where p and q are polynomial orders.

Calculation is done by the combination of the tensor products of corresponding univariate B-spline functions defined as follows:

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) \mathbf{B}_{i,j} \quad (2.3.13)$$

An example for the B-Spline surface is considered by using following knot vectors $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ of degree $q = 2$ and $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ of degree $p = 2$. Basis functions for these knot vectors are given in figure 2-9 and the created surface is shown in figure 2-10.

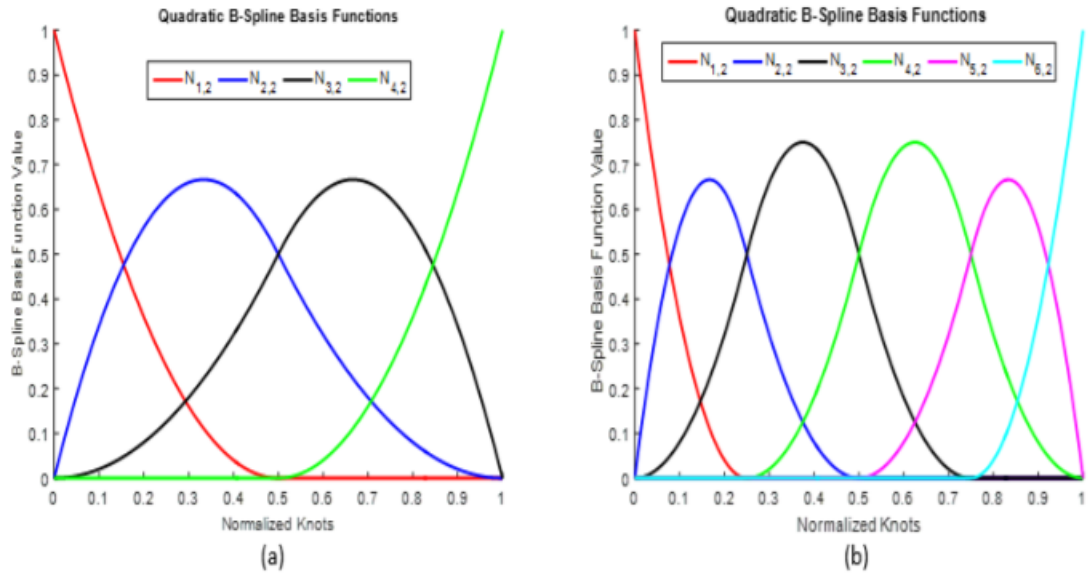


Figure 2-9: Basis functions of knot vectors (a) $\mathcal{E} = \{0, 0, 0, 0.5, 1, 1, 1\}$ and (b) $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ [23].

Coordinates of the utilized control net on the surface is given in figure 2-11 [22].

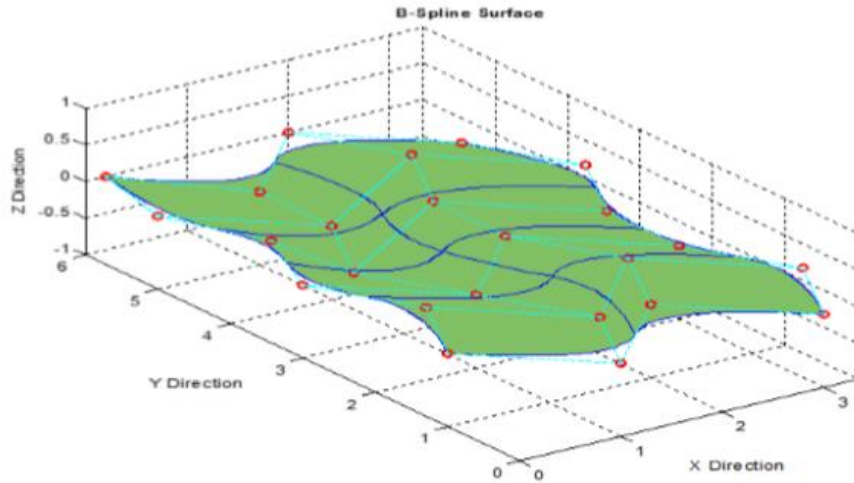


Figure 2-10: An Example B-Spline surface [22].

(i,j)	1	2	3	4	5	6
1	(0,1)	(0.5,2)	(0,3)	(0.4,4)	(0,5)	(0.2,6)
2	(1,0)	(1.5,1)	(1,2)	(0.5,3)	(1,4)	(1,5)
3	(2,1)	(2.5,2)	(2,3)	(2,4)	(2.5,5)	(2,6)
4	(3,0)	(3.5,1)	(3,2)	(3,3)	(3.5,4)	(3,5)

Figure 2-11: control net $B_{i,j}$ [22].

2.3.4 B-spline solids

B-spline solids are produced by using control lattice, $\{B_{i,j,k}\}$, $i = 1,2,...,n$; $j = 1,2,...,m$ and $k = 1,2,...,l$ knot vectors are $\Xi = \{ \xi_1, \xi_2, ..., \xi_{n+p+1} \}$, $\mathcal{H} = \{ \eta_1, \eta_2, ..., \eta_{m+q+1} \}$ and $\mathcal{L} = \{ \zeta_1, \zeta_2, ..., \zeta_{l+r+1} \}$ where p , q and r are polynomial orders.

Construction formulation for the B-Spline solids is given by:

$$S(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) B_{i,j,k} \quad (2.3.14)$$

2.4. Refinement techniques

In classical finite element method approach, in order to get more accurate results, basis enriched by using two common refinement techniques: h-refinement and p-refinement. The former method, h-refinement, increases the number of elements by decreasing the element size to get higher resolutions. On the other hand, the latter one, p-refinement, increases the polynomial degree of basis functions.

These refinement techniques in isogeometric analysis are named as knot insertion which is similar to h-refinement and order elevation similar to p-refinement.

Contrary to the finite element methods, the geometry remains unchanged under each refinement and the continuity across each element is more controllable in isogeometric analysis. Moreover, IGA has one more refinement technique superior to FEM. It is the combination of order elevation and knot insertion respectively and called as “k-refinement” that brings many benefits to analysis world. Details of these refinement techniques are given in the succeeding parts.

2.4.1 Knot Insertion (h-refinement)

In isogeometric analysis first technique used to enhance basis is knot insertion which is analogous to h-refinement in FEA. During the application of knot insertion, new knots are inserted into already existing knot vector without changing the geometry.

For a given knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, a new knot vector can be obtained by inserting additional knots as $\bar{\Xi} = \{\bar{\xi}_1, \bar{\xi}_2, \dots, \bar{\xi}_{n+m+p+1}\}$ such that $\bar{\xi}_1 = \xi_1$ and $\bar{\xi}_{n+m+p+1} = \xi_{n+p+1}$ and thus, $\Xi \subset \bar{\Xi}$. New $n+m$ basis functions should be calculated by using cox de boor algorithm. The novel $n+m$ control points, $\bar{\mathcal{B}} = \{\bar{\mathbf{B}}_1, \bar{\mathbf{B}}_2, \dots, \bar{\mathbf{B}}_{n+m}\}T$, are generated from linear combinations of the original control points, $\mathcal{B} = \{\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n\}T$, as defined by,

$$\bar{\mathbf{B}}_i = \alpha_i \mathbf{B}_i + (1 - \alpha_i) \mathbf{B}_{i-1} \quad (2.4.1)$$

where,

$$\alpha_i = \begin{cases} 1 & \text{if } 1 \leq i \leq k-p \\ \frac{\bar{\xi}_i - \xi_i}{\xi_{i+p} - \xi_i} & \text{if } k-p+1 \leq i \leq k \\ 0 & \text{if } k+1 \leq i \leq n+p+2 \end{cases} \quad (2.4.2)$$

Insertion of the already existing knot value causes a repetition and decreases the continuity of the basis functions. In order to preserve the continuity, equations 2.3.15 and 2.3.16 are developed for the choice of proper control points.

An example of knot insertion procedure for a simple, one-element, quadratic B-spline curve is given in the figure 2-12. A new knot is inserted at $\bar{\xi} = 0.5$ to the existing knot vector $\Xi = \{0,0,0,1,1,1\}$, which is used to create the original curve. The newly created curve is geometrically and parametrically identical to the original curve. However, control points have been changed, the mesh has partitioned, and the basis functions have been enriched. In the new case, the number of control points, elements and basis functions, all increased by one. This process may be repeated to enrich the solution space by adding more basis functions of the same order until the desired sensitivity is reached.

Knot insertion refinement method is similar to the h-refinement technique of finite element method. As can be understood from the above mentioned procedure, knot insertion creates new knot spans i.e. new elements in the knot vector. Similarly, in finite element method, h-refinement increases the element number, creates a finer mesh of the same type of element to improve the results. However, IGA and FEA differ in the number of new basis functions and in the continuity of the basis across the novel element boundaries. To perfectly replicate h-refinement, one would need to insert each of the new knot values p times so that the functions will be C_0 continuity.

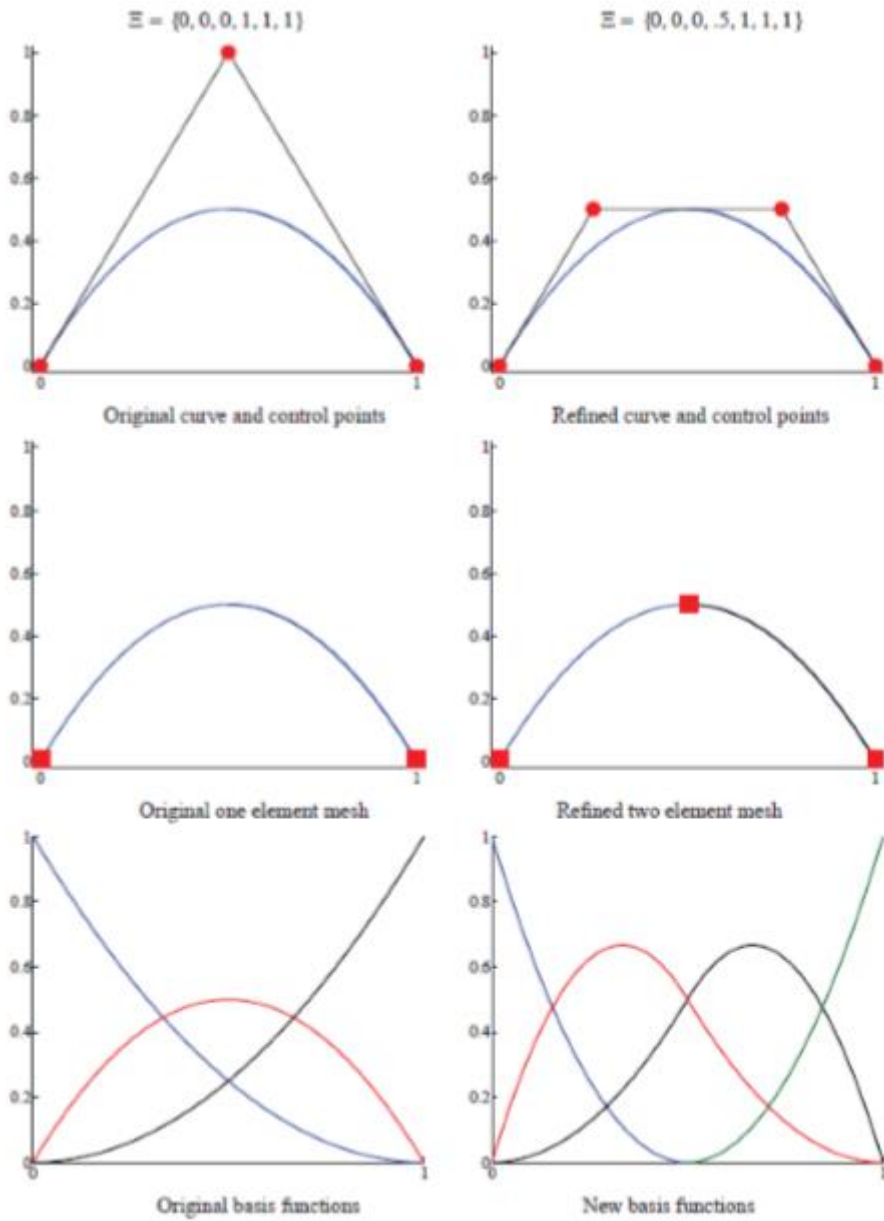


Figure 2-12: Knot insertion refinement technique [18].

2.4.2 order elevation (p-refinement)

Another basis improvement method which B-Spline theory enables the users is order elevation also called as degree elevation since the degree and order terms are used interchangeably in B-Spline theory. In this technique, polynomial order of the basis functions is increased. Since the basis functions have $p-m_i$ continuous derivatives across element boundaries, if the continuity is desired to be preserved,

it is obvious that when the order p is increased, multiplicity m must also be increased by the same amount of degree. Therefore, during order elevation process, the multiplicity of each knot value is increased. As in the case of knot insertion, geometry and parameterization remains unchanged.

Order elevation begins by replicating existing knots by the same amount as the increase in polynomial order. Thereafter, the order of polynomial is increased. Several efficient algorithms for the application of order elevation procedure can be found in Piegl and Tiller, 1997 [17].

An example for the order elevation procedure from quadratic to cubic order is represented in Figure 2-18. The original control points, mesh, and quadratic basis functions are shown on the left. Each knot value in knot vector has been increased by one but no new knot values were added. For this example, number of control points and basis functions increased from 8 to 13. The new control points calculation procedure is again given in Piegl and Tiller, 1997.

Although the locations of the control points change, the order elevated curve is geometrically and parametrically identical to the original curve. Additionally, multiplicities of the knots have been increased but the element number is preserved.

Order elevation process is analogous to p-refinement technique in finite element analysis. Both of the strategies increases the order of basis functions. The most critical distinction between these two is that p-refinement always begins with a basis that is C^0 everywhere, while order elevation is compatible with any combination of continuities.

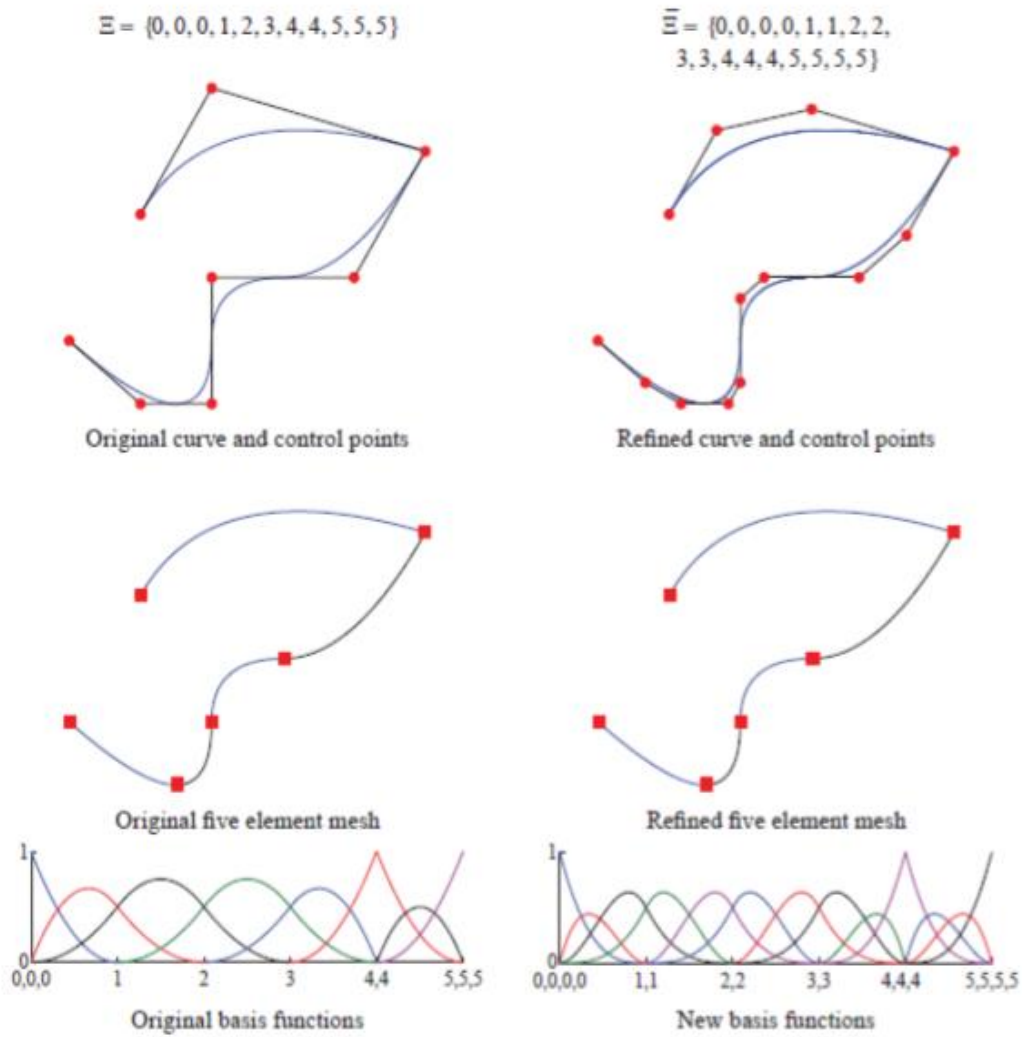


Figure 2-13: Order elevation refinement technique [18]

2.4.3 k-refinement

As mentioned in the previous parts, when a new knot values with multiplicities equal to one are inserted, functions across the boundaries will have C^{p-1} continuity. It is possible to lower the continuity by increasing the multiplicity as well. This shows knot insertion is a more flexible process than simple h-refinement. Likewise, order elevation technique is also more flexible than p-refinement technique. The stated flexibilities of knot insertion and order elevation techniques force us to develop another refinement technique which is unique in the field.

In a curve of order p , if a unique knot value, ξ , is inserted between two distinct knots, the number of continuous derivatives of the basis functions at ξ is $p-1$. After knot insertion, even though the order is elevated, the basis still has $p-1$ continuous derivatives at ξ . However, if we change the sequence i.e. if the order of the original curve elevated to order q first and then a unique knot value is inserted, the basis will have $q-1$ continuous derivatives at ξ . Thus an alternative order elevation method which has significant advantages over standard order elevation emerges. This procedure is called k-refinement. There is no analogous refinement technique in finite element method similar with k-refinement. The concept of k-refinement is important and potentially a superior approach to high-precision analysis than p-refinement. In traditional p-refinement there is an inhomogeneous structure to arrays due to the different basis functions associated with surface, edge, vertex and interior nodes. In k-refinement, there is a homogeneous structure within patches and growth in the number of control variables is limited.

In order to make it more clear, two different sequences of refinement processes are compared with an example in Figure 2-14. Initial domain consists of one element and $p+1$ basis functions. On the left side of the figure, firstly, knot is inserted until getting $n-p$ elements and n basis functions and then order is elevated. In this process, to maintain the continuity at the $p-1$ level each distinct knot value is replicated and the total number of basis functions is increased by $2n-p$. After a total of r order elevations of this type, we have $(r+1)(n)-(r)(p)$ basis functions, where p is still the order of our original basis functions. On the right side of figure, beginning with the same element domain this time order elevation is applied primarily, and then knot insertion is proceeded which is suitable to k-refinement procedure. In this case for each order elevated r times, total number of basis functions increases by only one for each refinement. Then domain can be h-refined until having $n-p$ elements. The final number of basis functions is $n+r$, each having $r + p - 1$ continuity. This amounts to an enormous savings in the number of basis functions as $n + r$ is considerably smaller than $(r+1)(n)-(r)(p)$. Moreover, this technique enables the arrangement of continuity of basis contrast to p-refinement.

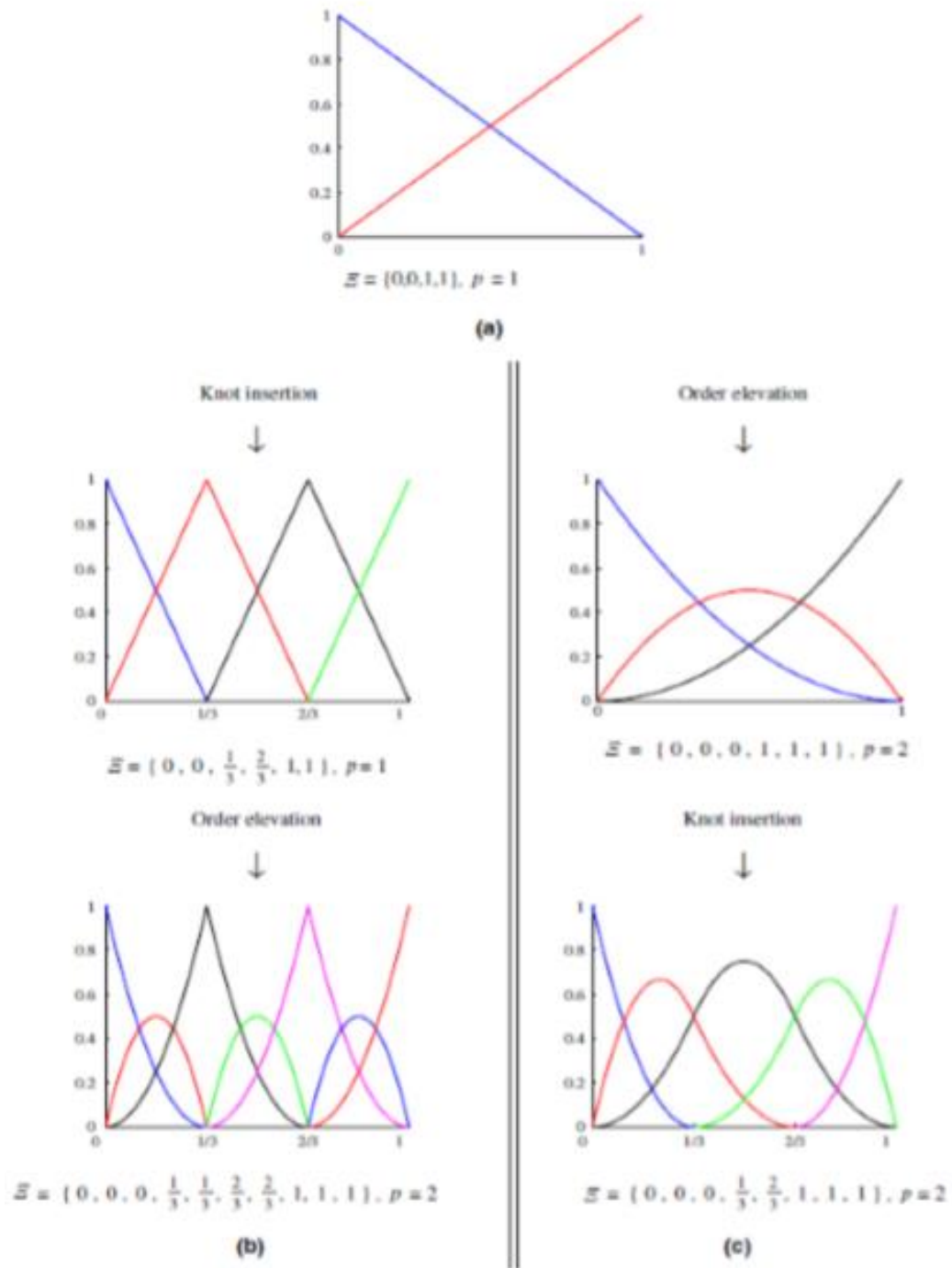


Figure 2-14: k-refinement sequence comparison (a) Base case of one linear element. (b) Classic p-refinement approach: knot insertion followed by order elevation results in seven piecewise quadratic basis functions that are $C0$ at internal knots (c) New k-refinement approach: order elevation followed by knot insertion results in five piecewise quadratic basis functions that are $C1$ at internal knots. [14]

2.5. Non Uniform Rational B-Splines (NURBS)

Although the B-Splines are convenient for free-form modeling and provide some advantages in geometry definition which were mentioned in the previous sections, they have still deficiencies in exact representation of some simple shapes such as circles and ellipsoids. In order to overcome this lack of ability, NURBS, a superset of B-Splines with its rational nature, is preferred. Today, NURBS is accepted as a de facto standard in CAD technology. Therefore, this section was devoted to discussion of NURBS concept and aims to show how they are constructed, what their advantages are and what separates them from B-splines.

As its name implies, NURBS are piecewise rational polynomials built from B-Splines and inherit all the favorable properties of them. The rational term refers to the fact that NURBS are a combination of B-splines basis functions multiplied by a weighting factor. If all the weights are equal to one, then NURBS will be equal to B-splines. On the other hand, non-uniform term is used to define non-uniform knot vector. Therefore, in addition to the polynomial degree, knot vector values and multiplicity parameters, one more parameter weight is introduced to obtain more flexible design with desired properties.

NURBS are constructed in \mathbb{R}^d by the projective transformation of B-Splines defined in \mathbb{R}^{d+1} . To illustrate, a circle in \mathbb{R}^2 constructed by the projective transformation of a piecewise quadratic B-spline defined using homogenous coordinates in \mathbb{R}^3 is shown in Figure 2-15.

In this figure, $Cw(\xi)$ is a B-spline curve in \mathbb{R}^3 which is created by $\{Biw\}$ set of control points. These control points are defined utilizing homogenous coordinates. Terminologically, this curve is called as “projective curve” and its associated control points are called as “projective control points”, Biw , while the terms “curve” and “control points” are used to describe NURBS curve $C(\xi)$ and its control points Bi respectively.

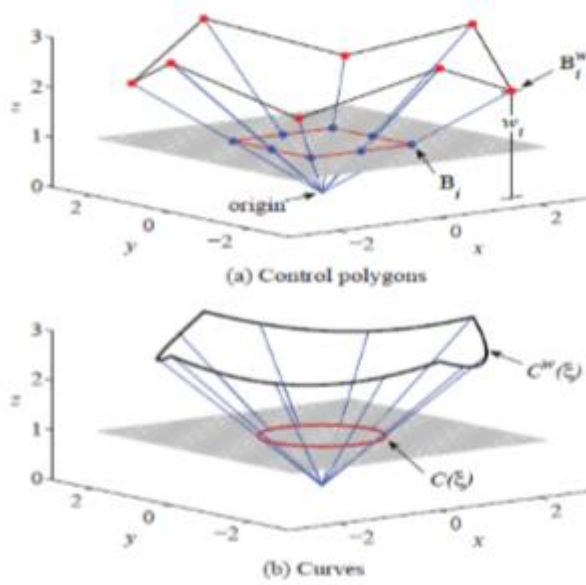


Figure 2-15: An Example of projective transformation of (a) Control points (b) Curves [14]

The projected control points for the NURBS curve are obtained by the following relations:

$$(B_i)_j = \frac{(B_i^w)_j}{w_i} \quad j = 1, \dots, d \quad (2.5.1)$$

Where,

$$w_i = (B_i^w)_{d+1} \quad (2.5.2)$$

Here, $(B_i)_j$ is the j th component of the vector B_i and w_i is the i th weight. In \mathbb{R}^{d+1} , the weights correspond to the $(d + 1)$ th component of the homogenous coordinates of B-spline curve. For example, in Figure 2-15, weights are taken as z -components of projective curves. Dividing the B-Spline control point by its corresponding weight is thus named as a projective transformation. The same transformations need to be exploited on every point on the curve by the definition of weighting function:

$$W(\xi) = \sum_{i=1}^n N_{i,p}(\xi) w_i \quad (2.5.3)$$

Now the NURBS curve can be defined as

$$(C(\xi))_j = \frac{(C^w(\xi))_j}{W(\xi)} \quad j = 1, \dots, d \quad (2.5.4)$$

The curve $C(\xi)$ is a piecewise rational function since each element of it is found by division of $C^w(\xi)$ to $W(\xi)$ which are both piecewise polynomial functions. Since this projective transformation seems intimidating, it is rarely used in practice. The main reason behind the explanation of projective transformation is to understand the underlying nature of NURBS and recognize that everything that have been discussed thus far for B-splines still holds true for NURBS.

2.5.1 NURBS basis functions and derivatives

In order to define the construction and manipulation of NURBS geometries it is necessary to introduce a basis function as in the case of B-Splines. NURBS basis function can be defined as follows:

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} = \frac{N_{i,p}(\xi)w_i}{\sum_{i=1}^n N_{i,p}(\xi)w_i} \quad (2.5.5)$$

Thereafter, NURBS curve defined by:

$$C(\xi) = \sum_{i=1}^n R_i^p(\xi) \mathbf{B}_i \quad (2.5.6)$$

One should note that, the weighting function in equation 2.5.3 is developed for the projection of B-Spline curve from \mathbb{R}^{d+1} into \mathbb{R}^d . Since it is embedded into basis function definition, we can built geometries and meshes in \mathbb{R}^d without regarding the projective geometry behind the scenes. For this reason, equation 2.5.6 is generally preferred to Eqn. 3.5.4 due to the usage of practical basis function although they are equivalent.

Rational basis functions are also defined analogously for the generation of rational surfaces and solids in Eqn. 2.5.7 and 2.5.8 respectively as follows

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}} \quad (2.5.7)$$

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}} \quad (2.5.8)$$

Related NURBS surfaces and solids are defined respectively by

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^{p,q}(\xi, \eta) \mathbf{B}_{i,j} \quad (2.5.9)$$

$$V(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \mathbf{B}_{i,j,k} \quad (2.5.10)$$

The derivatives of NURBS basis functions get by using quotient rule and their non-rational similitudes

$$\frac{d}{d\xi} R_i^p(\xi) = w_i \frac{W(\xi)N'_{i,p}(\xi) - W'(\xi)N_{i,p}(\xi)}{(W(\xi))^2} \quad (2.5.10)$$

Finally, an example of a NURBS surface represents a torus geometry which is difficult to create by using B-Splines is given in figure 2-16.

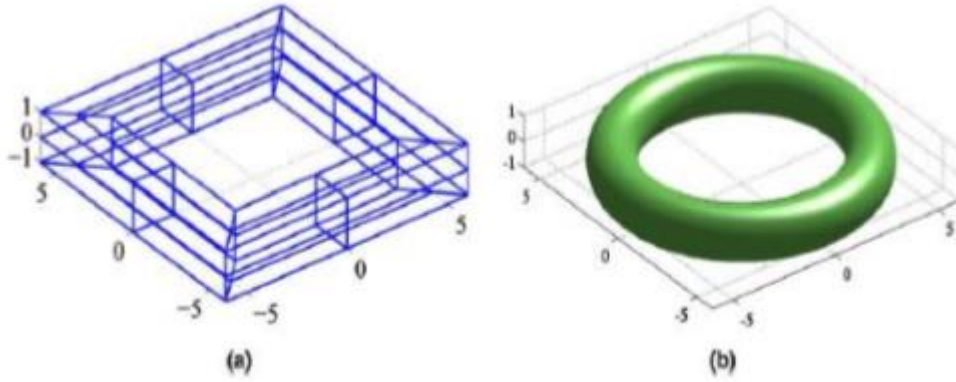


Figure 2-16: An Example for NURBS surface (a) Control net for toroidal surface (b) Toroidal surface [14]

2.6. NURBS modelling softwares

The high standards that CAD have reached nowadays allow the maximum freedom of choice to designers and engineers, with a good compatibility of NURBS solid and surfaces with almost all the softwares.

Some of them are powerful and/or easy to use. The most famous free form surface modeler, that utilizes NURBS mathematical model, is Rhinoceros 3D.

Rhinoceros is used in processes of computer-aided design (CAD), computer-aided manufacturing (CAM), rapid prototyping, 3D printing and reverse engineering in industries including architecture, industrial design (e.g. automotive design, watercraft design), product design (e.g. jewelry design) as well as for multimedia and graphic design.

The Rhinoceros file format (.3DM) is useful for the exchange of NURBS geometry. The Rhino developers started the open NURBS Initiative to provide computer graphics software developers the tools to accurately transfer 3-D geometry between applications.

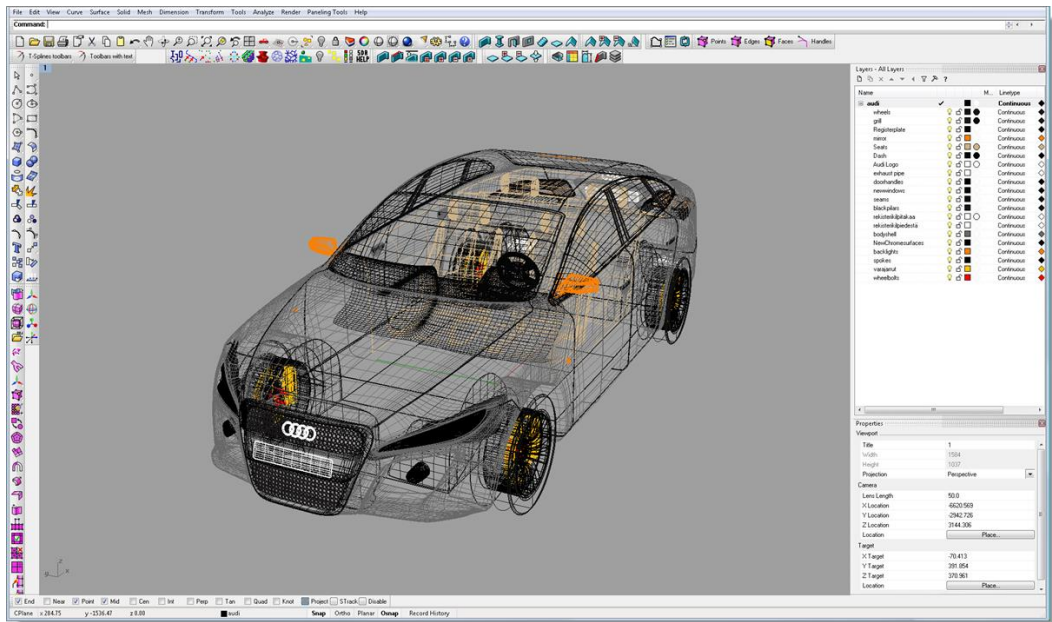


Figure 2-17: Rhinoceros 3D environment.

To make some other mentions, two powerful parametric CAD softwares used for engineering are, definitely, NX (siemens) and CATIA (Dassault systèmes). Their huge capabilities range over all the types of mechanical modelling, NURBS included.

NX, formerly known as "unigraphics", is an advanced high-end CAD/CAM/CAE, which has been owned since 2007 by Siemens PLM Software. It is spreading all around the world and it is used in many famous companies (Ducati, FCA and Beretta armi are some of the instances).

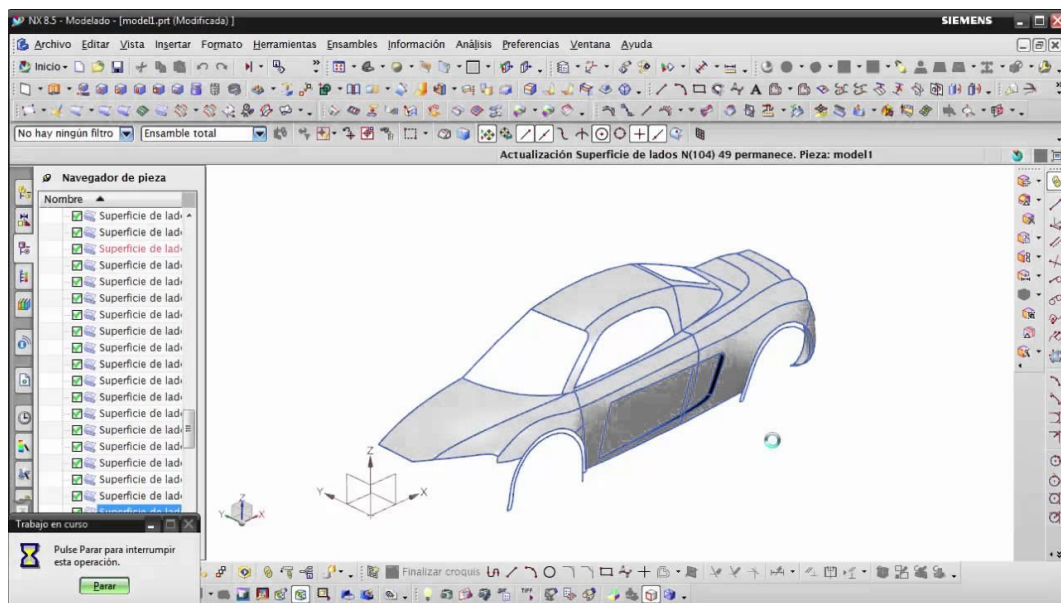


Figure 2-18: Car body design on Siemens NX.

CATIA, the most famous one, started as an in-house development in 1977 by French aircraft manufacturer AVIONS MARCEL DASSAULT, at that time customer of the CADAM software to develop Dassault's Mirage fighter jet. It was later adopted by many companies of the aerospace, automotive, shipbuilding and other industries (Boeing, Ferrari, Volkswagen, Audi...).

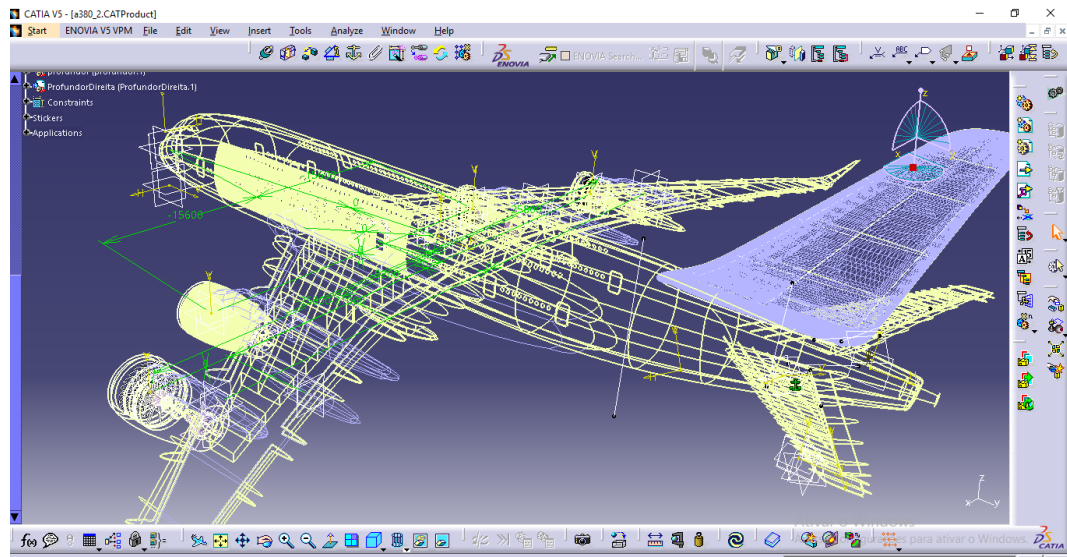


Figure 2-19: Airbus A380 modelling in CATIA V5.

3. ISOGEOMETRIC ANALYSIS WITH LS-DYNA

3.1. LS-DYNA

LS-DYNA is an advanced general-purpose multiphysics simulation software package developed by the Livermore Software Technology Corporation (LSTC). It is widely used in many fields, such as automotive, aerospace, bioengineering and civil engineering.

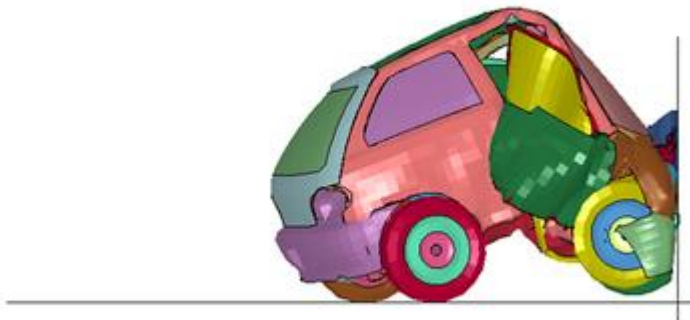


Figure 3-1: Screenshot from LS-PrePost showing the results of an LS-DYNA simulation of a car impacting a rigid wall at 120 kph. [Wikipedia]

LS-DYNA originated from the 3D FEA program **DYNA3D**, developed by Dr. John O. Hallquist at Lawrence Livermore National Laboratory (LLNL) in 1976. DYNA3D was created in order to simulate the impact of the Full Fusing Option (FUFO) or "Dyal-A-Yeld" nuclear bomb for low altitude release (impact velocity of ~ 40 m/s). At the time, no 3D software was available for simulating impact, and 2D software was inadequate. Though the FUFO bomb was eventually canceled, development of DYNA3D continued.

LS-DYNA's potential applications are numerous and can be tailored to many fields. LS-DYNA is not limited to any particular type of simulation. In a given simulation, any of LS-DYNA's many features can be combined to model a wide variety of physical events. An example of a simulation that involves a unique combination of features is the NASA JPL Mars Pathfinder landing which simulated the space probe's use of airbags to aid in its landing.

It is typically used for non-linear analysis (where therefore large deformations, variable boundary conditions or non-linear materials are involved), or for transient dynamic events simulations (automotive crash, explosions, sheet metal stamping etc.).

LS-DYNA is therefore used by the automotive industry to analyze vehicle designs. LS-DYNA accurately predicts a car's behavior in a collision and the effects of the collision upon the car's occupants. With LS-DYNA, automotive companies and their suppliers can test car designs without having to tool or experimentally test a prototype, thus saving time and expense.

LS-DYNA's specialized automotive features:

- Seatbelts
- Slip rings
- Pretensioners
- Retractors
- Sensors
- Accelerometers
- Airbags
- Hybrid III dummy models
- Inflator models

One example among all LS-DYNA's applications is sheet metal forming. LS-DYNA accurately predicts the stresses and deformations experienced by the metal, and determines if the metal will fail. LS-DYNA supports adaptive remeshing and will refine the mesh during the analysis, as necessary, to increase accuracy and save time.

Metal forming applications for LS-DYNA include:

- Metal stamping
- Hydroforming
- Forging
- Deep drawing
- Multi-stage processes

3.1.1 Pre-Post processors

LS-DYNA consists of a single executable file and is entirely command-line driven. Therefore, all that is required to run LS-DYNA is a command shell, the executable, an input file, and enough free disk space to run the calculation. All input files are in simple ASCII format and thus can be prepared using any text editor. Input files can also be prepared with the aid of a graphical preprocessor. There are many third-party software products available for preprocessing LS-DYNA input files. LSTC also develops its own preprocessor, **LS-PrePost**, which is freely distributed and runs without a license. Licensees of LS-DYNA automatically have access to all of the program's capabilities, from simple linear static mechanical analysis up to advanced thermal and flow solving methods. Furthermore, they have full use of LSTC's **LS-OPT** software, a standalone design optimization and probabilistic analysis package with an interface to LS-DYNA. [Wiki]

IGA models set up in this thesis will be carried out using LS-PrePost.

3.2. IGA modelling

The implementation of IsoGeometric Analysis in LS-DYNA started in 2011, when the keyword `ELEMENT_NURBS_PATCH_2D` appeared for the first time. During the years a lot of improvements, that will be explained in the next chapter have been made.

3.2.1 Shell elements

Since the most widely used and best understood mathematical description in CAD is based on Non Uniform Rational B-splines, NURBS-based shell and solid finite element have been implemented in LS-DYNA over the last few years.

The term “isogeometric analysis” is not restricted to any special type of basis functions. It just indicates that the geometrical description that is used for FEA is the same than was used in CAD before.

In case of NURBS shells, thin shell element based on Kirchhoff-love theory as well as shear deformable shell elements based on the Reissner-Mindlin shell theory are available, moreover the software allows to trim the standard patch with defined trimming loops and to apply contact boundary conditions. [27]

As indicated before, the definition of a NURBS surface necessitates a set of NURBS basis functions and associated control points with proper weights. The set of control points is called a control net (or control grid), which is similar to a finite element mesh with the very important difference, that the individual control points are normally not a part of the actual geometry.

This fact makes the application of boundary conditions, at the spot where they should be, a little bit more complicated. To solve this problem, the keyword `*CONSTRAINED_NODE_TO_NURBS_PATCH` is available, which allows to define a massless node on the actual NURBS surface and tie it to the NURBS patch. This will allow the application of either Neumann or Dirichlet boundary condition. [16]

Another difference to the definition of a classical finite element, a NURBS surface is described rather through a so-called NURBS-patch than through individual elements. A typical definition of a NURBS patch using the keyword `*ELEMENT_NURBS_PATCH_2D` is depicted in Fig. 4-2 together with the resulting subdivision into “finite elements”. [15]

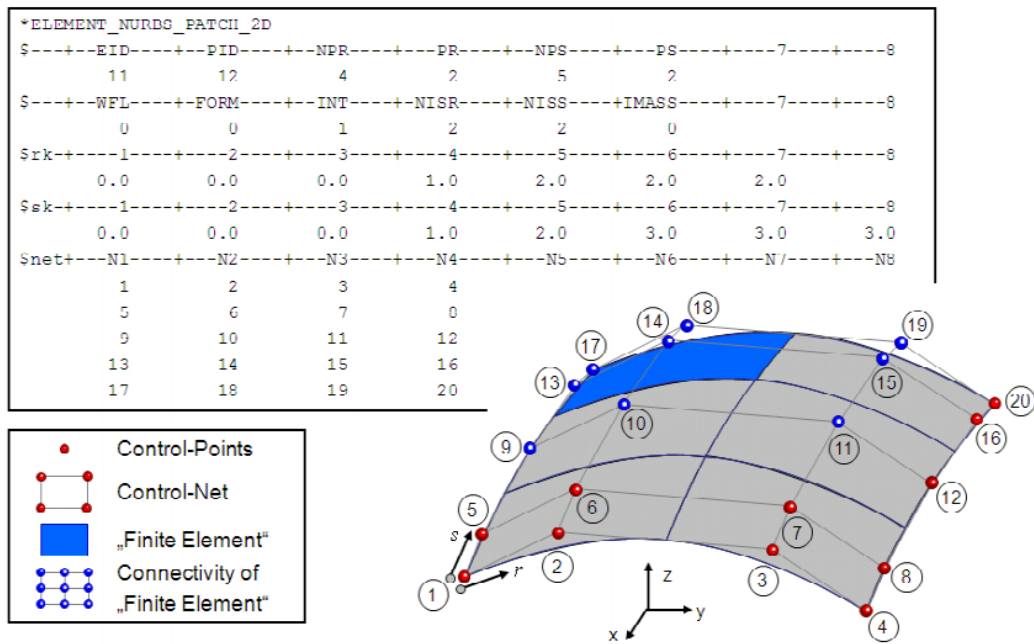


Figure 3-2: The keyword *ELEMENT_NURBS_PATCH_2D in LS-DYNA [15]

The contact treatment as well as the post-processing of the NURBS-finite elements in LS-DYNA is based on so-called interpolation nodes and elements. The idea is to superimpose a standard bi-linear mesh on top of each NURBS-finite-element by generating interpolation-nodes placed on the real surface. In Fig. 3-3 the result of an automatic generation of interpolation-nodes and elements is shown for the marked NURBS-element. With the parameters NISR and NISS (number of interpolation shells in local r/s-direction per NURBS-element) the user can specify the mesh density of the resulting interpolation elements. The term interpolation indicates that the constructed interpolation nodes are dependent nodes with respect to the control points. Their particular position is interpolated on basis of the actual location of the control points by using the corresponding NURBS-basis functions. In case of contact, the contact forces evaluated at the interpolation nodes will be extrapolated to equivalent forces at the primary variables at the control points. **Therefore the mesh density of the interpolation elements will not have any influence on the time step size nor on the overall number of degrees of freedom.**

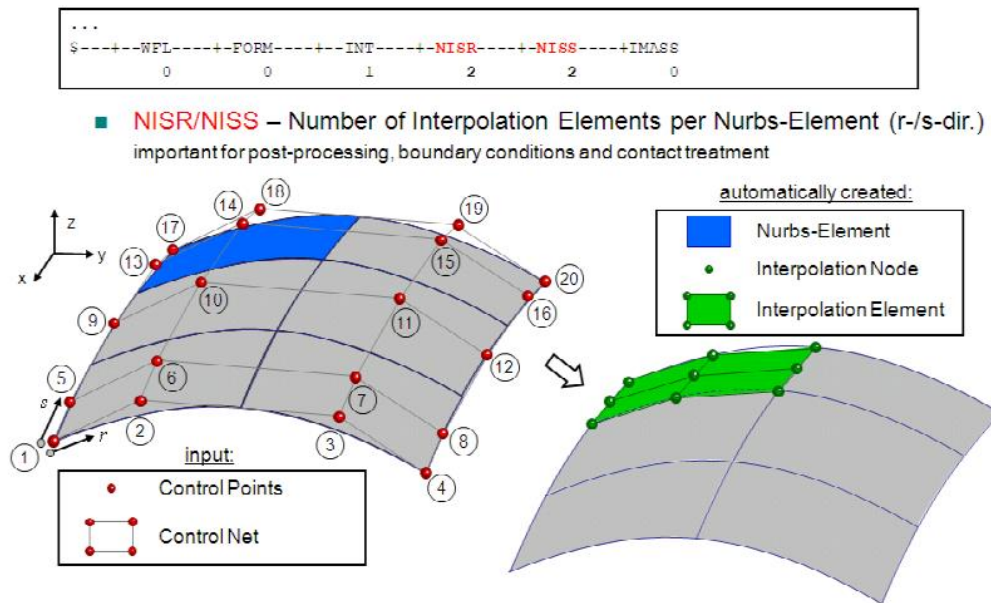


Figure 3-3: Interpolation nodes and elements [15]

3.2.3 Available shell formulations

At present five different shell formulations are available in LS-DYNA to use with the new NURBS based finite elements. They can be chosen under the parameter FORM in the keyword *ELEMENT_SHELL_NURBS_PATCH, with the following code:

- 0: “shear deformable theory” with rotational degrees of freedom
- 1: “shear deformable theory” without rotational degrees of freedom
- 2: “thin shell theory” without rotational degrees of freedom
- 3: “thin shell theory” with rotational degrees of freedom
- 4: combination of FORM 0 and FORM 1(allows the mixture of control points with and without rotational DOFs. This might be useful at the boundaries of NURBS patches where the continuity usually drops to C0 and rotational DOFs are necessary)

The shear deformable theory is based on the degenerated solid element formulation. The kinematics is defined in terms of the nodal coordinates at the reference lamina of the shell, and a unit orientation vector, \hat{y} , which we take to be the unit normal as in the Belytschko-Tsay element,

$$x(\eta, \xi, \zeta) = \sum_i N_i(\eta, \xi) \left(x_i + \frac{h}{2} \zeta \hat{y}_i \right) \quad (3.2.1)$$

and the velocity is expressed as

$$\dot{x}(\eta, \xi, \zeta) = \sum_i N_i(\eta, \xi) \left(\dot{x}_i + \frac{h}{2} \zeta \dot{\hat{y}}_i \right) \quad (3.2.2)$$

The formulations with and without rotations differ only in the expression of the time derivative of the unit orientation vector. With rotations, the derivative is:

$$\dot{\hat{y}}_i = \omega_i \times \hat{y}_i \quad (3.2.3)$$

where ω_i is the angular velocity at the control point, and, for the formulation without rotations, the rate is obtained by differentiating the expression for the orientation vector as a function of the control point coordinates with time,

$$\dot{\hat{y}}_i = \sum_j \frac{\partial \hat{y}_i}{\partial x_j} \dot{x}_j \quad (3.2.4)$$

The thin shell formulation is similar, however, the normal vector is evaluated at the current point on the reference lamina,

$$x(\eta, \xi, \zeta) = \sum_i N_i(\eta, \xi) x_i + \frac{h}{2} \zeta \hat{n}(\eta, \xi) \quad (3.2.5)$$

Differentiating with time gives the velocity:

$$\dot{x}(\eta, \xi, \zeta) = \sum_i N_i(\eta, \xi) \dot{x}_i + \frac{h}{2} \zeta \dot{\hat{n}}(\eta, \xi) \quad (3.2.6)$$

As with the shear deformable theory, implementations with and without rotations are created based on the how the time derivative of the normal is evaluated. For the formulations that are rotation free, the basis functions must have first derivatives that are continuous across the element boundaries to correctly transmit the moments between adjacent elements. Due to the generally higher continuity of the NURBS-finite elements it is possible to use rotation free shell formulations. This leads to a significant reduction of global degrees of freedom and automatically removes possible problems with the treatment of rotational inertias of classical shell formulations. [15]

3.2.4 ELEMENT_SHELL_NURBS_PATCH KEYWORD

ELEMENT_SHELL_NURBS_PATCH is the keyword used to define isogeometric shell elements in the latest release of LS-DYNA. This chapter will focus on the explanation of its cards and variables.

Title Card. Additional card for the TITLE keyword option.

Card Title	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	A80							

<u>VARIABLE</u>	<u>DESCRIPTION</u>
TITLE	Description of the NURBS patch surface

Card 1	1	2	3	4	5	6	7	8
Variable	NPID	PID	NPR	PR	NPS	PS	PERIR	PERIS
Type	I	I	I	I	I	I	I	I
Default	none	none	None	none	none	none	0	0

Card 2	1	2	3	4	5	6	7	8
Variable	WFL	FORM	INT	NISR	NISS	IMASS		IDFNE
Type	I	I	I	F	F	I		I
Default	0	0	0	PR	PS	0		0

Figure 3-4: Data cards: title card, card 1, card 2 [28]

The variables operations and values are described in the table below. [28]

VARIABLE	DESCRIPTION
NPID	NURBS surface element / patch ID. A unique number must be chosen.

PID	PART ID.
NPR	Number of control points in the local r -direction.
PR	Polynomial degree of the basis function in the local r -direction.
NPS	Number of control points in the local s -direction.
PS	Polynomial degree of the basis function in the local s -direction.
PERIR	Flag for surface periodicity in the r -direction: EQ.0: Non-periodic EQ.1: Periodic
PERIS	Flag for surface periodicity in the s -direction: EQ.0: Non-periodic EQ.1: Periodic
WFL	Flag for user defined control weights: EQ.0: Control weights are assumed to be uniform and positive; that is, the surface is a B-spline surface. No optional Cards D is allowed. EQ.1: Control weights are defined using optional Cards D.
FORM	Available shell formulations, as previously discussed.
INT	In-plane numerical integration rule: EQ.0: Uniformly reduced Gauss integration. $NIP = PR \times PS$.

	<p>EQ.1: Full Gauss integration. $NIP = (PR+1) \times (PS+1)$.</p> <p>EQ.2: Reduced, patch-wise integration rule for C1 continuous quadratic NURBS surfaces.</p>
NISR	<p>Number or average edge length of automatically created interpolation shell elements per each knot span in the r-direction.</p> <p>GT.0: $NINT(NISR)$ is the number of interpolation elements in the r-direction</p> <p>LT.0: $NISR$ is the average edge length of the interpolation elements in the r-direction.</p>
NISS	<p>Number or average edge length of automatically created interpolation shell elements per each knot span in the s-direction.</p> <p>GT.0: $NINT(NISS)$ is the number of interpolation elements in the s-direction</p> <p>LT.0: $NISS$ is the average edge length of the interpolation elements in the s-direction.</p>
IMASS	<p>Mass matrix lumping scheme:</p> <p>EQ.0: Row sum.</p> <p>EQ.1: Diagonal weighting.</p>
IDFNE	<p>Element ID of first NURBS-Element within this NURBS-Patch definition.</p>

Knot Vector Cards (for r -direction). The knot vector in r -direction with length $NPR + PR + 1$ is given below requiring a total of $\text{ceil}[(NPR+PR+1)/8]$ cards. The knot vector must be normalized to the $[0,1]$ interval.

Cards A	1	2	3	4	5	6	7	8
Variable	RK1	RK2	RK3	RK4	RK5	RK6	RK7	RK8
Type	F	F	F	F	F	F	F	F
Default	none	none	None	none	none	none	none	none

Figure 3-5: knot vector cards for r -direction. Variables RK_m are the values of the univariate knot vector in r -direction [28]

Knot Vector Cards (for s -direction). The knot vector in s -direction with length $NPS + PS + 1$ is given below requiring a total of $\text{ceil}[(NPS+PS+1)/8]$ cards. The knot vector must be normalized to the $[0,1]$ interval.

Cards B	1	2	3	4	5	6	7	8
Variable	SK1	SK2	SK3	SK4	SK5	SK6	SK7	SK8
Type	F	F	F	F	F	F	F	F
Default	none	none	None	none	none	none	none	none

Figure 3-6: knot vector cards for s -direction. Variables SK_m are the values of the univariate knot vector in s -direction [28]

Connectivity Cards. The connectivity of the control grid is a two dimensional table of NPS rows and NPR columns. This data fills the NPS sets (one set for each row) of NPR points tightly packed into $\text{ceil}(NPR/8)$ connectivity cards, for a total of $NPS \times \text{ceil}(NPR/8)$ cards.

Cards C	1	2	3	4	5	6	7	8
Variable	N1	N2	N3	N4	N5	N6	N7	N8
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Figure 3-7: Control Grid cards, variables N_k are the control point IDs, defined via *NODE, to define the control grid - LT.0: Control point with rotational DOFs for FORM = 4 /-4; [28]

Control Weight Cards (Optional). Additional cards are used to set a weight for each control point if $WFL = 1$ on Card 2. These cards have an ordering identical to the connectivity cards (Cards C).

Cards D	1	2	3	4	5	6	7	8
Variable	W1	W2	W3	W4	W5	W6	W7	W8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Figure 3-8: Control weight cards, variables W_k are the control weights of the surface patch. [28]

Trimming loop title card

Cards E	1	2	3	4	5	6	7	8
Variable	TITLE							
Type	A80							
Default	trimming loop							

Figure 3-9: Trimming loop title card. [28]

Trimming Loop Connectivity Cards

Cards F	1	2	3	4	5	6	7	8
Variable	C1	C2	C3	C4	C5	C6	C7	C8
Type	I	I	I	I	I	I	I	I

Figure 3-10: Variables C1 are Trimming curve ID pointing to a curve defined using *DEFINE_NURBS_CURVE. A unique number has to be chosen. [28]

Trimmed NURBS surface elements / patches can be analyzed by defining trimming loops. A trimming loop is formed by a set of NURBS curves defined in the surface parametric coordinate system.

Each trimming curve is defined using the *DEFINE_NURBS_CURVE keyword. Trimming loops may be given a distinct title on Cards E and the connectivity of trimming curves defining a loop is stored on Cards F. The end and starting points of two consecutive curves must coincide. If the loop is defined by a single curve, the starting and end points of the curve must match. Furthermore, the orientation of the trimming loop is essential to define the trimmed surface. Travelling along the trimming loop, the surface on the right-hand side of the loop will be trimmed. There is no limitation on either the number of trimming curves forming a loop or the number of loops used to trim a NURBS surface element/patch. [28]

Important remarks:

- Shell thickness** is defined in *SECTION_SHELL and referenced via *PART.
- **ELFORM=201** has to be used in *SECTION_SHELL.

3.2.5 DEFINE_NURBS_CURVE keyword

Purpose: Define a NURBS curve using a univariate knot vector, a control polygon, and optionally a set of control weights. The knot vector defines the necessary shape functions and parameterize the curve. There is no limit on the size of the input data. Hence, the total number of keyword cards depends on the

parameters defined on the first card. The total number of cards is $1 + \text{ceil}[(N+P+1)/8] + N$, where N and P designate the number of control points forming the control polygon and the polynomial degree, respectively. While the keyword is meant to store NURBS curves in two or three spatial dimension, it is also employed to describe trimming curves defining trimmed NURBS elements/surfaces, see *ELEMENT_SHELL_NURBS_PATCH for further details. In the latter case, the control point coordinates are in fact parametric coordinates of the surface to be trimmed, i.e. $(x, y, z, w) = (r, s, 0, w)$ and TYPE = 1 on Card 1.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	N	P	PERI	TYPE	WFL		
Type	I	I	I	I	I	I		
Default	none	none	none	0	0	0		

Figure 3-11: *DEFINE_NURBS_CURVE card. [28]

Cards A	1	2	3	4	5	6	7	8
Variable	K1	K2	K3	K4	K5	K6	K7	K8
Type	F	F	F	F	F	F	F	F
Default	none	none	none	none	none	none	none	none

Figure 3-12: Knot vector card. The knot vector of length N+P+1 is given below requiring $\text{ceil}[(N+P+1)/8]$ cards in total. The knot vector has to be normalized to the [0,1] interval. [28]

Cards B	1	2	3	4	5	6	7	8
Variable	X	Y	Z	W				
Type	F	F	F	F				
Default	none	none	none	none				

Figure 3-13: Control point cards. The spatial coordinates of the control points and the control weights are listed on N cards. Control weight entries are disregarded unless WFL = 1 on Card 1. [28]

Variable	Description
ID	Curve ID. A unique number has to be chosen.
N	Number of control points.
P	Polynomial degree.
PERI	Flag for curve periodicity. EQ.0: Non-periodic. EQ.1: Periodic.
TYPE	Coordinate type. EQ.0: Spatial. EQ.1: Parametric.
WFL	Flag for user defined control weights. EQ.0: Control weights are assumed to be uniform and positive, i.e. the curve is a B-spline curve, and the fourth entries on cards B are disregarded.

	EQ.1: Control weights are defined on the forth entry of cards B.
Kn	Values of the univariate knot vector defined in cards A with $n=1,\dots,N+P+1$.
Xk	Spatial coordinates in the global X direction defined in cards B with $k=1,\dots,N$.
Yk	Spatial coordinates in the global Y direction defined in cards B with $k=1,\dots,N$.
Zk	Spatial coordinates in the global Z direction defined in cards B with $k=1,\dots,N$.
Wk	Control weights defined in cards B with $k=1,\dots,N$.

```

*DEFINE_NURBS_CURVE
$ Card 1
$-----ID-----+-----N-----+-----P-----+-----PBRI-----+-----TYPE-----+-----WFL-----
               1               6               2               0               0               1
$ Cards A
$K-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----
               0.0               0.0               0.0               0.25               0.5               0.75               1.0               1.0
               1.0
$ Cards B
$-----+-----X-----+-----Y-----+-----Z-----+-----W-----
-0.353    0.471    0.850    1.000
 0.050    0.600    1.000    1.000
 0.444    0.444    0.900    0.995
 0.125    0.000    0.800    1.000
 0.222   -0.556    0.900    1.010
-0.353   -0.529    0.850    1.000

```

Figure 3-14: Defining a quadratic NURBS curve using the *DEFINE_NURBS_CURVE keyword. [28]

3.2.6 Solid NURBS elements

Cad models are collections of surfaces and for this reason not suitable for the analysis of solid structures. Therefore the behaviour of IGA solid models is not well understood but still in evolution and already implemented in LS-DYNA. [29]

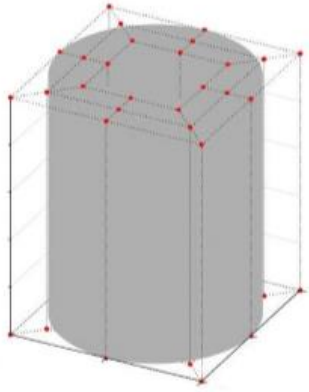


Figure 3-15: A NURBS volume example. [29]

The keyword `ELEMENT_SOLID_NURBS_PATCH` is used to define the solid NURBS element. The concept is similar to the `ELEMENT_SHELL_NURBS_PATCH` keyword, with a further spatial direction. A NURBS-block element (patch) based on a cuboid grid of control points is defined. This grid consists of $NPR \cdot NPS \cdot NPT$ control points, where NPR , NPS and NPT are the number of control points in local r -, s - and t -directions, respectively. The necessary shape functions are defined through three knot-vectors:

1. Knot-Vector in r -direction with length $NPR + PR + 1$
2. Knot-Vector in s -direction with length $NPS + PS + 1$
3. Knot-Vector in t -direction with length $NPT + PT + 1$

There is no limit on the size of the underlying grid to define a NURBS-block element, so the total number of necessary cards depends on the parameters given in the first two cards and is given by:

$$\begin{aligned} \# \text{ of cards} = & 2 + \left\lceil \frac{NPR + PR + 1}{8} \right\rceil + \left\lceil \frac{NPS + PS + 1}{8} \right\rceil + \left\lceil \frac{NPT + PT + 1}{8} \right\rceil \\ & + NPT \times NPS \times \left\lceil \frac{NPT}{8} \right\rceil \end{aligned}$$

where $\lceil x \rceil = \text{ceil}(x)$. (NOTE: the last term in the sum is doubled if WFL = 1, indicating that the weights are user-specified).

Card 1	1	2	3	4	5	6	7	8
Variable	NPID	PID	NPR	PR	NPS	PS	NPT	PT
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

Card 2	1	2	3	4	5	6	7	8
Variable	WFL	NISR	NISS	NIST	IMASS	IINT		IDFNE
Type	I	I	I	I	I	I		I
Default	0	PR	PS	PT	0	0		0

Figure 3-16: ELEMENT_SOLID_NURBS_PATCH, Cards 1 and 2. [28]

The Variables have an analogy in name and function to the case of shell formulation, obviously with the addition of t-direction.

The same worths for the **knot vector cards**, with variables RK, SK, TK, the values of the univariate knot vector in local $r/s/t$ -direction.

The connectivity of the control grid is a two dimensional table of $NPT \times NPS$ rows and NPR columns. This data fills $NPT \times NPS$ sets (one set for each row) of NPR points tightly packed into $\text{ceil}(NPR/8)$ **Connectivity Cards**, for a total of $NPT \times NPS \times \text{ceil}(NPR/8)$ cards.

The same ordering is kept in the **Weight cards**, as in shell case additional card for $WFL \neq 0$ sets a weight for each control point. (Variables W_i are the weighting factor of the i^{th} control point.

For $WFL = 0$, all weights at the control points are set to 1.0 (B-spline basis) and no optional Card 7 sets are allowed.

Important remark: **ELFORM = 201** has to be used in *SECTION_SOLID.

Example:

```
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
$ An Isogeometric Solid NURBS Example :
$...|...1...|...2...|...3...|...4...|...5...|...6...|...7...|...8
*SECTION SOLID
$#   secid   elform
      1      201
*ELEMENT SOLID_NURBS_PATCH
$CARD 1
$#   npeid   pid   npr   pr   nps   ps   npt   pt
      1      1      3      2      6      2      3      2
$CARD 2
$#   wfl   nistr   niss   nist   imass
      0      2      2      2      0
$CARD A
$#   rk1   rk2   rk3   rk4   rk5   rk6   rk7   rk8
      0.0   0.0   0.0   1.0   1.0   1.0
$CARD B
      0.0   0.0   0.0   0.25   0.5   0.75   1.0   1.0
      1.0
$CARD C
      0.0   0.0   0.0   1.0   1.0   1.0
$CARD D
      1001   1002   1003
      ""
      1052   1053   1054
$CARD E (Optional if wfl .eq. 0)
```

Figure 3-17: Example of definition of ELEMENT_SOLID_NURBS_PATCH. [28]

3.3. Contacts and boundary conditions

On top of the NURBS patches, LS-DYNA automatically creates bi-linear shell elements (interpolation elements), whose nodes (interpolation nodes) are placed on the real surface. The interpolation elements may be used to apply boundary conditions (i.e. contact) and for postprocessing. Its resolution can be defined using the parameters NISR and NISS (see figure 3-18). It is important to notice, that the interpolation nodes are fully constrained to the underlying NURBS patch. For instance, contact forces are in fact first evaluated at the interpolation nodes, but then transferred to the primary degrees of freedom (DOF) at the control points. The actual analysis is exclusively performed using the NURBS elements and their corresponding DOFs. For post-processing, results at the integration points of the NURBS elements are mapped onto the interpolation elements, such that standard post-processing tools can be used. [30]

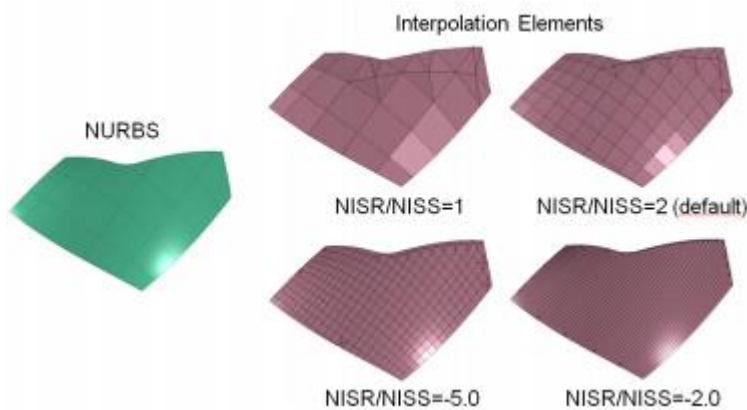


Figure 3-18: Bi-quadratic NURBS patch and interpolation elements dependent on the parameter NISR and NISS. [30]

3.3.1 Penalty based contacts

Up to now, all penalty based contacts are available for IGA.

“Single surface” are the most widely used contact options in LS-DYNA, especially for crashworthiness applications. With these types, the slave surface is typically defined as a list of part ID’s. No master surface is defined. Contact is

considered between all the parts in the slave list, including self-contact of each part. If the model is accurately defined, these contact types are very reliable and accurate. However, if there is a lot of interpenetrations in the initial configuration, energy balances may show either a growth or decay of energy as the calculation proceeds.

For crash analysis, ***CONTACT_AUTOMATIC_SINGLE_SURFACE** is recommended. This contact has been improved from version to version of LS-DYNA implementing now also NURBS elements and is the most popular contact option.

In automotive crashworthiness, also the two-ways treatment of contact can be used. ***CONTACT_AUTOMATIC_SURFACE_TO_SURFACE** is a good choice with Isogeometric elements.

3.3.2 Tied NURB contacts

In tied contact types, the slave nodes are constrained to move with the master surface. At the beginning of the simulation, the nearest master segment for each slave node is located based on an orthogonal projection of the slave node to the master segment. If the slave node is deemed *close* to the master segment based on established criteria, the slave node is moved to the master surface. In this way, the initial geometry may be slightly altered without invoking any stresses.

It is always recommended that tied contacts NOT be defined by part Ids but rather by node/segment sets. In this way, the user has more direct control over what gets tied to what and thus can prevent unintended constraints.

As the simulation progresses, the isoparametric position of the slave node with respect to its master segment is held fixed using kinematic constraint equations. Currently, 3 pure nurbs tied contact definition are implemented:

-NURBS_TIED_NODES_TO_SURFACE

-NURBS_TIED_EDGE_TO_SURFACE

-NURBS_TIED_EDGE_TO_EDGE

All of them are defined with the same variables: **SSID**, **MSID** that are the master and slave IDs, whose types are chosen with **SSTYP** and **MSTYP**:

EQ.0:Nurbs-Patch Element ID.

EQ.2:part set ID.

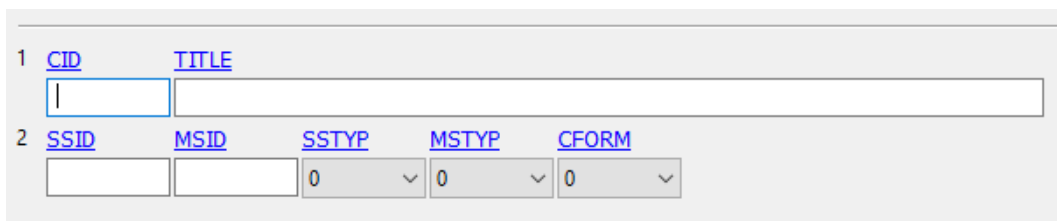
EQ.3:part ID,

EQ.4:node set ID for node to surface contact.

CFORM is the constraint formulation, defined with:

EQ.0:mortar method (currently it doesn't work).

EQ.1:point-to-point formulation.



1	<u>CID</u>	<u>TITLE</u>			
2	<u>SSID</u>	<u>MSID</u>	<u>SSTYP</u>	<u>MSTYP</u>	<u>CFORM</u>
			0	0	0

Figure 3-19: NURBS_TIED keyword interface in LS-PrePost.

NURBS_TIED_EDGE_TO_EDGE is the most interesting of the 3, because it allows the connection of multi-patch structures. It is currently available only in the SMP (shared memory parallel processing) version of LS-DYNA.

In CAD environment, geometries are typically defined by a so called boundary representation (B-Rep). This means that the standard NURBS patch is described together with a set of outer trimming curves, that define visible and invisible regions on the surface. Moreover specific topology information are embedded in the cad to specify if two or more patches represent a connected part. In analysis environment, anyway, the solver needs to make sure that certain mechanical properties, like stresses, strain and bending moments are transferred across the patches, through the common interface.

To mechanically couple the individual elements, some algorithm basic ideas:

Interface condition, strong form:

The condition to fulfill is that rotations and displacements must be the same at both sides of the interface. In a strong form this can be written as:

$$u_1 = u_2|_I; \theta_1 = \theta_2|_I \quad (3.3.1)$$

Where I represents the interface.

Interface condition, penalty weak form:

Using a penalty type formulation, the interface conditions can be translated into a weak, integral form:

$$\alpha^{disp} \int_I (\mathbf{u}_1 - \mathbf{u}_2)(\delta \mathbf{u}_1 - \delta \mathbf{u}_2) d\Gamma = 0 \quad (3.3.2)$$

$$\alpha^{rot} \int_I (\theta_1 - \theta_2)(\delta \theta_1 - \delta \theta_2) d\Gamma = 0 \quad (3.3.3)$$

Where α^{disp} and α^{rot} are the penalty factors for enforcing the displacements and the rotational constraints and $\delta \theta$ and $\delta \mathbf{u}$ are variations of the continuous displacement and rotation fields, sometimes also called virtual displacement and rotations, in the context of the principle of virtual work. [27]

The enforcement of the rotational constraint (EQ. 3.3.3), can be directly related to the rotational DOFs in case of shear deformable shell formulation, but, for thin shell element formulations, no rotational DOFs are introduced, so the rotational constraint is enforced in a different way. A total Lagrangian constraint formulation is used:

$\begin{aligned} \sin(\theta - \theta_0) &= \sin(\theta) \cos(\theta_0) - \cos(\theta) \sin(\theta_0) \\ &= 0; \cos(\theta) = \mathbf{n}_1 \cdot \mathbf{n}_2 \text{ and } \sin(\theta) \\ &= \mathbf{t} \cdot \mathbf{n}_1 \otimes \mathbf{n}_2 \end{aligned}$	$(3.3.4)$
---	-----------

Where θ and θ_0 represent the initial and current angle at the interface, \mathbf{n}_1 and \mathbf{n}_2 are the surface normal vectors of the two joining patches at the interface, and \mathbf{t} is the tangent vector at the interface curve. This constraint expression is evaluated at the integration points of the B-Rep edge elements and numerically integrated along the interface [27].

B-Rep edge elements are used to discretize the integration domain along the interface, then, as in standard contact algorithms, one side of the interface is chosen to be the master side and one side is chosen to be the slave side. On the master side the numerical integration is performed, thus, necessary terms including shape functions and their derivatives need to be mapped from the parameter space of the slave side to the one of the master side.

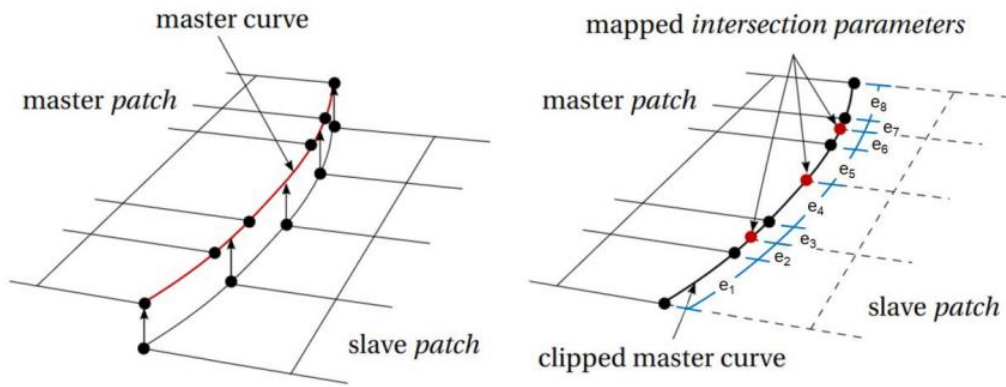


Figure 3-20: Mapping procedure to define the B-Rep edge elements on the master curve. A B-Rep edge element is defined between any two consecutive points (either black or red) shown in the right part of the figure. [27]

Once the numerical integration in the B-Rep edge elements is performed, equivalent penalty forces and stiffnesses (for implicit analyses) for the involved control points are computed and assembled to the global force vector and stiffness matrix if necessary. [27]

4. LS-PrePost Isogeometric analysis set up and capabilities

LS-PrePost is an advanced pre/post processor available for free from LSTC, LS-PP is particularly capable of importing, editing and exporting LS-DYNA keyword files for generating LS-DYNA input files and processing the results from LS-DYNA analyses.

For the purpose of this thesis, the capabilities of the softwares regarding IGA models have been examined, in order to set up a guide updated to the latest improvements.

As stated by Mr. Hartmann from Dynamore, it is still difficult and cumbersome to set up an IGA model for LS-DYNA. This is especially true in case the underlying CAD data is rather complex, with many patches, different orders and highly different “mesh sizes” (knot spans mapped to physical space).

The instability and the not easy set up of the models, nowadays, can be the biggest downside of IGA analyses compared with classic FEA. Anyway the great potential of the method is pushing the software houses to develop new solutions every year.

4.1. Shell NURBS elements set up

As explained before, to define a NURBS shell element in LS-DYNA, the keyword `ELEMENT_SHELL_NURBS_PATCH` is needed.

4.1.1 ELEMENT_SHELL_NURBS_PATCH creation

For simple patch shapes, this operation can be manually done from the keyword input form, with free choice for what concerns knot spans and polynomial orders.

Anyway, this operation is definitely not practical and not suited for multi patch structures and complex geometries, that are the main target of IGA.

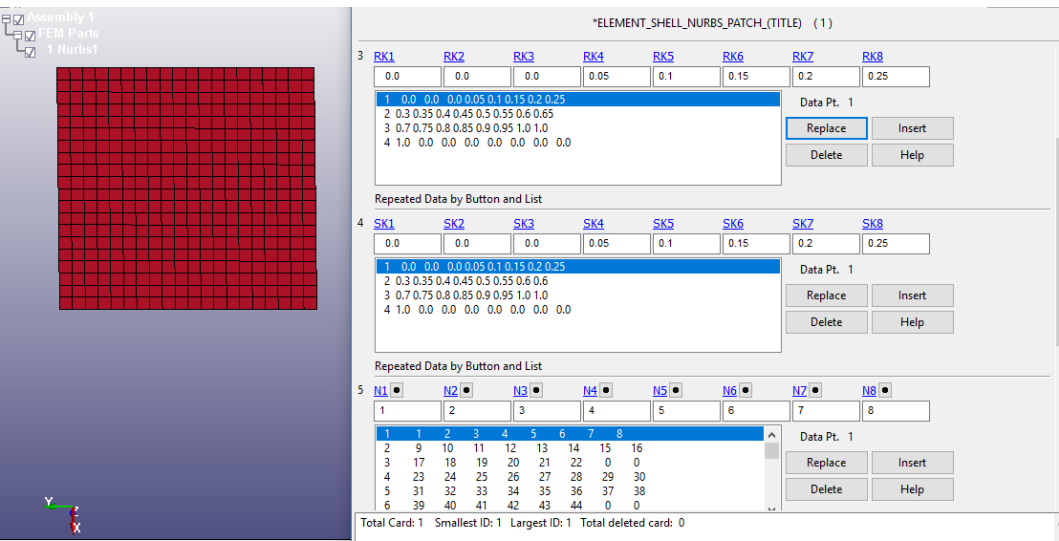


Figure 4-1: Keyword input editor. Variables RK, SK and N are defined for the NURBS patch on the left.

Although LS-PrePost allows simple spline, surface and solid modelling, the easiest way to define a model, is obviously to start from an imported CAD geometry. All the most common formats (STEP, IGES, STL) are supported by the software.

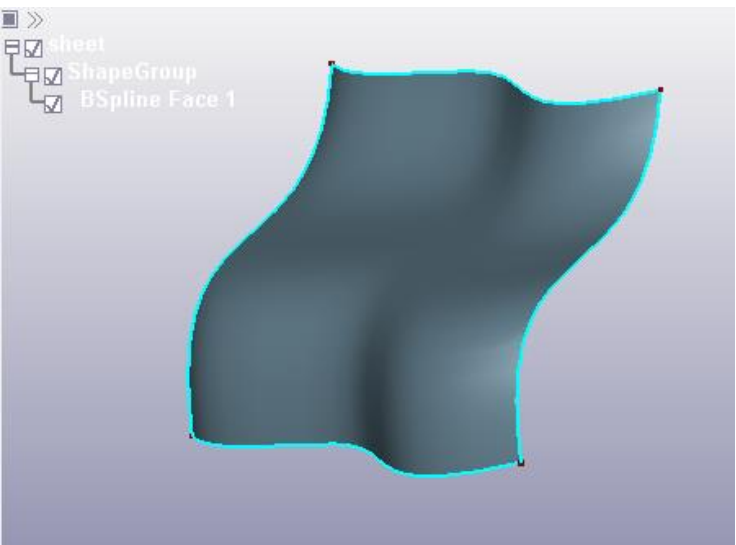


Figure 4-2: Single patch CAD surface imported in LS-PrePost.

To create Isogeometric elements, a NURBS editor is implemented in LS-PrePost, to deal with shell elements the NURBS 2D EDITOR must be utilized.

By means of the function “create”, the user can generate one or more NURBS patch element. A useful subfunction is “build contact”, that builds a NURBS_TIED_EDGE_TO_EDGE contact on the common edges of multi patch structures. In case this procedure does not work automatically (no common edge found), the edge curve must be manually defined via DEFINE_NURBS_CURVE or DEFINE_CURVE_TRIM_3D keyword. With the latter the conversion of CAD imported curve into the keyword itself is possible.

This issue can increase the complexity of the model set up, especially in case of intricate geometries.

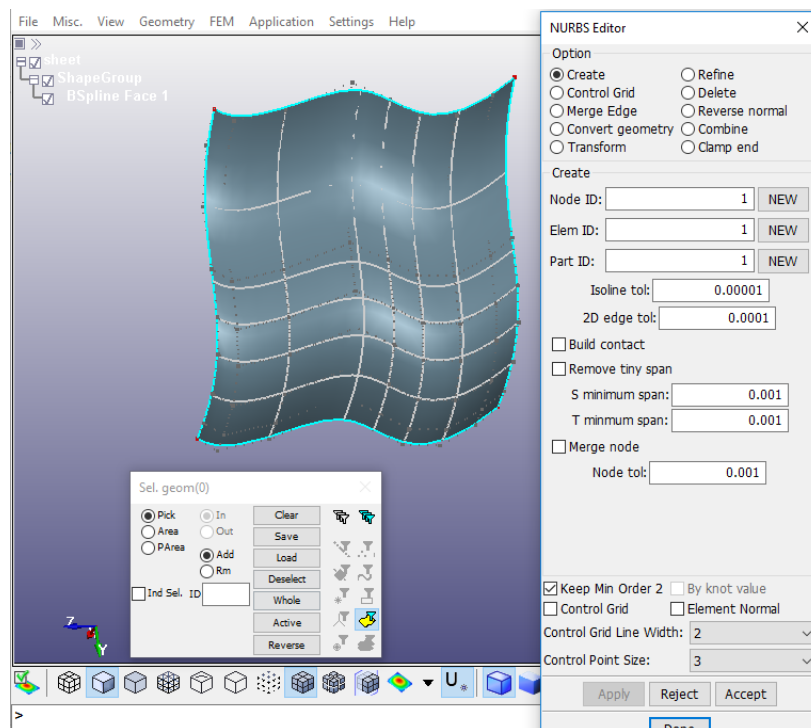


Figure 4-3: NURBS single patch element creation in LS-PrePost

In figure 4-3 a single NURBS patch is created from CAD geometry, while in figure 4-4 a simple multi patch structure is examined (square tube). This is useful to understand the “build contact” command but also to understand the

concept of “elements” in IGA. In IGA the number of elements does not depend on the mesh size, that represents the knot vector spans, but only on the number of patches.

For this reason, even with a refined mesh, the tube in figure 4-4 is composed by four elements, with four NURBS_TIED_EDGE_TO_EDGE contacts, automatically created using the command “build contact” (one for each common edge between the patches).

The number of parts created (one) can be also noticed from the picture. This is not due to the presence of the contact but only to the fact that the patches were created together (into the same part ID), so the existence of a unique part with not mechanically connected patches is possible.

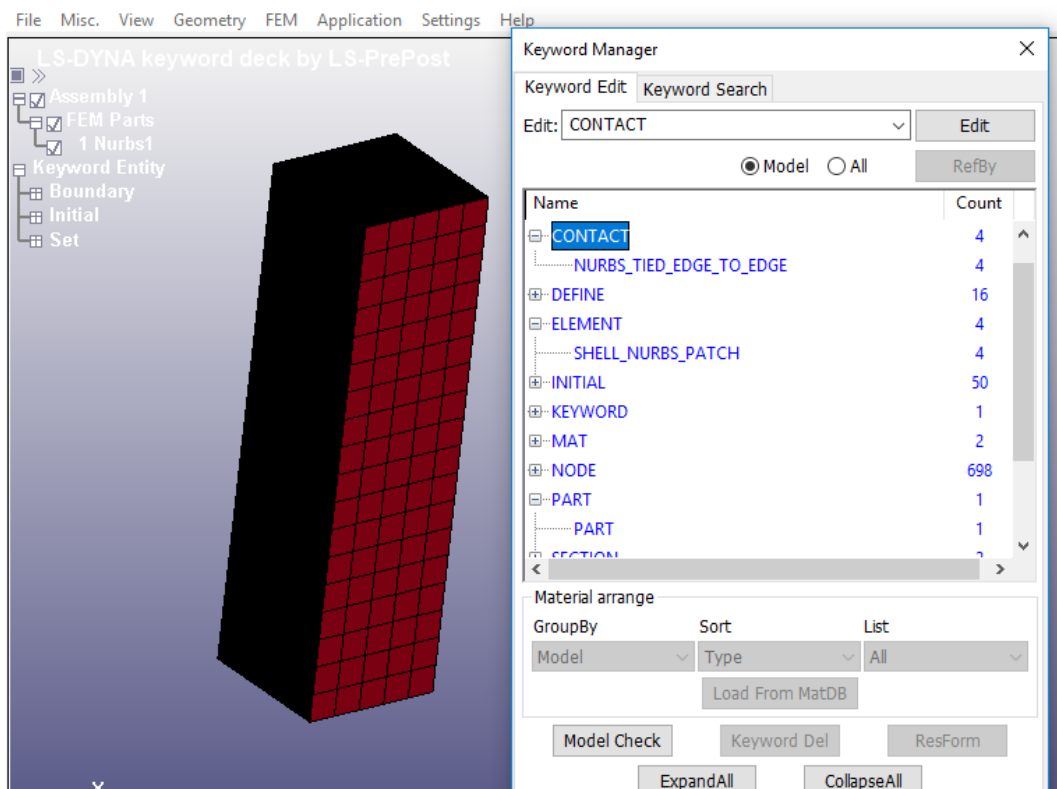


Figure 4-4: Multi patch Isogeometric part. Four SHELL_NURBS_PATCH elements are created, with four NURBS_TIED_EDGE_TO_EDGE contacts. In order to form a single part, the NURBS patch elements were created using the same part ID.

4.1.2 ELEMENT_SHELL_NURBS_PATCH refinement

As explained in chapter 1, in b-spline environment we have the possibility to perform 3 types of refinement: H, P and K-refinement.

H-refinement: LS-PrePost allows either to change the knot vector spans or to choose the number of segments (“subdivide” command) in **r** and **s** directions, the mesh size of the model derives from this process. To have reliable analysis results the mesh size should be chosen comparable to a normal FE model.

Due to IGA nature, obviously, the geometry doesn’t change if the mesh size does.

P-refinement: The polynomial degree can be increased without changing the geometry or the parametrization (nodes multiplicity must be increased in order to maintain the initial continuity).

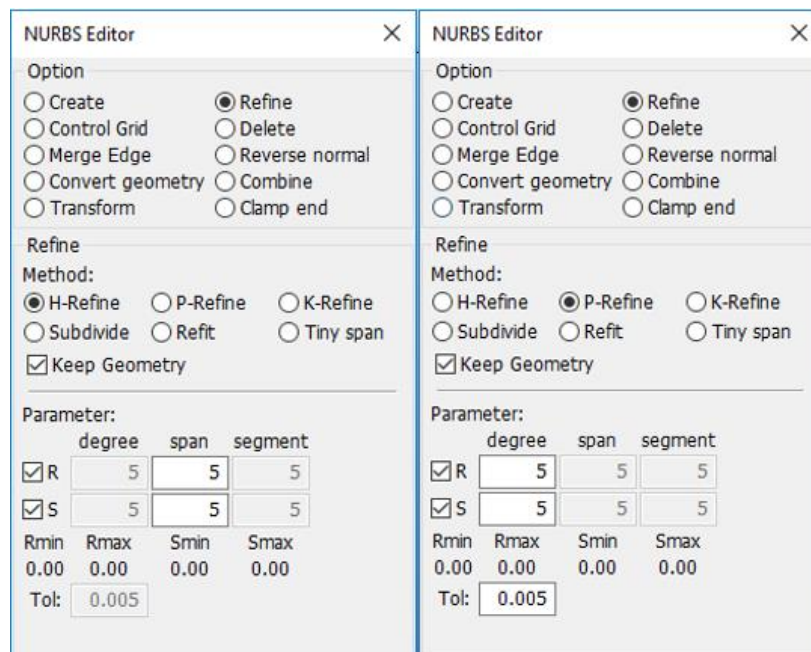


Figure 4-5: H-refinement (left) and P-refinement (right) interfaces.

K-refinement: order elevation is followed by knot insertion in order to keep a good continuity, LS-PrePost allows to play with both polynomial degree and knot vectors spans.

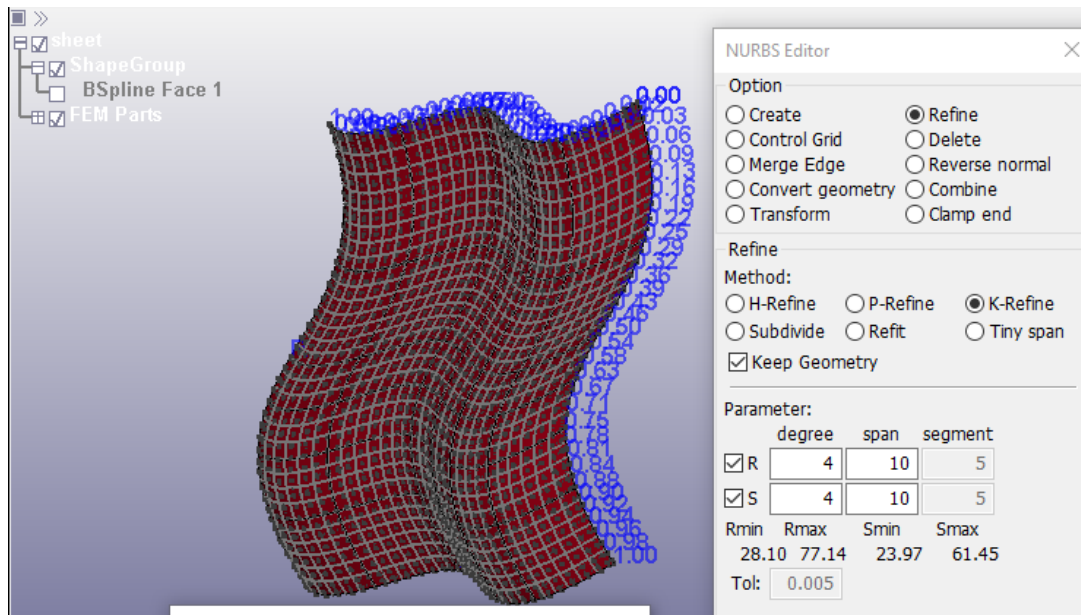


Figure 4-6: K-refinement interface of LS-Prepost.

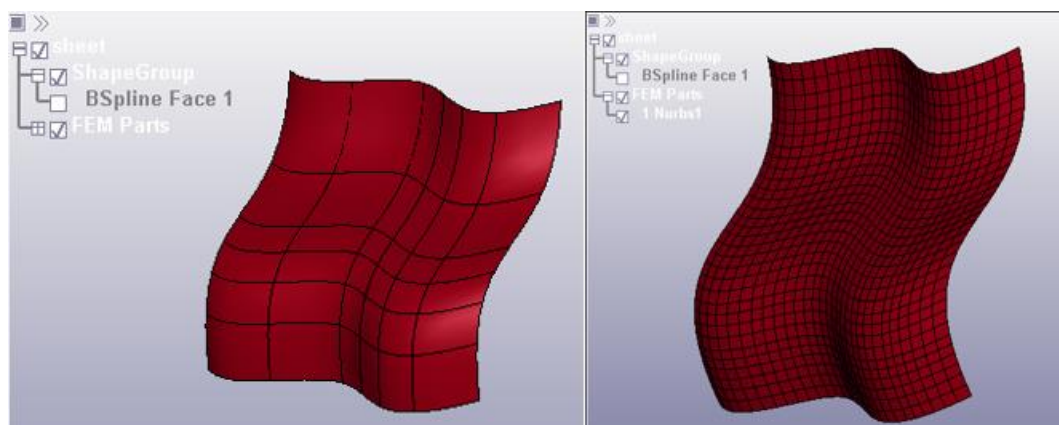


Figure 4-7: The same NURBS patch element, after creation (left) and after refinement (right)

4.1.3 Trimmed NURBS patch

When a trimmed NURBS patch is imported from CAD, LS-PrePost is able to define the trimming loops and to create the trimmed element (ELEMENT_SHELL_NURBS_PATCH_TRIMMED) automatically from the editor.

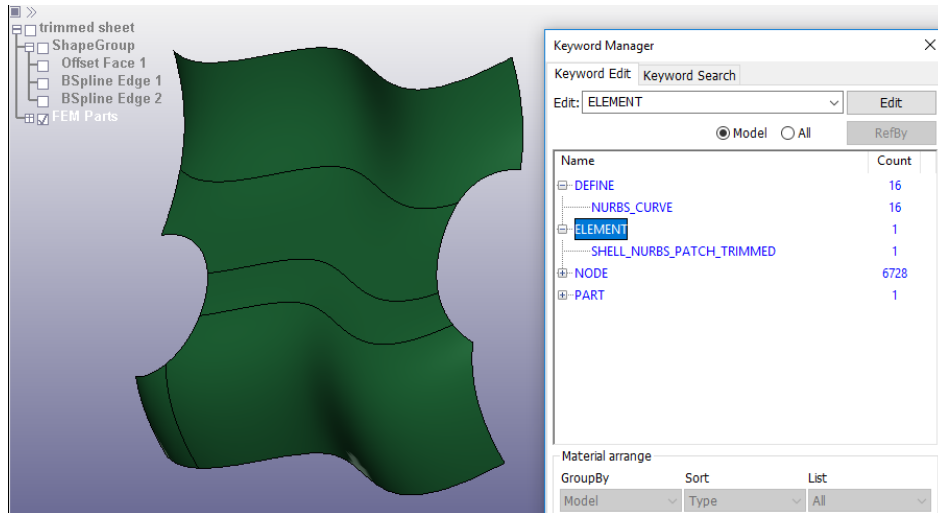


Figure 4-8: Example of trimmed NURBS patch. The geometry imported from the CAD is already trimmed, the element created by the editor is a `SHELL_NURBS_PATCH_TRIMMED` and 16 `NURBS_CURVE` have been automatically defined (used as trimming loops).

Anyway, if needed, the user is free to trim a shell NURBS element by the keyword input form. NURBS curves to be used as trimming loops must be defined by means of the keyword `DEFINE_NURBS_CURVE`.

Trimming loops must then be added in `ELEMENT_SHELL_NURBS_PATCH_TRIMMED` keyword.

4.2. Solid nurbs elements

To create solid NURBS elements, `ELEMENT_SOLID_NURBS_PATCH` is the keyword needed.

For simple shapes, similarly to the shell case, this operation can be manually done from the keyword input form, with free choice for what concerns knot spans and polynomial orders. This is not practical and not suited for complex solid geometries.

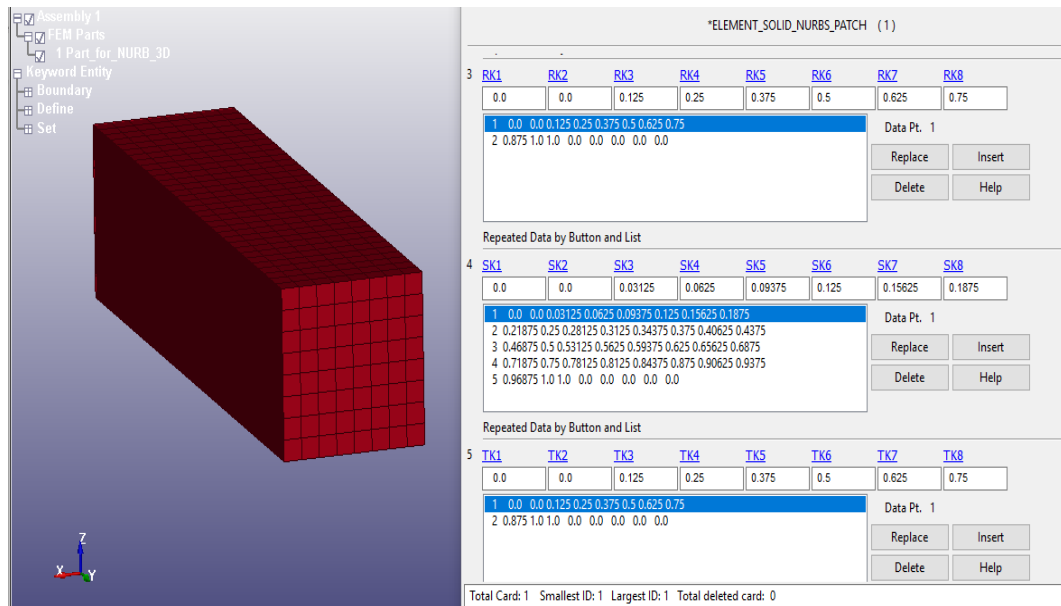


Figure 4-9: simple parallelepiped NURBS element created from the keyword input form, on the right the 3 knot vectors variables.

Also in case of solid NURBS elements, in LS-PrePost a 3D NURBS editor is available, with the purpose to create, refine and modify solid NURBS elements from CAD imported geometries.

In this case, anyway, there are problems in the creation process. After many trials it turned out that is better to import only the basic generation geometries of the solid that we want to analyze (surfaces and lines) and use the 3D solid editor of LS-PrePost to create the final solid shape .

4.2.1 NURBS solid elements creation with 3D NURBS editor

After the definition of the Solid geometry, by means of LS-PrePost 3D functions (example in figure 4-10), the 3D NURBS editor is used to create an ELEMENT_SOLID_NURBS_PATCH (figure 4-11).

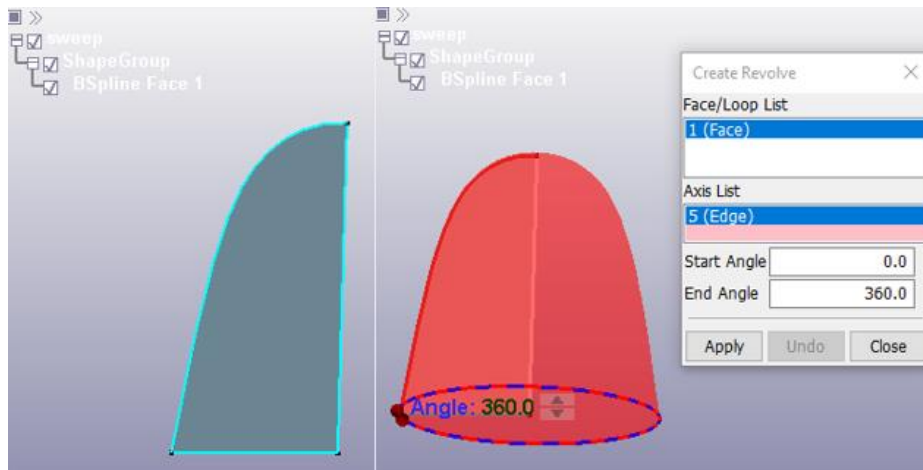


Figure 4-10: CAD imported basic surface (left), revolution solid created in LS-PrePost (right)

The creation is easy and immediate after the selection of the solid body faces. However, the stability of the tool is probably not there yet to deal with all the types of geometries, infact in many situations the creation process doesn't work at all.

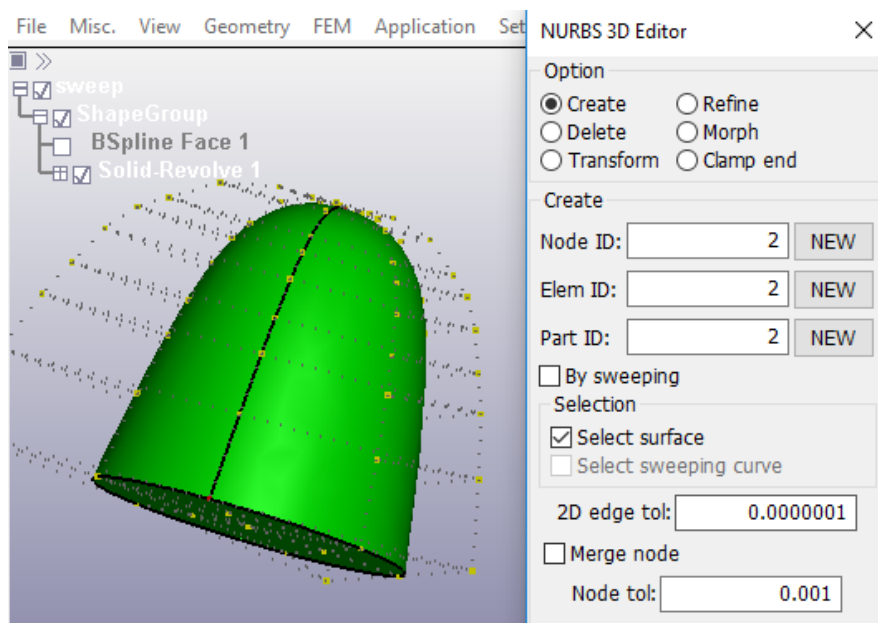


Figure 4-11: Solid NURBS element creation with the 3D NURBS editor

The “by sweeping” command, allows the generation of a swept solid NURBS patch starting from a surface and a sweeping curve.

Anyway, also in this case, the software seems to have problems in the recognition of the basis sweeping surface, as shown in Fig. 5-11. The refinement is not a solution.

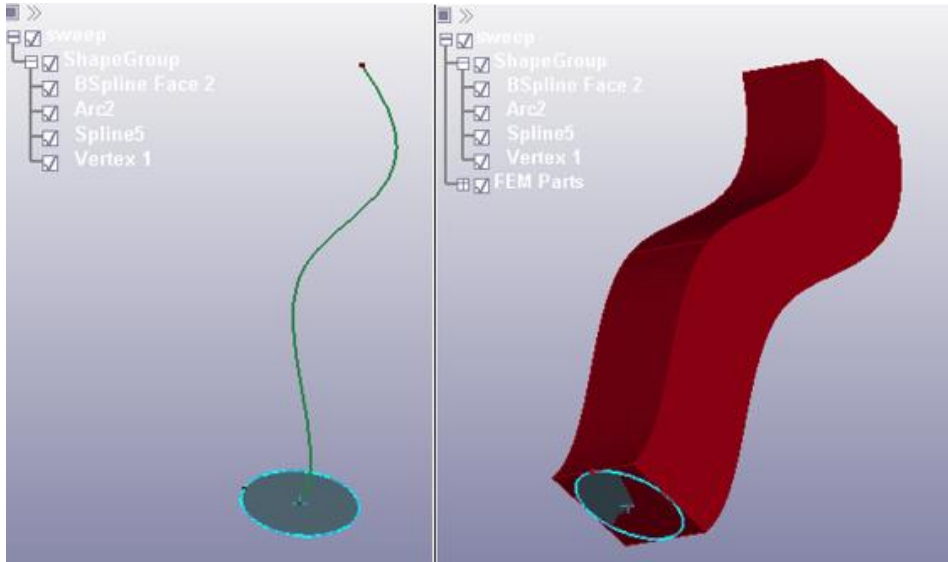


Figure 4-12: Sweeping surface and curve (left), solid NURBS element created with the editor (right)

To obtain swept solids the normal 3D editor of LS-PrePost could be used before.

4.2.2 Refinement

As in the shell element case, H-P and K-refinement are available. The functionality is the same with the addition of the T knot vector.

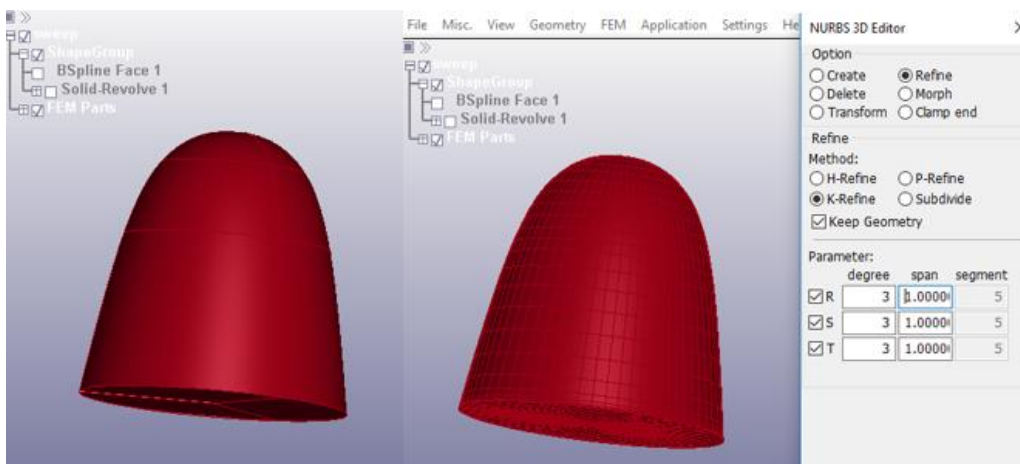


Figure 4-13: Solid bullet model, before and after refinement

4.3. Analysis performance

As a first approach, some simple isogeometric analyses are shown. An hemispheric bullet (FE shell) impacts on a plate (Isogeometric element). Three different versions of the plate are presented: the first is flat, the second with bulges and the third with the addition of trimming loops.

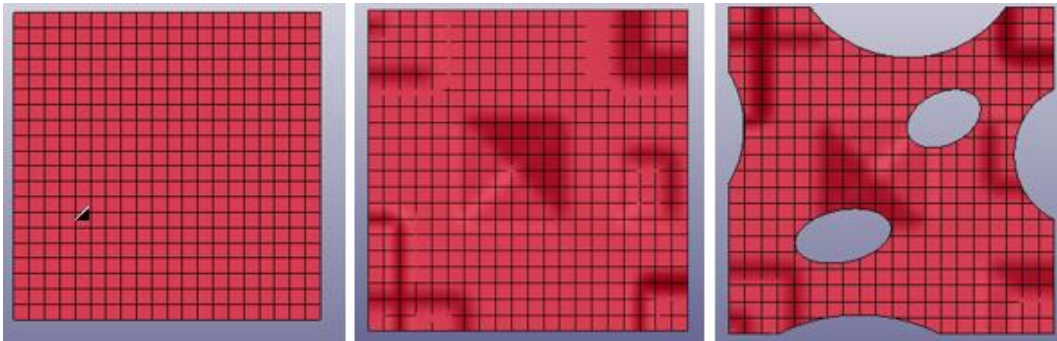


Figure 4-14: The three versions of the plate.

The bulges are created changing the position of some of the nodes of the control grid, manually from the nodes keyword or by means of the command “control grid” in the nurbs editor.

4.3.1 Model set up

The set up of this kind of analysis is simple and similar to a normal FE analysis.

The **SECTION_SHELL** that will be assigned to the NURBS part must have **ELFORM=201**, that defines isogeometric shell elements.

The main difference to consider, is that in IGA the nodes are part of the control grid, and not included among the mesh elements. This is important for the definition of boundary conditions (SPCs, loads, velocity ecc.), and the accuracy depends on geometry and refinement (In these simple examples this is negligible). However, in case of need the keyword **CONSTRAINED_NODE_TO_NURBS_PATCH** is available as already explained.

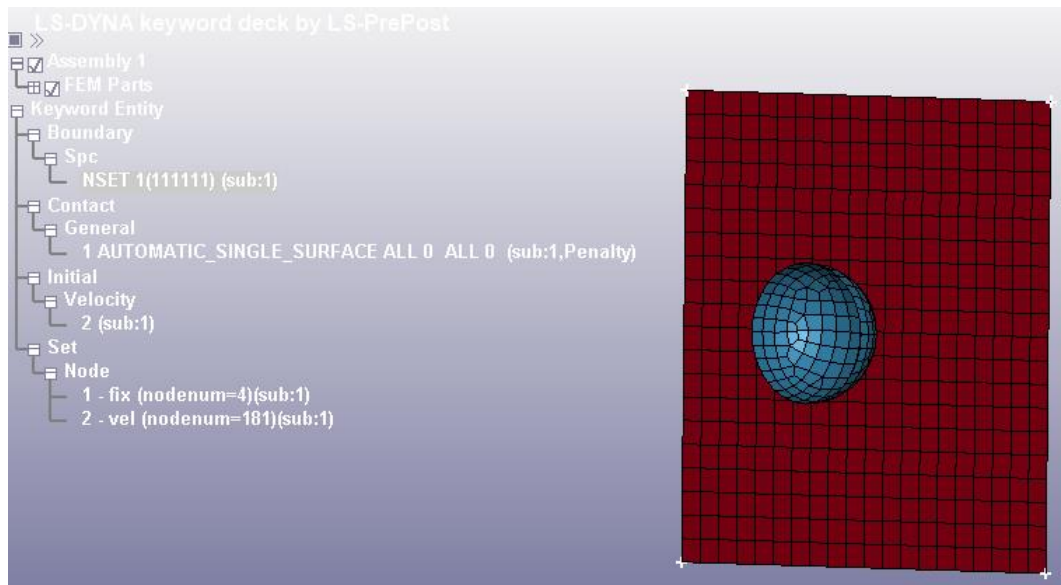


Figure 4-15: Simple IGA model, impact of an emispheric bullet on a plate

The plate has four fixed constraints at the four corners, the bullet impacts with a constant velocity of 5 m/s and an AUTOMATIC_SINGLE_SURFACE contact is defined including all surfaces.

A MAT 003 (PLASTIC-KINEMATIC) is defined for the plate and a MAT 001 (ELASTIC) is assigned to the bullet, both steel. The thickness assigned to the plate is 0.5 mm.

4.3.2 Run

The purpose of this first examples is to test the stability of simple IGA performed with LS-DYNA and check the pre/post processing capabilities of LS-PrePost, so there won't be a comparative with FEA.

The first point to underline, in this case, is that IGA functions are still in an initial stage and are being improved year by year by LS-DYNA developers, so to have the possibility to run all the models, **the latest version of the software is needed**.

Right now, the latest build available is **LS-DYNA 11.1.0**. Some simulations have been carried out with the 11.0.0 version and for instance any model with trimmed nurbs elements didn't run.

4.3.3 Post-processing

The first thing noticed in the post processing environment is the presence of two nurbs element instead of one in the components tree.

The first one (red plate in Fig. 5-16), is the NURBS patch itself, while the second one is the interpolated mesh, that is generated with an igaplot file and is stored in the regular d3plot file.

The interpolated mesh is useful from the user point of view because many fringe data can only be processed with it for now. (The NURBS patch, if trimmed, presents graphical issues during the animations or contour plots as shown in Fig. 4-17).

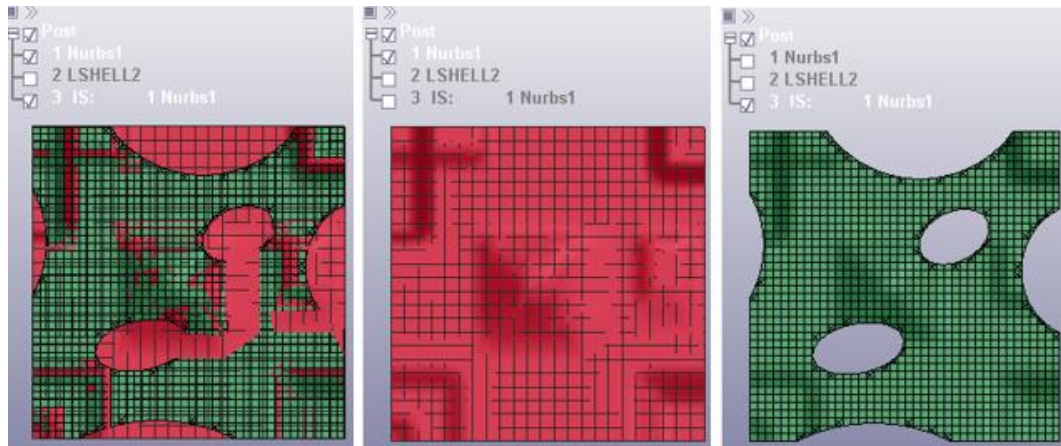


Figure 4-16: From left to right: Both geometries are turned on, only NURBS patch turned on(red), interpolated mesh turned on (green).

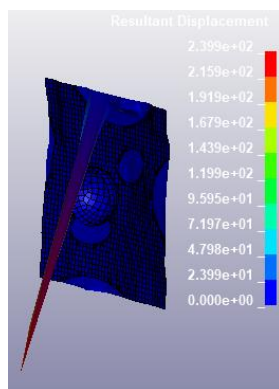


Figure 4-17: Displacement contour plot in case the NURBS patch is not turned off.

However, with the correct visualization, the results for all the 3 examples are satisfactory. The initial states are represented in fig. 4-18 , than the maximum displacements are plotted in fig. 4-19 and they present the expected trend: the bulged plate is slightly stiffer than the original one that, in turn, is slightly stiffer than the bulged/trimmed one.

The maximum stresses in the plates are plotted in Fig. 5-20. They are located at the four corners, where the constraints are placed, with a distribution coherent with the bulges location (in the first case it is homogeneous).

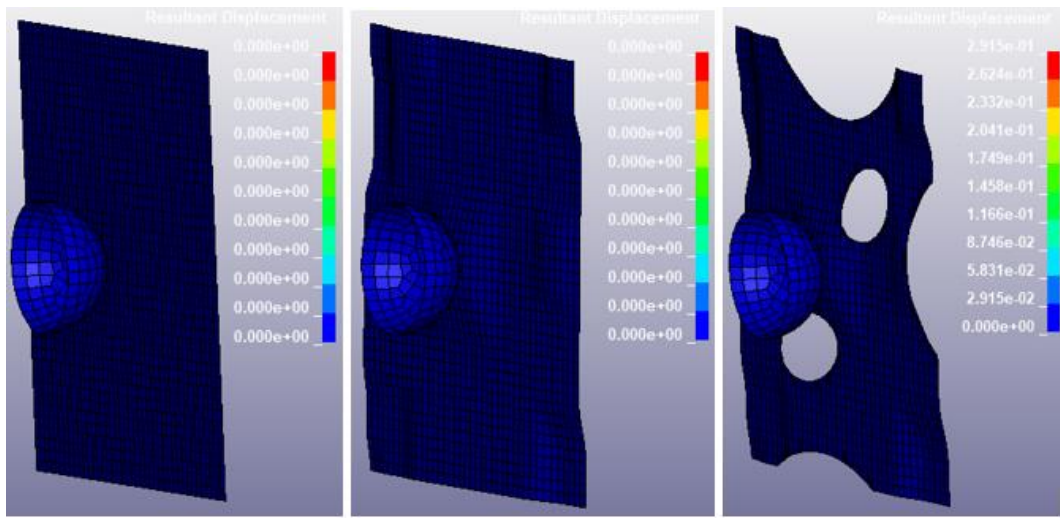


Figure 4-18: The three models are represented at time $t=0$.

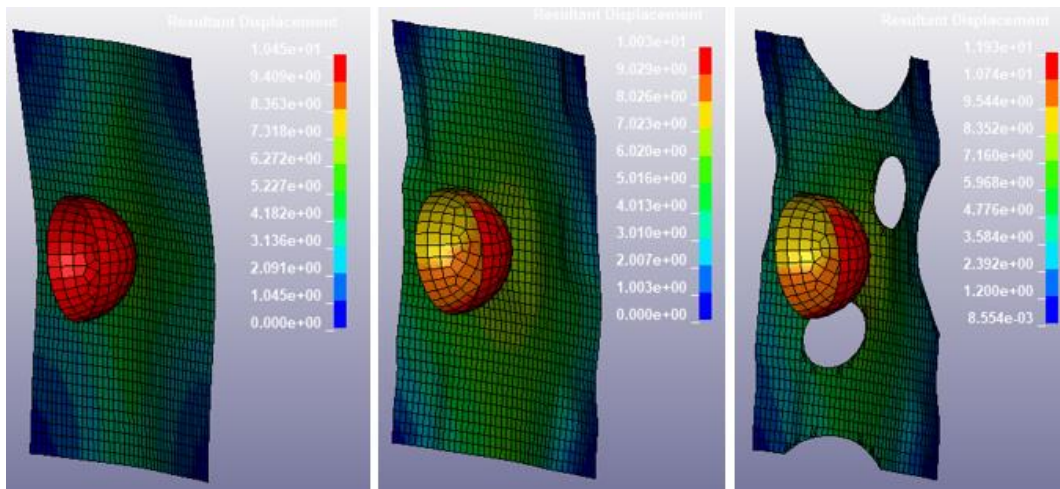


Figure 4-19: Maximum displacements of the plates during the impacts.

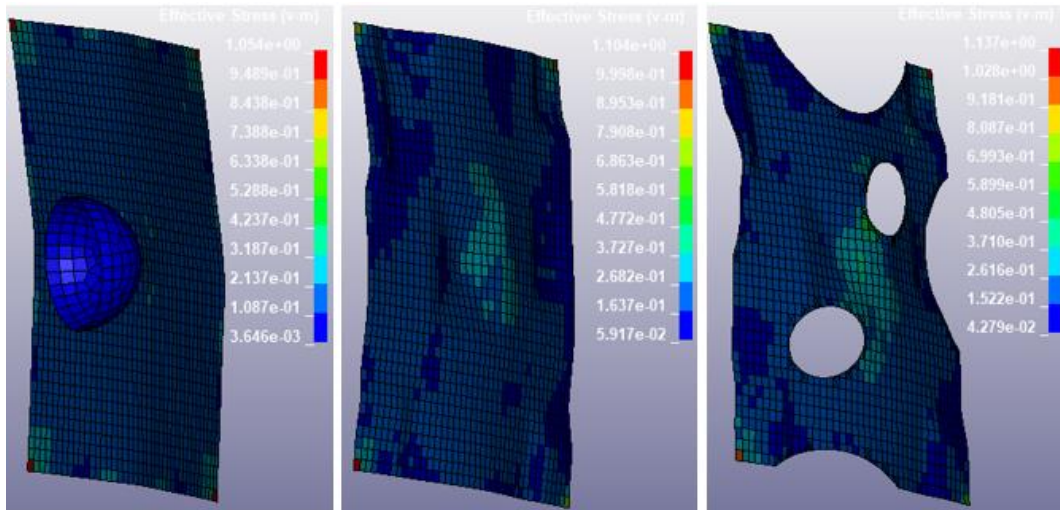


Figure 4-20: Maximum stresses suffered by the plates.

The plots of the total kinetic energy over time are shown in the figures below. Results are as expected in the first and second models, while in the last trimmed one the graph is quite unreal.

This is probably due to the incorrect behaviour of the trimmed parts. This issue will be encountered also in the next chapters. (see Fig. 5-17).

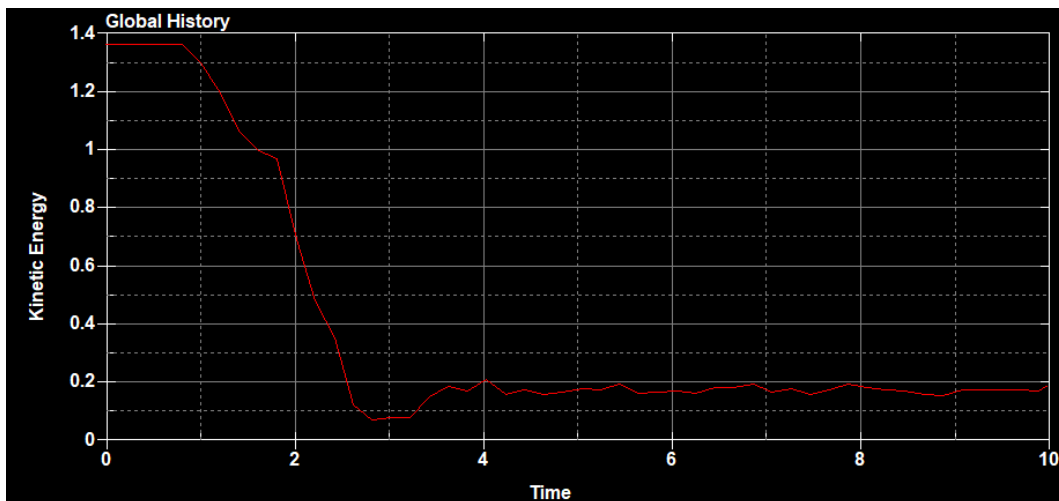


Figure 4-21: Kinetic energy plot: flat plate.

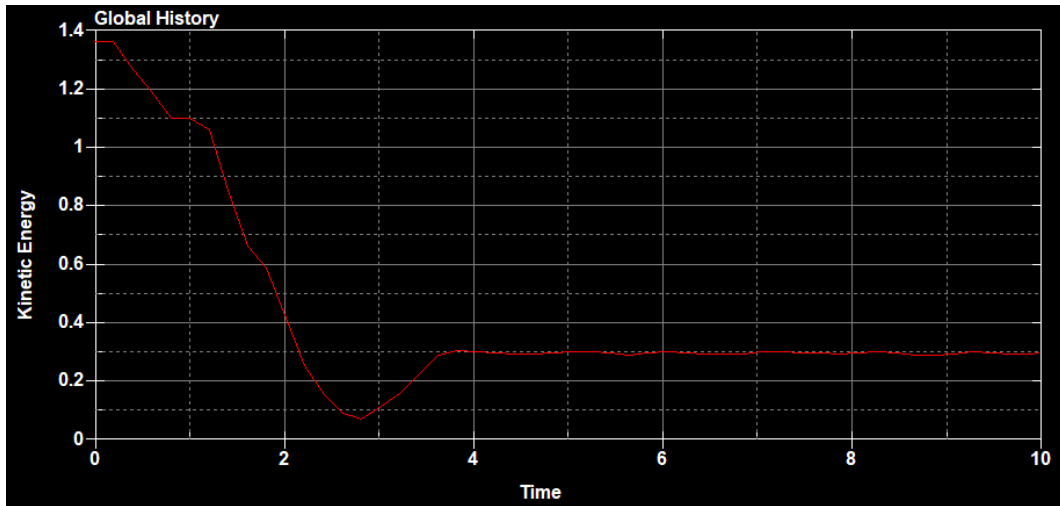


Figure 4-22: Kinetic energy plot: bulged plate.

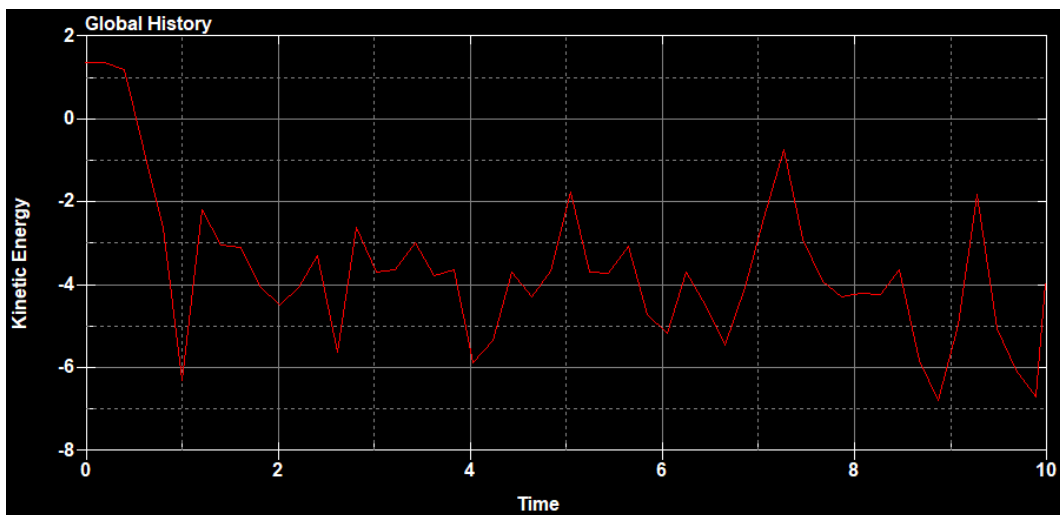


Figure 4-23: Kinetic energy plot: bulged/trimmed plate.

To obtain the real energy data separate parts plot must be produced in order to isolate the error.

The three plots below are, in order: kinetic energy of the bullet, internal energy of the plate and kinetic energy of the plate. The problems are clearly located in the last one, anyway all the data needed can be obtained with this strategy.

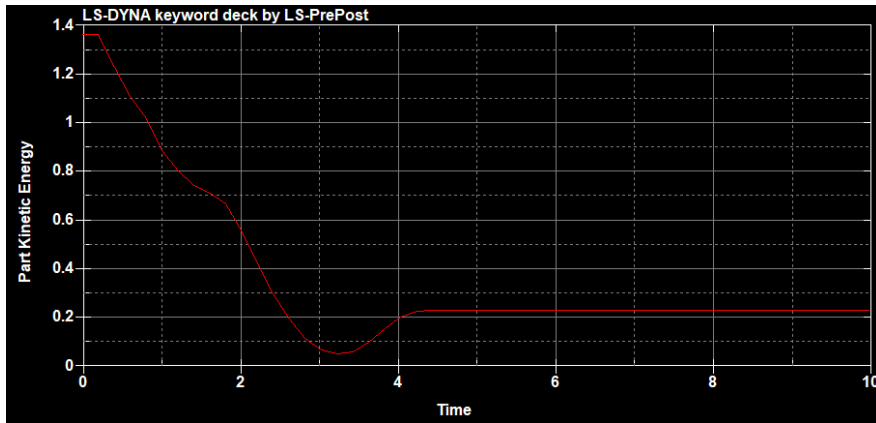


Figure 4-24: Kinetic energy of the bullet

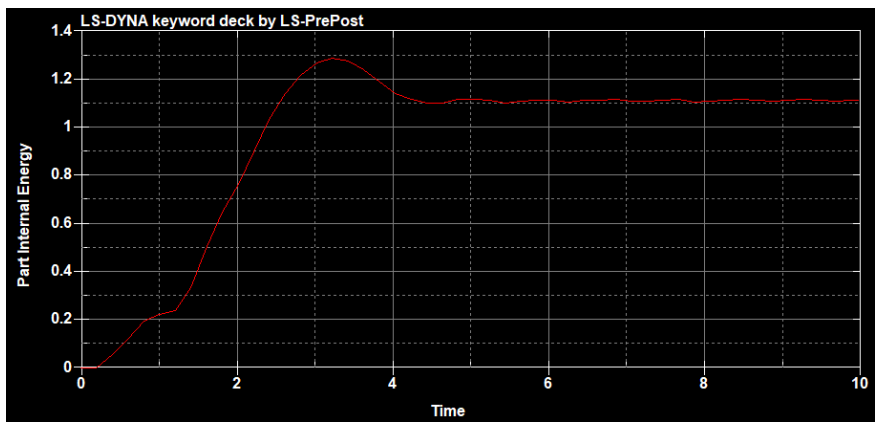


Figure 4-25: Internal energy of the plate

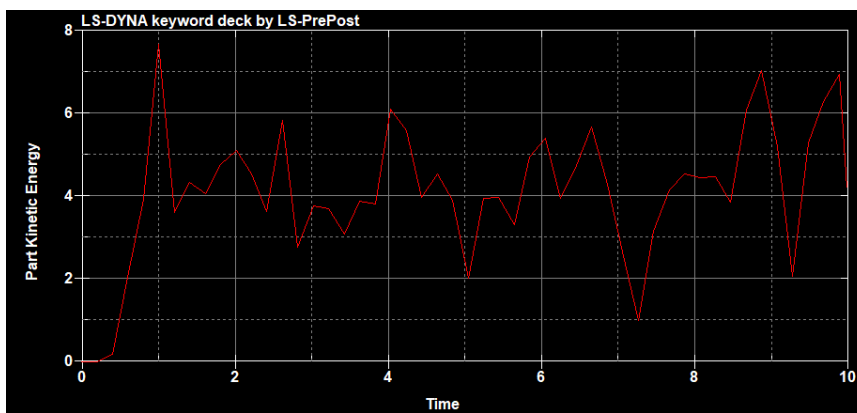


Figure 4-26: Kinetic energy of the plate.

5. IGA PERFORMANCE

The purpose of this chapter will be the discovery of the IGA limits and capabilities of the software in case of more complex crash simulations. The results will be compared to standard FEA.

5.1. Multi-patch simple crashbox impact

5.1.1 Set up

The first model considered consists in the impact of a rigid barrier on a square tube crashbox.

The impactor consists in a simple FE block normally meshed, with a speed of 8 m/s. The material assigned is a steel - MAT 24 (RIGID).

The crashbox is composed by 4 shell NURBS patch elements, 1 mm thick, with a fixed constraint on one side. The material assigned is a steel modelled as MAT 024 (PIECEWISE LINEAR PLASTICITY).

An AUTOMATIC_SINGLE_SURFACE contact is defined for the crashbox, while an AUTOMATIC_NODE_TO_SURFACE contact is used to model the interface of the two parts, both contacts have non-zero friction coefficients.

In the IGA model 4 NURBS_TIED_EDGE_TO_EDGE contact are generated in order to connect the four patches.

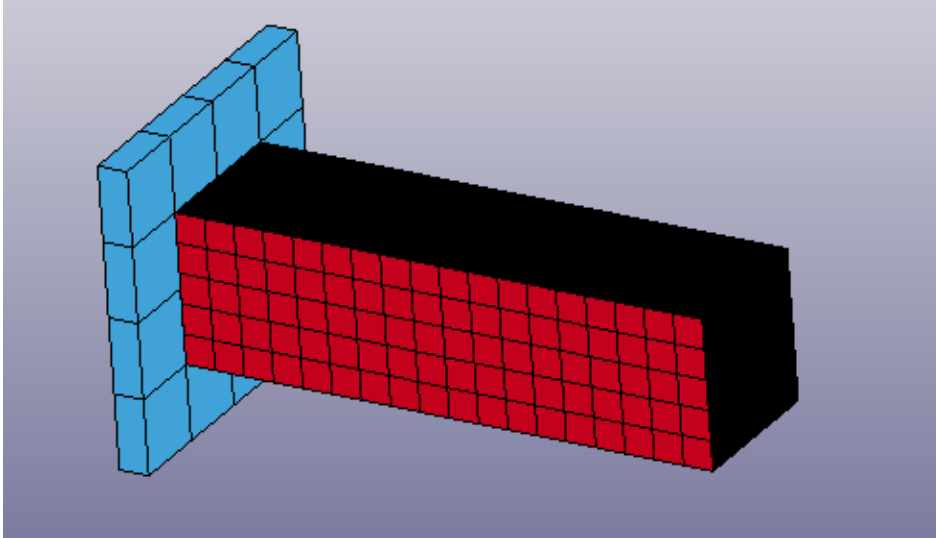


Figure 5-1: Impactor (blue) and crashbox (red)

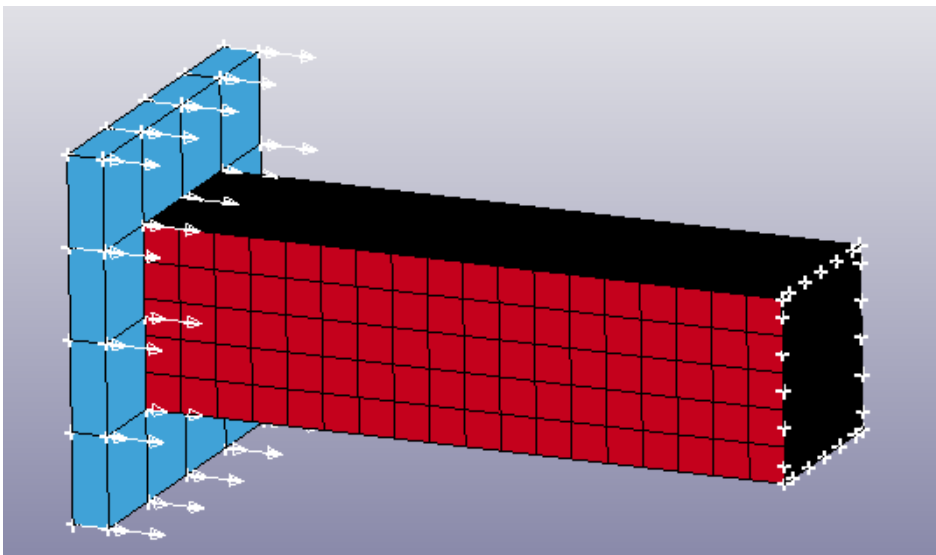


Figure 5-2: Model boundary conditions

5.1.2 Results

Nine different simulations have been carried out, three classic FE analysis, three IGA with quadratic polynomial degree in both directions, and three IGA with cubic polynomial degree in both directions. All the analyses are built with comparable mesh sizes (20, 10, 5mm).

	Max. crashbox displacement at t=30ms	Runtime (NCPU=2)
FEM (20mm)	8.56mm	15 seconds
FEM (10mm)	10.58mm	1 minute 54 seconds
FEM (5mm)	12.26mm	20 minutes 15 seconds
IGA(quadratic, 20mm)	6.05mm	24 seconds
IGA(quadratic, 10mm)	8.13mm	1 minute 52 seconds
IGA(quadratic, 5mm)	11.48mm	1 hour 18 minutes
IGA(cubic, 20mm)	10.98mm	1 minute 48 seconds
IGA(cubic, 10mm)	12.01mm	54 minutes 7 seconds
IGA(cubic, 5mm)	12.70mm	4 hours 11 minutes

The maximum displacement of the crashboxes clearly converges towards a value that's major than 12mm. This underlines the weakness of quadratic IGA and the importance of all the types of refinement. Moreover, in the kinetic energy plot (fig. 5-7), the trend of the blue curves (the three cubic IGA) is regular and close to the others even in case of coarse mesh (20mm), while the green curves (quadratic IGA) reach a satisfactory result only in case of fine mesh (5mm). FEA needs a minimum of refinement to converge to the right trend, anyway the result are satisfactory because of the low runtimes (the maximum displacement of 5mm cubic IGA is probably more precise, but the difference with fine meshed FEA does not justify the runtime, that is twelve times higher).

In general, the higher runtimes with respect to FEA, suggest that the accuracy of the result and the complexity of the model are affected by the presence of the contacts between patches.

In the figures below the displacements contour plots at time $t=30$ are shown. As stated in chapter 4, many fringe data are available only for the interpolated mesh elements, so this graphic visualization will be used. An example of not interpolated NURBS patches deformation is shown in fig. 5-3.

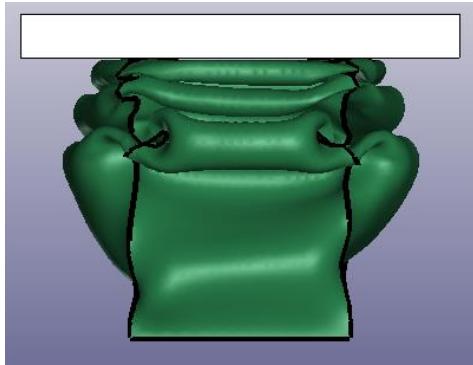


Figure 5-3: NURBS patch deformation in the post processor.

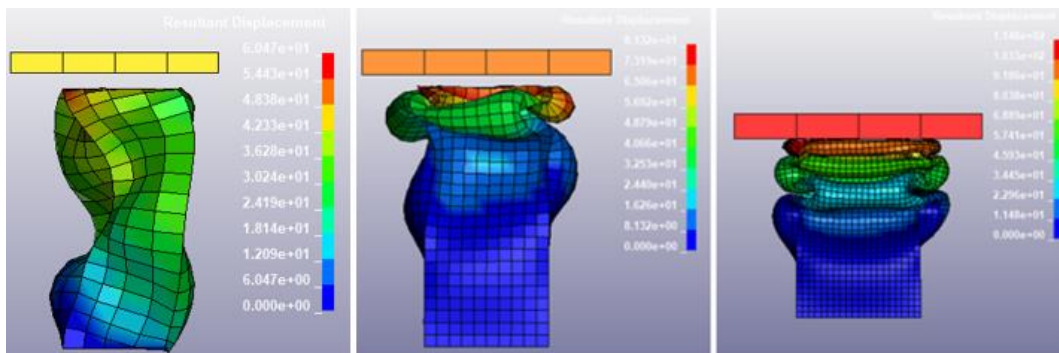


Figure 5-4: IGA- quadratic patch - Resultant displacement at time $t=30$ ms. Mesh size decreases from left to right: 20, 10, 5 mm.

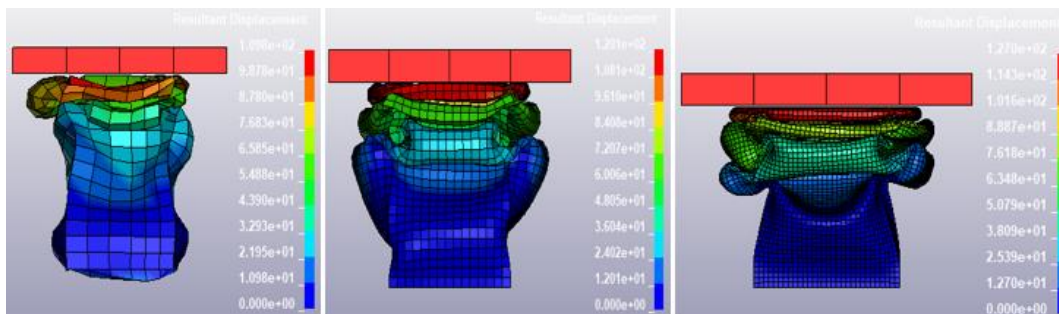


Figure 5-5: IGA- cubic patch - Resultant displacement at time $t=30$ ms. Mesh size decreases from left to right: 20, 10, 5 mm.

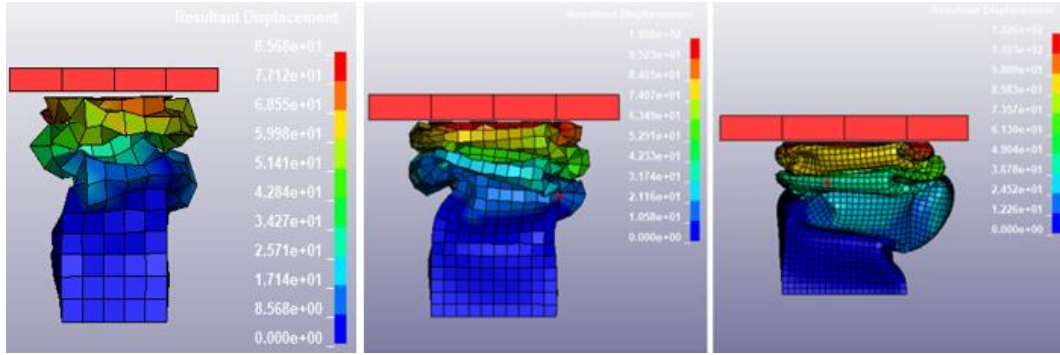


Figure 5-6: FEM - Resultant displacement at time $t=30\text{ms}$. Mesh size decreases from left to right: 20, 10, 5 mm.

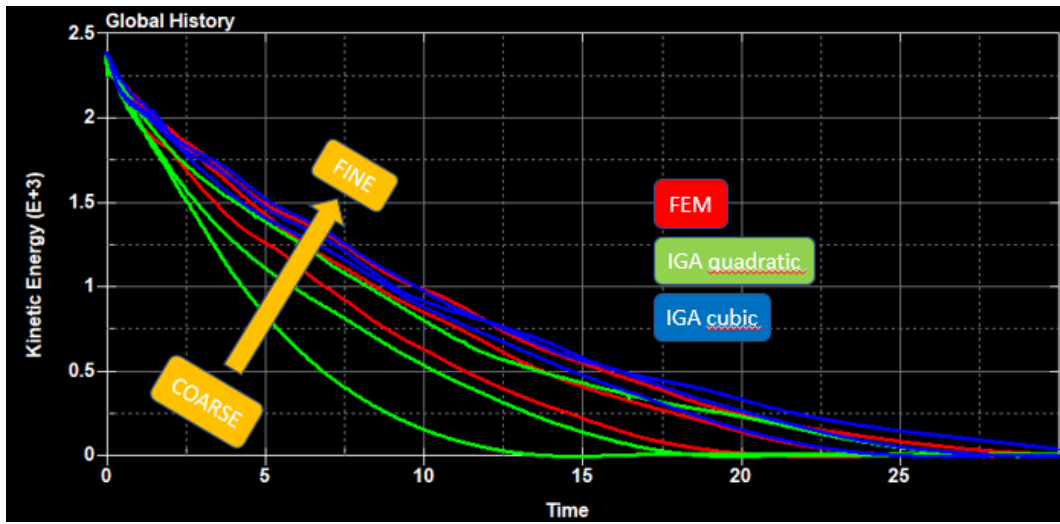


Figure 5-7: Kinetic energy plot for all the models. IGA quadratic (green), IGA cubic (blue) and FEM (red), mesh size from coarse to fine in the direction of the arrow.

5.2. Spot welded s-rail impact analysis

5.2.1 Set up

Spotweld modelling is available for IGA in LSDYNA. An 's' shaped crashbox impact on a rigid wall is analyzed.

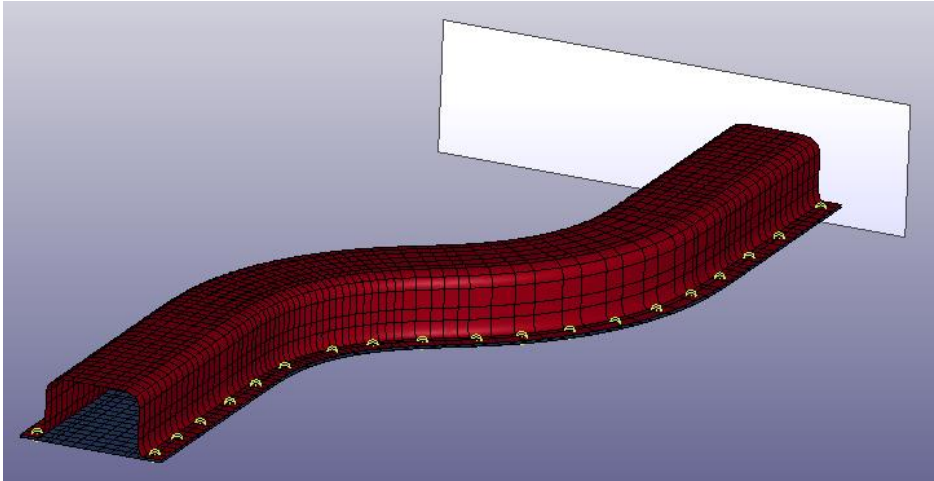


Figure 5-8: Spotwelded s-rail isogeometric model.

The spotwelds are modelled as beam elements, whose nodes are connected to the NURBS patches by means of a TIED_SHELL_EDGE_TO_SURFACE_BEAM_OFFSET contact. Another effective way to do this is to use a CONSTRAINED_NODE_TO_NURBS_PATCH_SET card. With this method the results obtained don't change, while trials performed using a common SPOTWELD contact produced problems with IGA.

To the part composed by the beam elements, A MAT 100 (SPOTWELD) is assigned together with the card SECTION_BEAM (where ELFORM=9 must be selected).

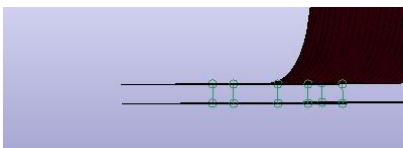


Figure 5-9: Particular of the spotweld beam elements

Analysis were performed with three different mesh sizes (20, 10, 5 mm). The NURBS patch polynomial degree is chosen to be cubic in both directions, further P-refinement would increase exponentially the runtime (from minutes to hours).

The effect of the mesh size on the geometry is shown in the figure below. In FE method, the resolution is very low in the case of a 20mm elements size, and this can cause inaccuracies during the analyses.

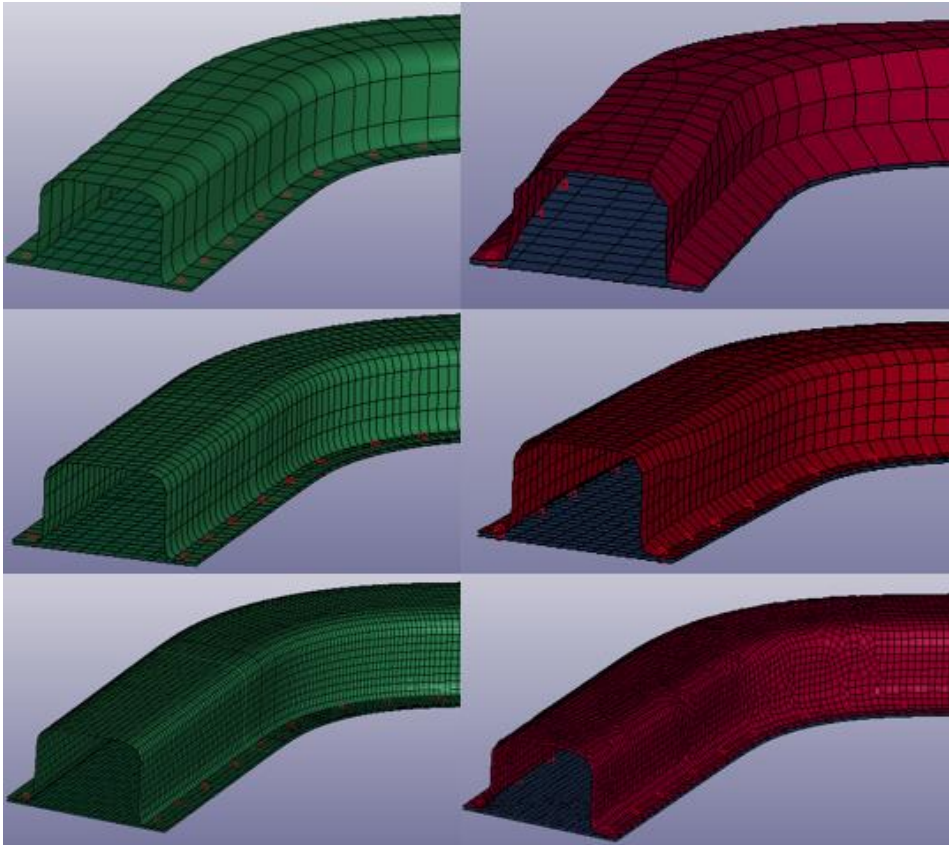


Figure 5-10: Three different levels of refinement (20, 10, 5 mm) in IG model (left) and FE model (right).

5.2.2 Results

The runtime of the Isogeometric analyses, also in this case, turned out to be higher than the FE ones. This is probably due to the polynomial degree of the NURBS patch elements (3, 3), that is anyway the minimum needed in order to have a good interpolation avoiding tiny knot vector spans.

ANALYSIS	RUNTIME (4 cpu)
FE 5mm	8 minutes 12 seconds
FE 10mm	52 seconds
FE 20mm	16 seconds
IGA (cubic, 5 mm)	3 hours 29 minutes
IGA(cubic, 10 mm)	21 minutes 8 seconds
IGA (cubic, 20 mm)	3 minutes 6 seconds

To focus on the spotweld elements failure, the behaviour of the two methods is similar, with some little differences (highlighted in the figures below), due to the different distribution of the reaction forces along the part because of the diversity of the geometries.

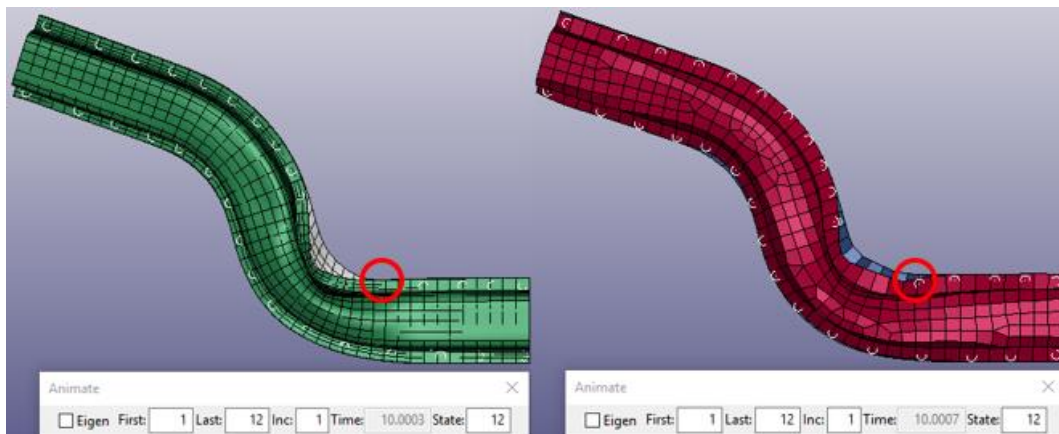


Figure 5-11: Spotwelds failure at time $t=10$ ms, mesh size 20mm, IGA (left) and FEM (right).

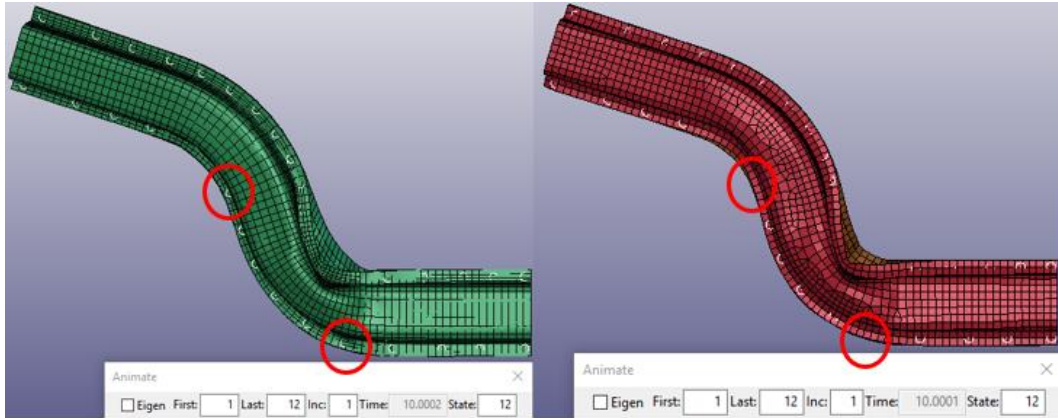


Figure 5-12: Spotwelds failure at time $t=10$ ms, mesh size 10mm, IGA (left) and FEM (right).

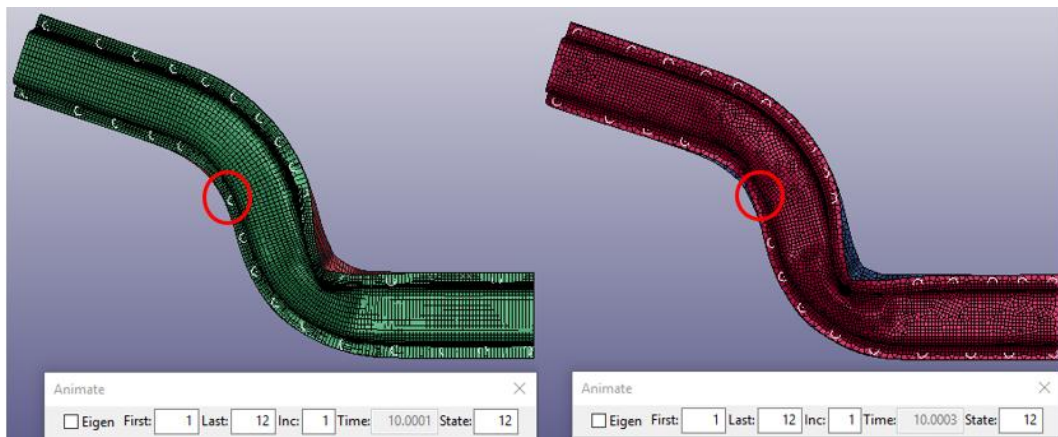


Figure 5-13: Spotwelds failure at time $t=10$ ms, mesh size 5mm, IGA (left) and FEM (right).

The kinetic energy dissipations converge to two slightly different values (4650 J for the FE model and 4875 J for the IG one). Anyway, known the geometry inaccuracies due to the FE discretization, the translation of the result between the two methods is coherent with the effect of the mesh refinement on the FE model plot. (details in Fig. 5-14) The accuracy of this isogeometric model, if compared to the previous one (squared crashbox), clearly underlines the better performance produced in case of single patch body with respect to a multi patch one.

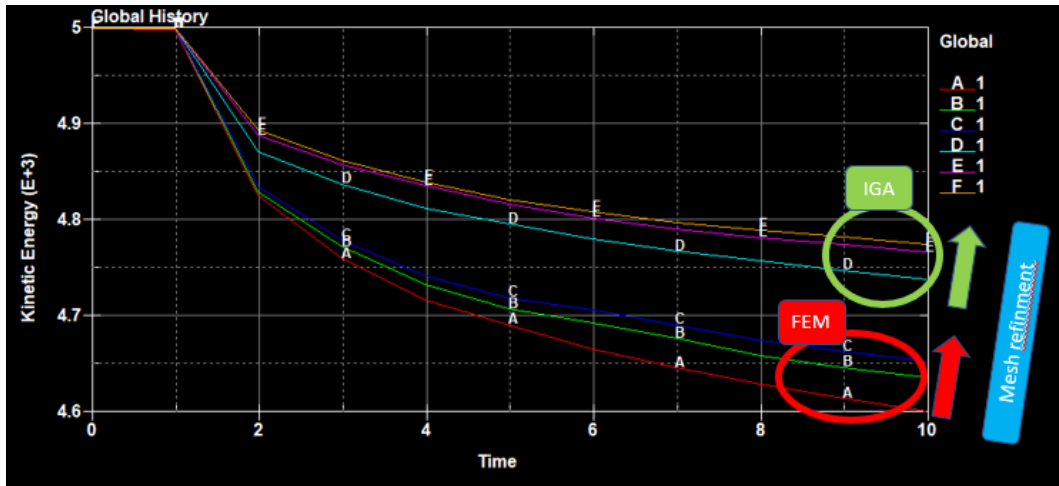


Figure 5-14: Kinetic energy plot. A, B, C curves represent the FE model, from coarse to fine mesh, D, E, F curves represent IG model from coarse to fine mesh.

5.2.3 Issues

Some problems have been found in spotwelds modelling, the usage of a SPOTWELD contact caused a wrong behaviour of the beam elements. (Fig. 5-15)

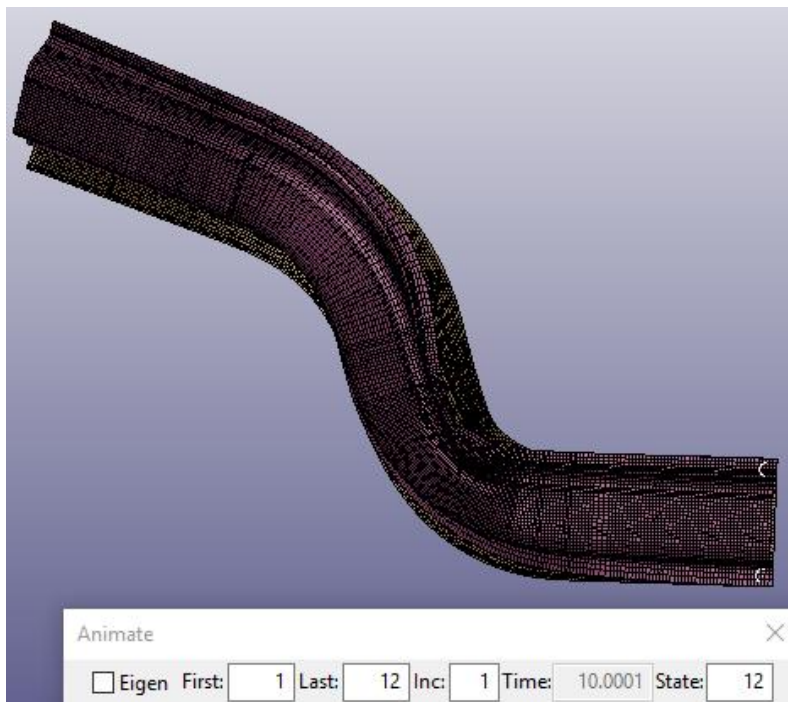


Figure 5-15: Spotwelds failure at time $t=10\text{ms}$, only two beam elements didn't fail. In this model a SPOTWELD contact is used.

5.3. Analysis of complex models

More intricate crash elements have been modelled in order to test the limits of IGA. The impact on a rigid wall of the geometry shown in fig. 5-14, composed by a bumper and two crashboxes, has been assessed. After IG modelling, the structure consisted in 141 NURBS patches.

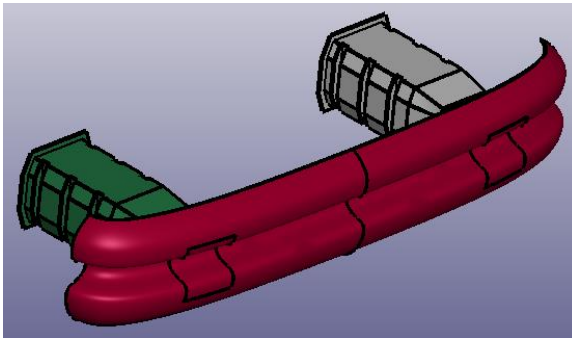


Figure 5-16: Automotive crash structure.

No problems were encountered during the NURBS creation and editing phase, while many issues regarding the boundary conditions between the patches have been found while running the solution.

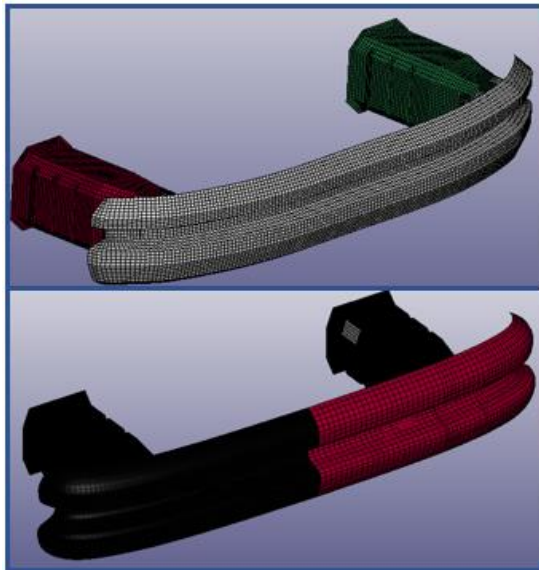


Figure 5-17: FE model (up), IG model (low).


```

*** Error 20211 (STR+211)
    reading *CONTACT_NURBS
    undefined node # 2

    input phase will continue if possible

*** Error 20211 (STR+211)
    reading *CONTACT_NURBS
    undefined node # 3

    input phase will continue if possible

```

Figure 5-18: Example of NURBS contact errors.

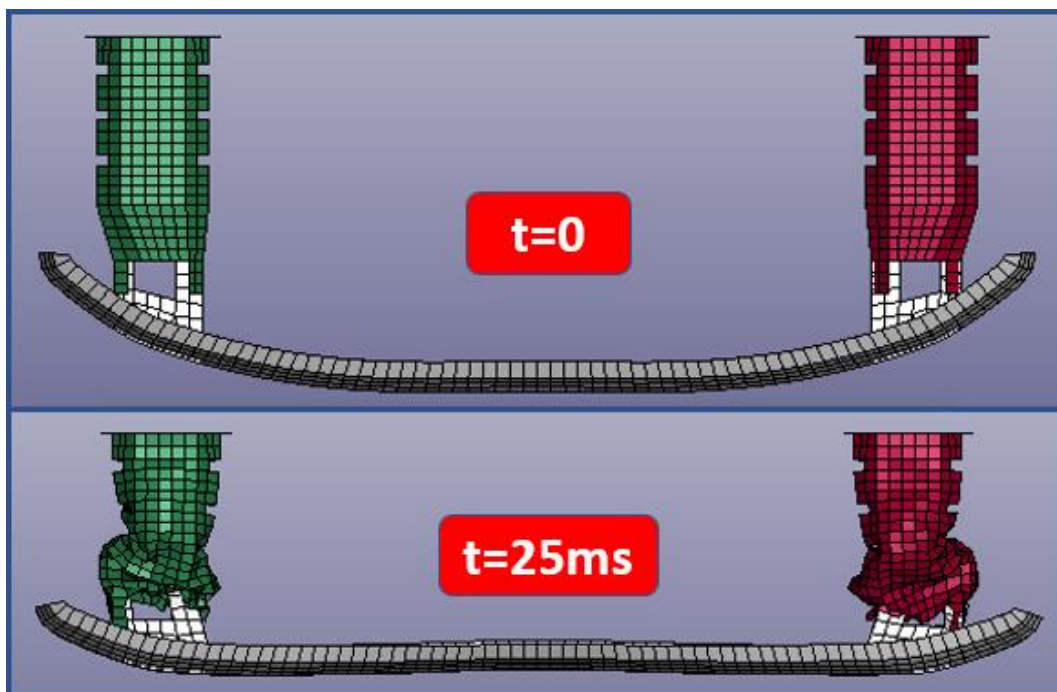


Figure 5-19: FEM solution.

An analysis of a simplified version of the same model has been performed in order to better understand the limits.

In this case, the whole model is composed by 14 NURBS patches (8 of which are trimmed), and the impact arrives with a non-zero angle. Anyway, some problems during the generation of the patch shown in figure 5-22 were spotted (on both car

sides): in the CAD file a non-open knot vector was used, but LS-PrePost somehow translated it to an open-knot vector with a non properly trimmed result.

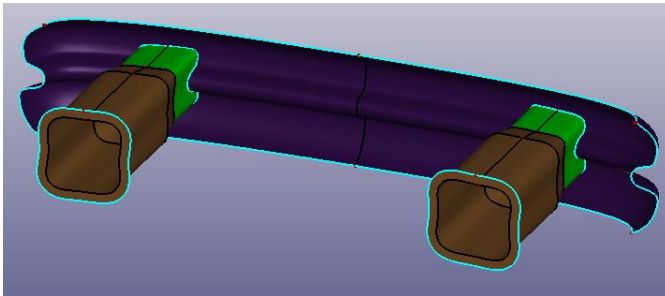


Figure 5-20: Simplified geometry.

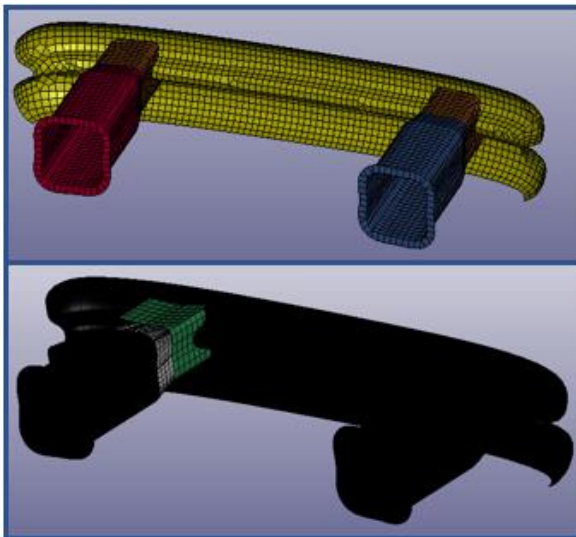


Figure 5-21: FE model (up), IG model (down).

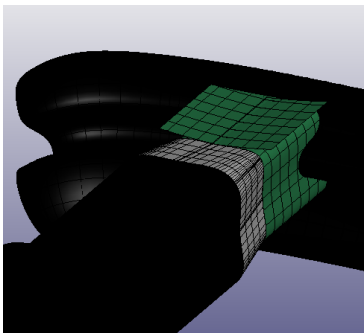


Figure 5-22: Particular of the badly trimmed patch (the problem is spotted on both crashboxes).

Moreover, the results produced are unsatisfactory, with graphic and mechanical issues generated by the trimmed NURBS patches and confirmed by the kinetic energy profile, where some impossible energy fluctuations can be spotted.

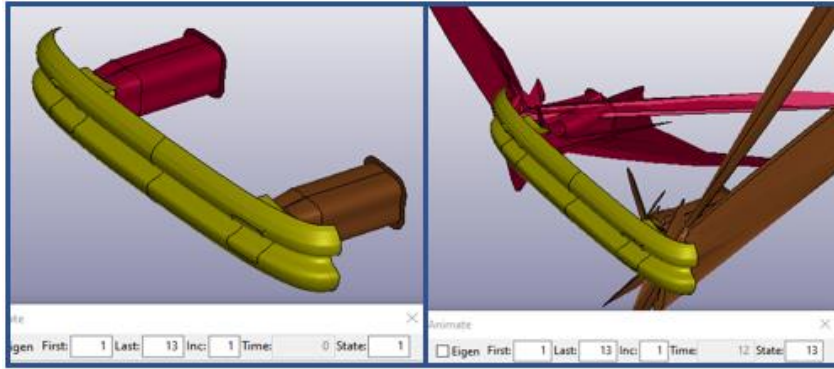


Figure 5-23: IGA solution.

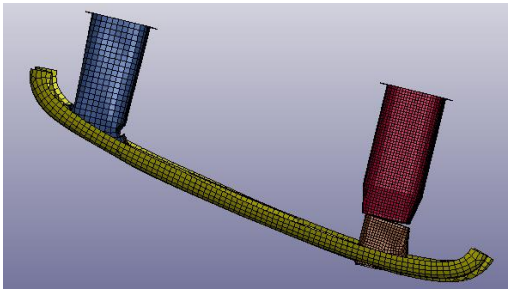


Figure 5-24: FEM solution.

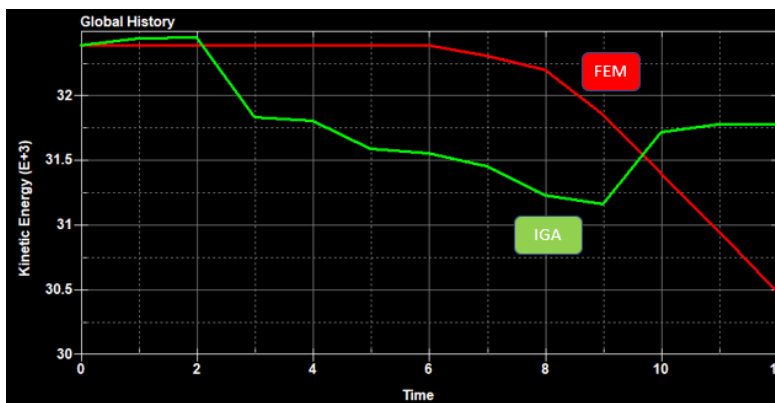


Figure 5-25: Particular of the kinetic energy plot. FEM (red) and IGA (green).

5.4. Implicit solver and eigenvalue analysis

Implicit solution of isogeometric analyses is supported by LS-DYNA. To do so the activation of the implicit solver, from CONTROL_IMPLICIT_GENERAL card, is needed (IMFLAG must be set equal to 1).

In the following example a foil with “L” profile is excited with a sinusoidal load. In Fig. 5-14, CONTROL_IMPLICIT_GENERAL card is shown, IMFLAG value is set to 1 and the initial time step size for implicit analysis DT0 is also selected.

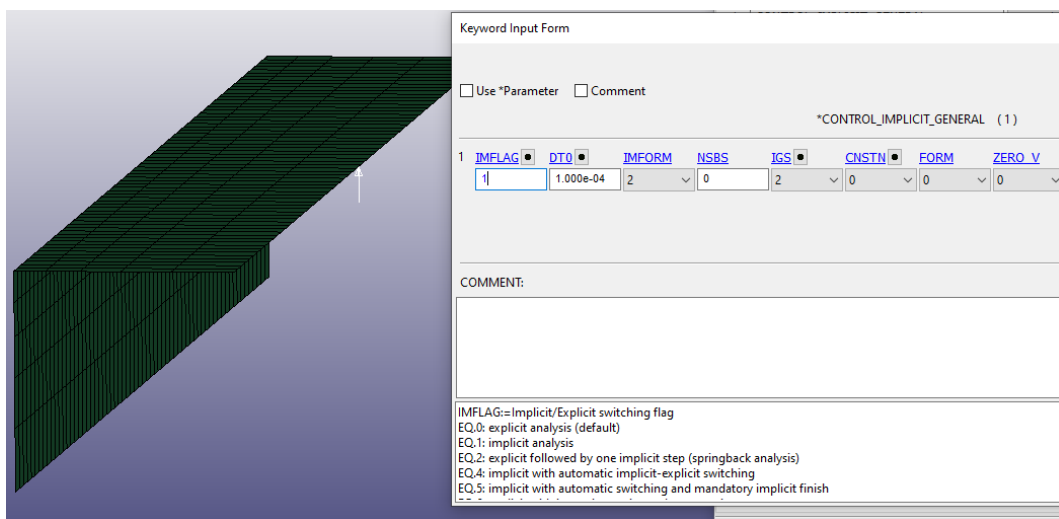


Figure 5-26: Set up of the implicit solution via CONTROL_IMPLICIT_GENERAL card.

After a certain number of iterations (in this case the timestep size is not automatically set), the solution is reached without problems. An explicit analysis is also performed in order to make a comparison.

	RUNTIME (4CPUs)
IMPLICIT	4 minutes 40 seconds
EXPLICIT	8 minutes 26 seconds

The explicit simulation takes almost the double of the time. Anyway, the results are coherent with some differences close to the peak values, probably due to the low accuracy of the implicit model.

in the following graphs stresses and strain in the excited point are plotted over time. Contour plot of the Von Mises stress, in the moment of maximum load is shown in Fig. 5-15.

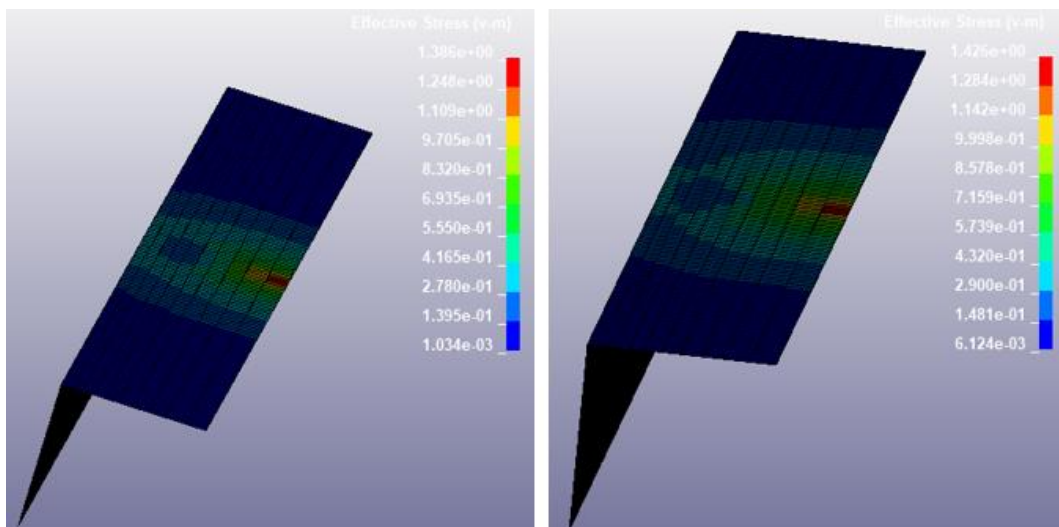


Figure 5-27: Stress distribution in the instant of max stress. Implicit (left) and explicit (right).

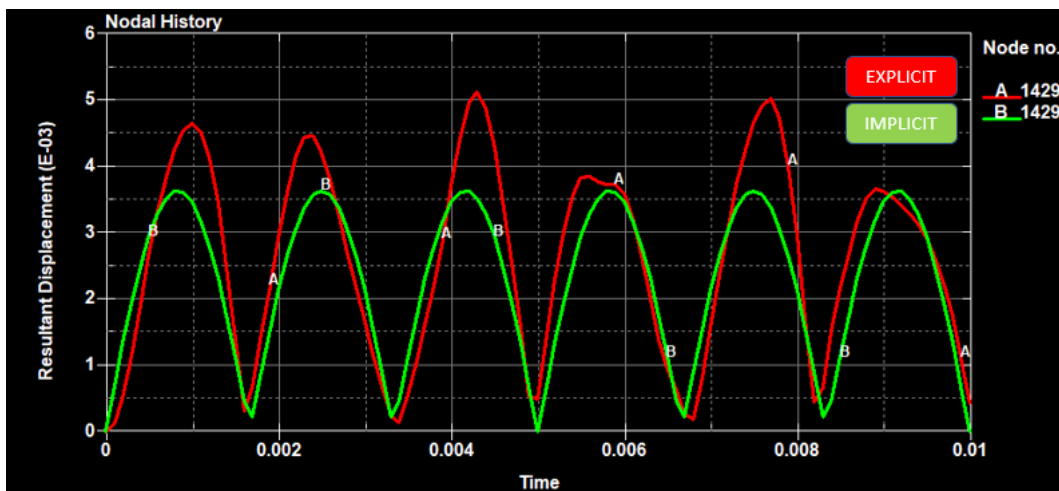


Figure 5-28: Displacement of the excited node. Explicit (red) and implicit (green).

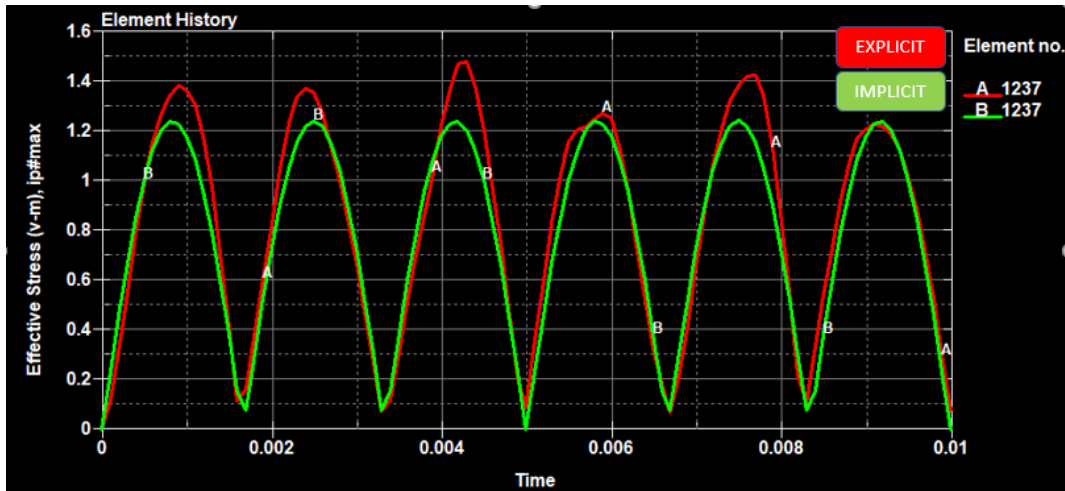


Figure 5-29: Stress of the most stressed interpolated element. Explicit (red) and implicit (green).

5.4.1 Eigenvalue analysis of shell isogeometric elements

To set and eigenvalue analysis with NURBS patch elements for simple models it is needed to operate in the classical way, including `CONTROL_IMPLICIT_EIGENVALUE` card to the implicit model and setting the number of eigenvalues to extract from the variable `NEIG`.

*CONTROL_IMPLICIT_EIGENVALUE (1)

	NEIG	CENTER	LFLAG	LFTEND	RFLAG	RHTEND	EIGMTH	SHFSCL
1	100	0.0	0	0.0	0	0.0	2	0.0
2	ISOLID	IBEAM	ISHELL	ITSHLL	MSTRES	EVDUMP	MSTRSCL	
	0	0	0	0	1	0	0.0100000	

Figure 5-30: `CONTROL_IMPLICIT_EIGENVALUE` card.

An igaplot file is generated together with the D3eigv file by the solver, so the interpolated mesh is available in the post-processing.

One hundred vibration modes have been extracted analyzing the foil already seen in this chapter, the first six of them are graphically shown in fig. 5-19 compared to an analogue FE model.

The results are satisfactory, the values of the frequencies diverge a few after the 50th mode probably because of the different deformations of the exact geometry with respect to the discretized one at high frequency (plot in figure 5-20).

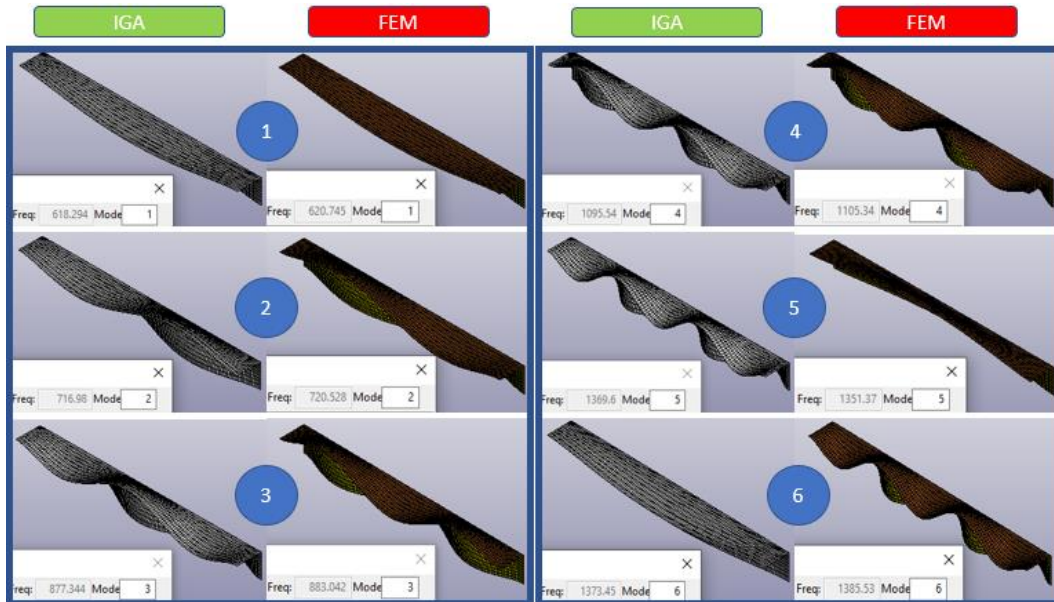


Figure 5-31: The first six modes in IG and FE model.

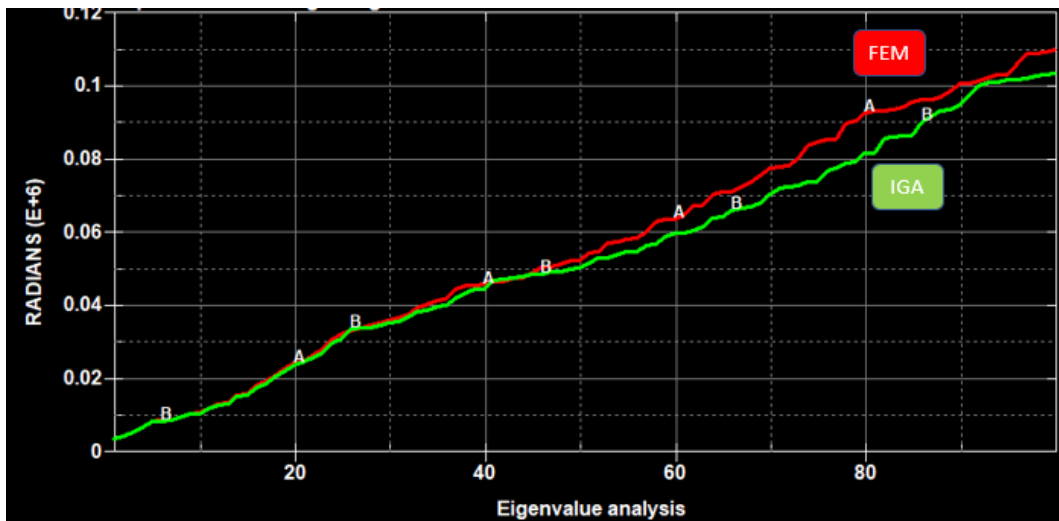


Figure 5-32: Plot of the frequency over one hundred modes.

5.4.2 Issues

The foil of the previous analysis has been modelled as a single NURBS patch, thanks to the knots multiplicity that allows the creation of sharp edges. Anyway also in case of eigenvalue analysis, the IGA performance with complex multi-patch geometries and trimmed NURBS elements is not assessable because of the amount of errors.

The rim shown in figure 5-21, composed by 5 shell NURBS patches and 15 trimmed NURBS patches, has been modelled to perform a frequency analysis that didn't work due to the problems produced by the trimmed NURBS elements.

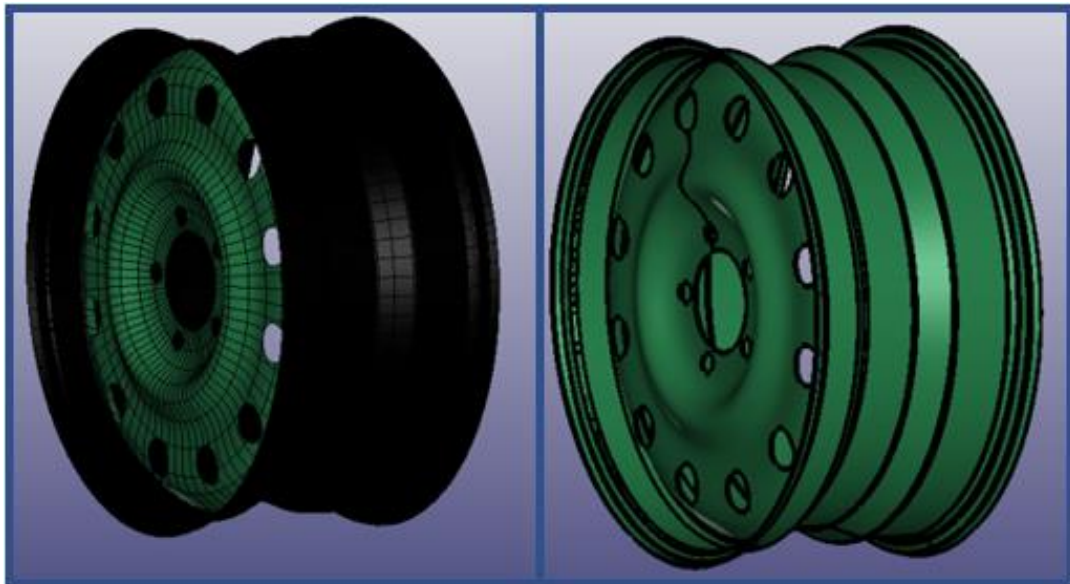


Figure 5-33: Wheel rim IGA model (left) and geometry (right).

```
*** Warning 30453 (INI+453)
*ELEMENT_SHELL_NURBS_PATCH_TRIMMED: Quadrature design #1 has failed.

*** Warning 30453 (INI+453)
*ELEMENT_SHELL_NURBS_PATCH_TRIMMED: Quadrature design #1 has failed.
```

Figure 5-34: Example of the warning messages appeared for all the trimmed NURBS patches.

5.4.2 Eigenvalue analysis with solid NURBS elements

In the same way, using the card `CONTROL_IMPLICIT_GENERAL` to activate the implicit solver and `CONTROL_IMPLICIT_EIGENVALUE` an eigenvalue analysis can be performed on solid NURBS patch elements.

In the following model a sinusoidal load is applied to one extremity of a cantilever beam.

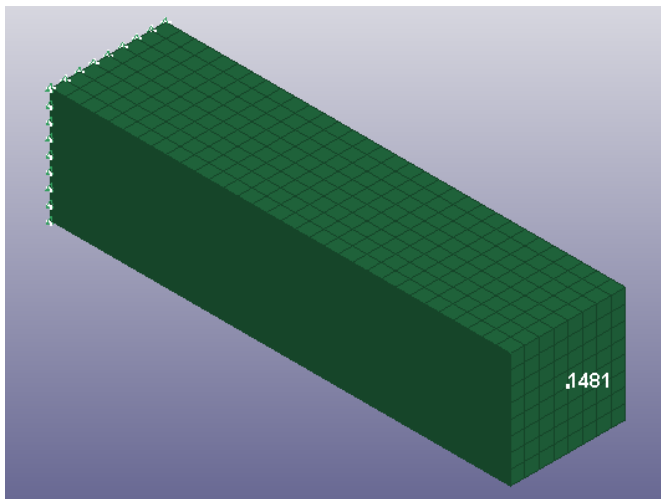


Figure 5-35: Solid NURBS model

In this case 20 eigenvalue are extracted and the comparison with the FE model produces good results, only in high frequency modes some considerable differences can be noticed.

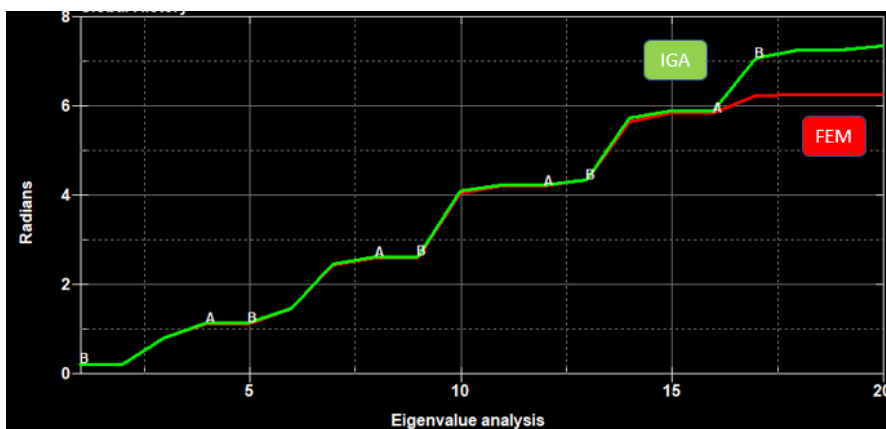


Figure 5-36: Plot of the frequency over twenty modes.

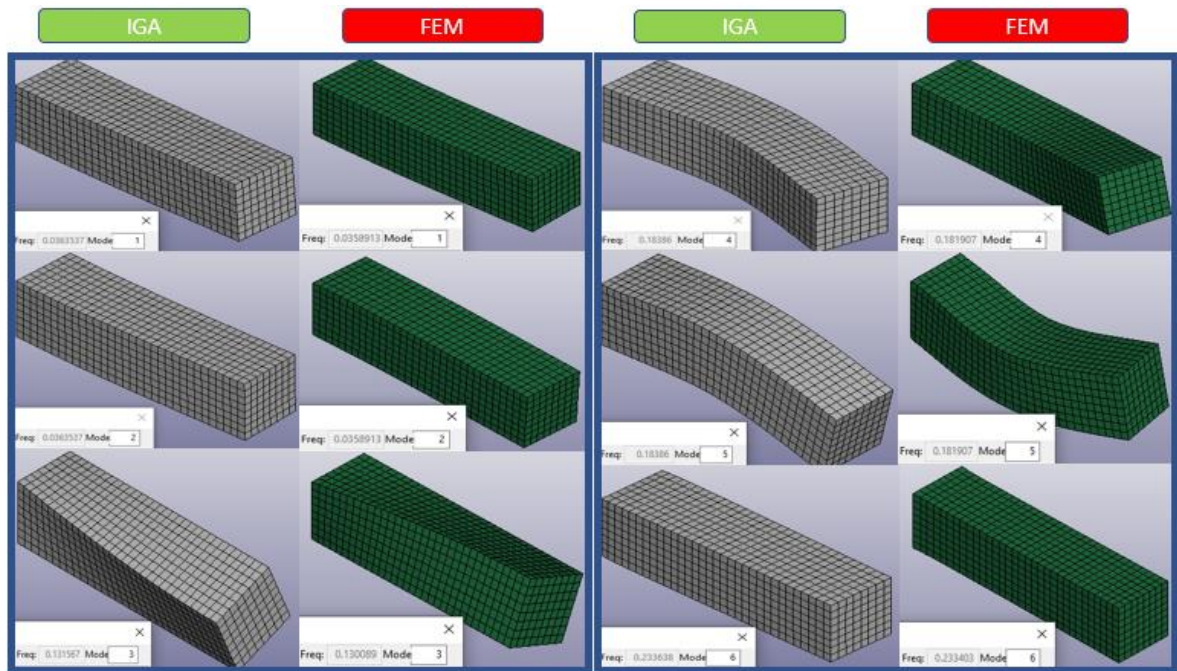


Figure 5-37: The first six modes in IG and FE model.

6. Composite modelling

IGA with composite materials can be performed in LS-DYNA, even though the possibility to model composites with NURBS elements is limited to rather simple geometries and constant through thickness properties.

Infact, the layup within a patch, i.e. number of layers, the thickness of individual layers, the orientations of the layers with respect to each other, need to be constant. Furthermore, the possibility to define the baseline orientation throughout a patch is currently limited to some global directions that are then mapped onto the shell surface.

6.1. Definition

6.1.1 Material definition

To map the material orientation, first of all an anisotropic/ortotropic layered material model must be defined. This option applies to material types 22, 23, 33, 34, 36, 40, 41-50, 54-56, 58, 59, 103, 116 and 194.

MAT-022 keyword is shown in figure below, to define the global directions, a suitable AOPT (material axes options) value must be chosen.

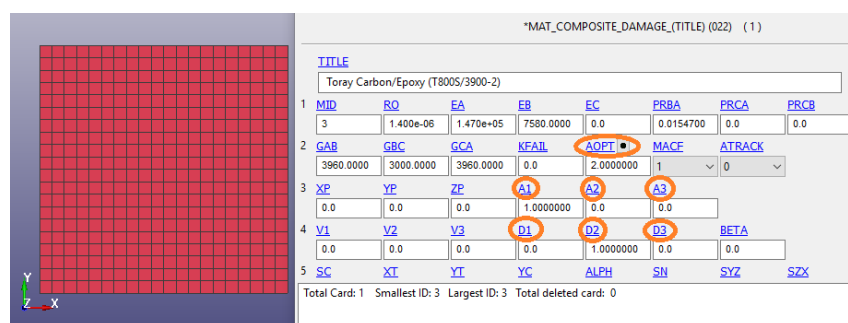


Figure 6-1: MAT-022 keyword, AOPT with A and D directions vector are highlighted.

AOPT values [28]:

EQ. 0.0: locally orthotropic with material axes determined by element nodes as shown in part (a) of Figure 6-2. The a-direction is from node 1 to node 2 of the element. The b-direction is orthogonal to the a-direction and is in the plane formed by nodes 1, 2, and 4. When this option is used in two-dimensional planar and axisymmetric analysis, it is critical that the nodes in the element definition be numbered counterclockwise for this option to work correctly.

EQ. 1.0: locally orthotropic with material axes determined by a point in space and the global location of the element center; this is the **a**-direction. This option is for solid elements only.

EQ. 2.0: globally orthotropic with material axes determined by vectors defined below, as with *DEFINE_COORDINATE_VECTOR.

EQ. 3.0: locally orthotropic material axes determined by rotating the material axes about the element normal by an angle, BETA, from a line in the plane of the element defined by the cross product of the vector **v** with the element normal. The plane of a solid element is the midsurface between the inner surface and outer surface defined by the first four nodes and the last four nodes of the connectivity of the element, respectively.

EQ. 4.0: locally orthotropic in cylindrical coordinate system with the material axes determined by a vector **v**, and an originating point, **P**, which define the centerline axis. This option is for solid elements only.

LT. 0.0: the absolute value of AOPT is a coordinate system ID number (CID on *DEFINE_COORDINATE_NODES, *DEFINE_COORDINATE_SYSTEM or *DEFINE_COORDINATE_VECTOR.

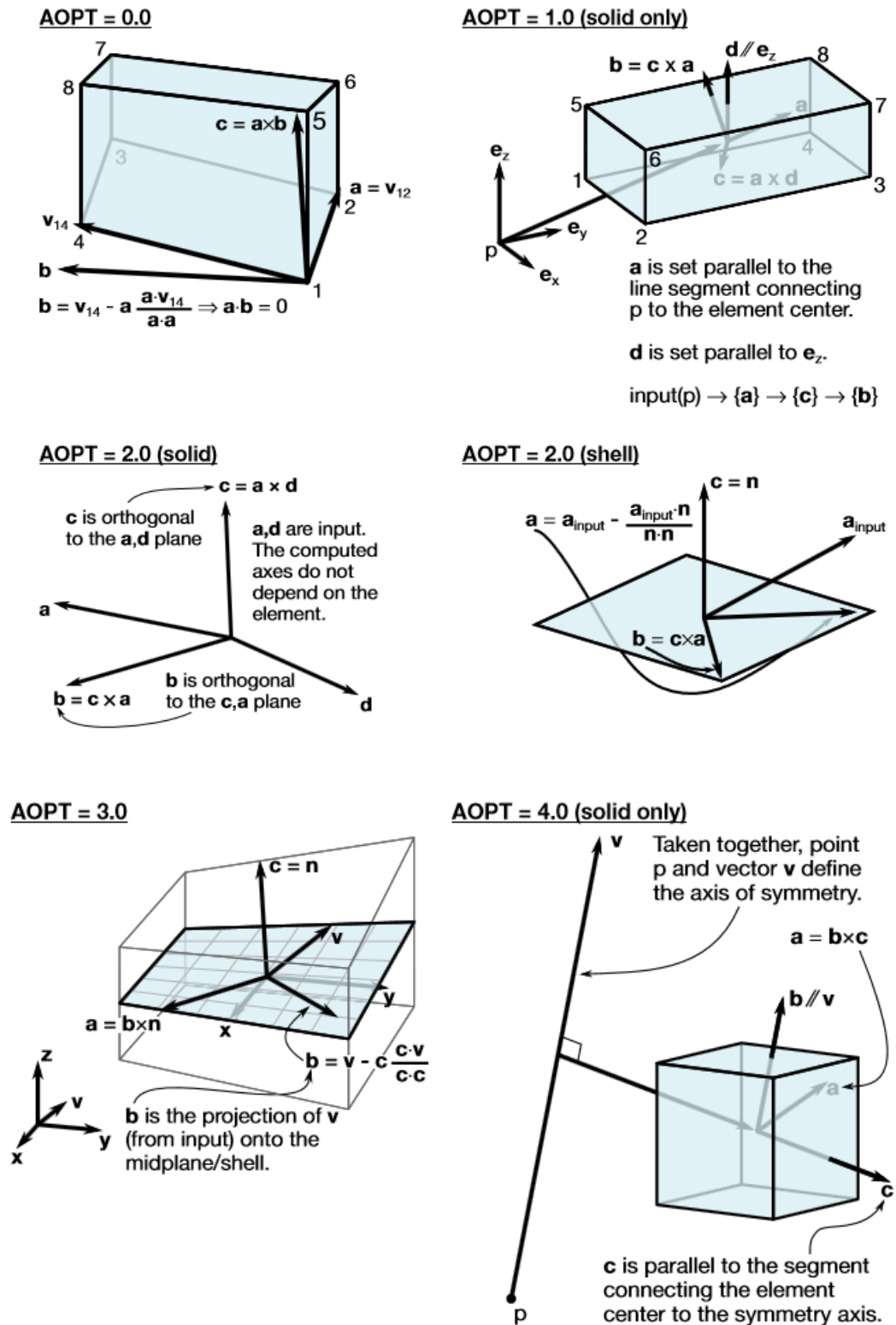


Figure 6-2: AOPT values explanation. [28]

6.1.2 Section and composite layers definition

To model composite layers on isogeometric element the only way is to switch on the option **ICOMP** in the **SECTION_SHELL** keyword.

ICOMP: Flag for orthotropic/anisotropic layered composite material model. This option applies to material types 22, 23, 33, 34, 36, 40, 41-50, 54-56, 58, 59, 103, 116 and 194.

EQ.0: Flag is tuned off (default),

EQ.1: A material angle in degrees is defined for each through thickness integration point. Thus, each layer has one integration point.

In Fig. 6-3 **SECTION_SHELL** keyword is shown. 8 carbon fiber layers are created, **Bi** variables represent the material angles of each layer (with respect to the chosen AOPT).

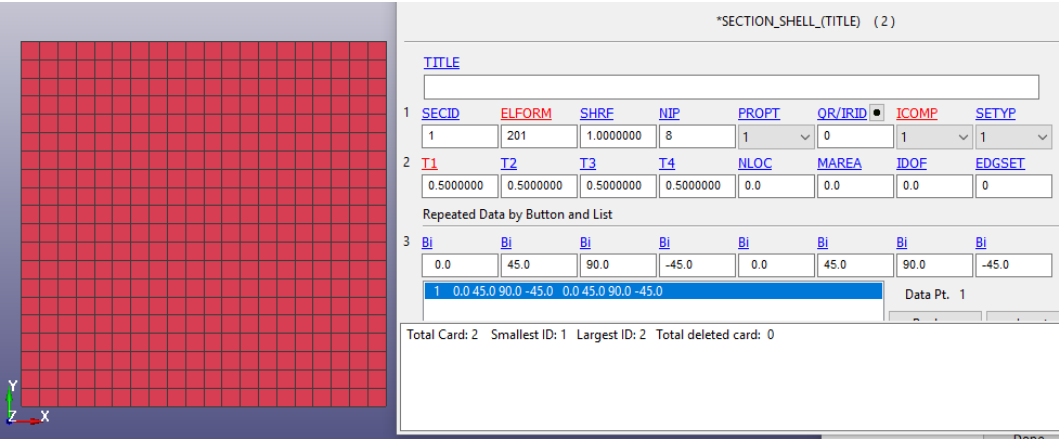


Figure 6-3: Definition of composite layers from SECTION_SHELL keyword.

An analysis with this laminate have been carried out, simulating the impact of the emispheric bullet already seen in chapter 5.

Results have been compared with a FE model created with the composite interface in the element edit tool. This simplified approach is not available yet for NURBS elements.

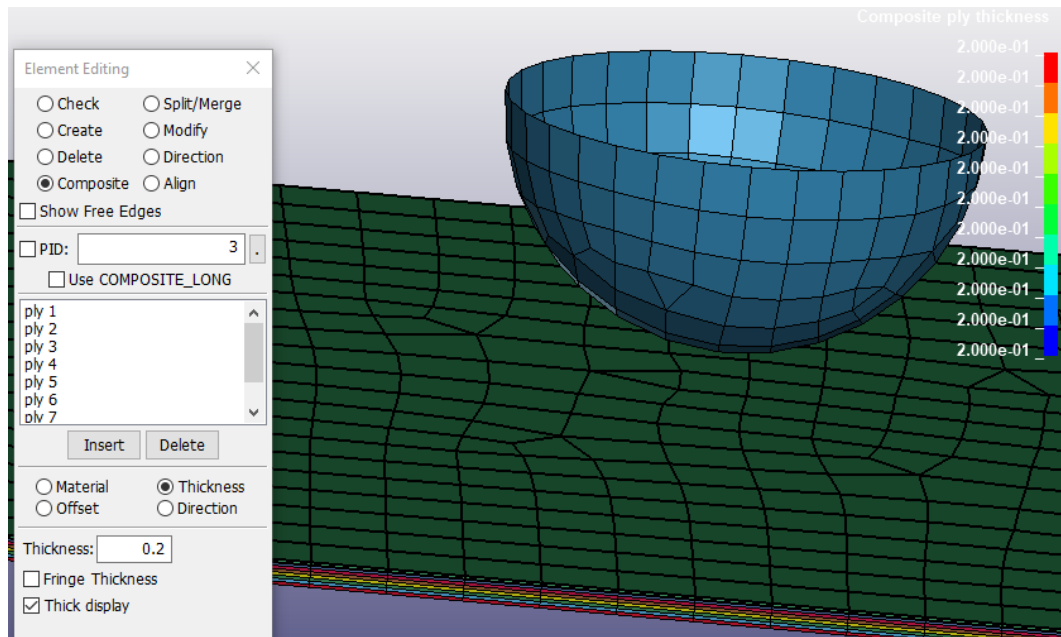


Figure 6-4: Composite interface with graphical rendering of the ply thickness.

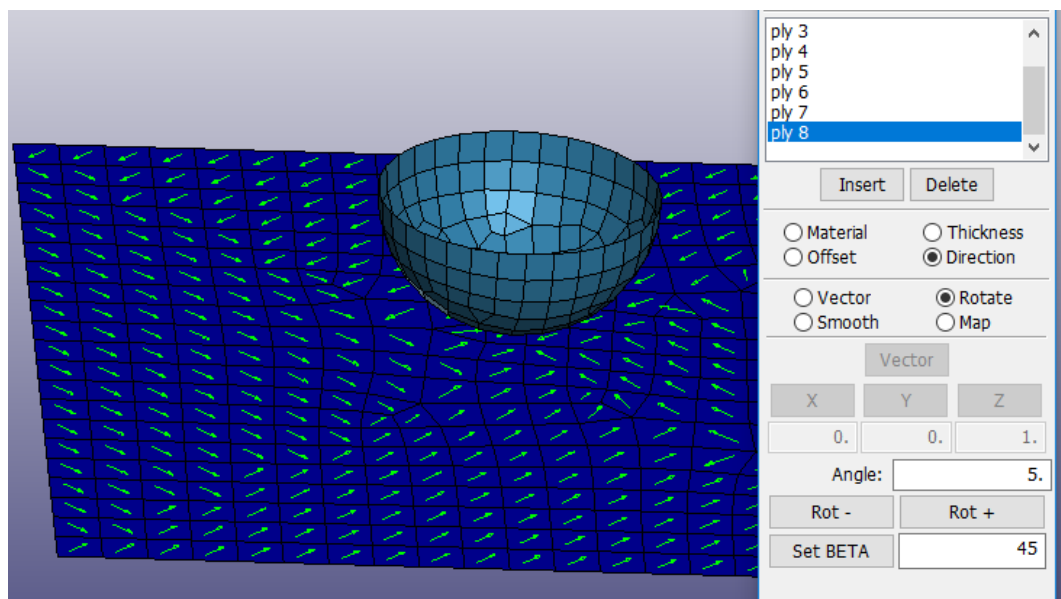


Figure 6-5: Ply orientation definition.

Using this method, normal SHELL elements are turned into SHELL_COMPOSITE elements, allowing the definition of a different layered structure for each selected element (Fig. 6-6). In IGA this freedom of modelling is

not achieved yet because, as seen before, a NURBS element is constituted by the whole patch, independently from the mesh size.

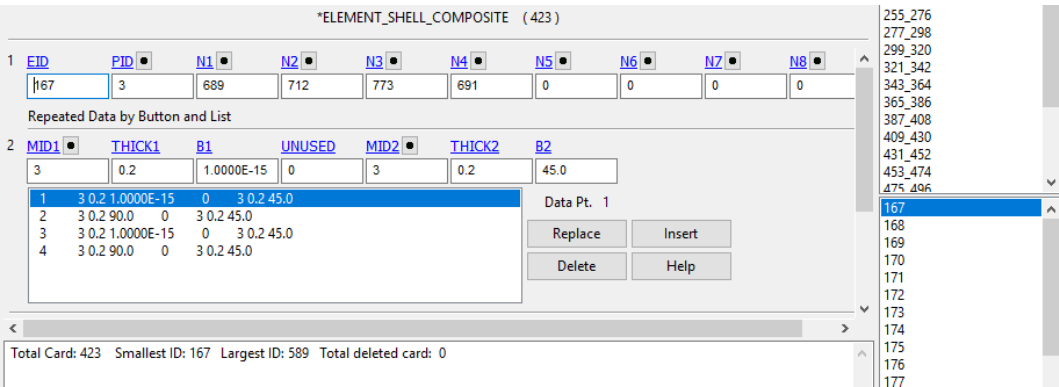


Figure 6-6: ELEMENT_SHELL_COMPOSITE keyword

6.1.3 Through thickness definition of different materials

If needed, different materials can be defined through the shell thickness. This is useful in case of sandwich panels or laminates composed by plies of different materials or properties.

A user defined integration rule must be created, using the keyword **INTEGRATION_SHELL** which, in turn, will be pointed by the SECTION_SHELL variable **IRID** (integration rule ID).

In defining the integration rule, the location and the weight for each integration point through the thickness can be specified and, moreover, the user can point to additional parts, with the purpose to specify the layer material.

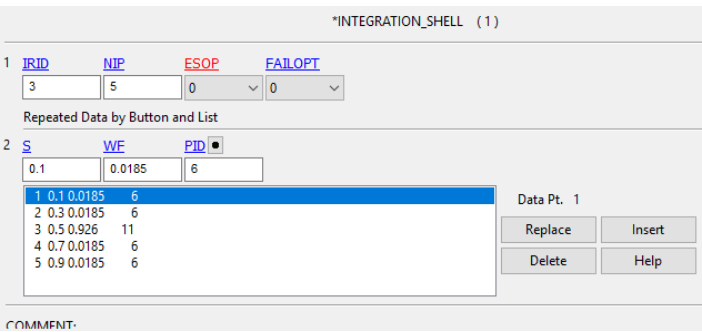


Figure 6-7: INTEGRATION_SHELL keyword.

IRID: Integration rule ID (IRID refers to IRID on *SECTION_SHELL card).

NIP: Number of integration points

ESOP: Equal spacing of integration points option:

-EQ.0: integration points are defined below,

-EQ.1: integration points are equally spaced through thickness such that the shell is subdivided into NIP layers of equal thickness.

FAILOPT: Treatment of failure when mixing different constitutive types, which do and do not include failure models, through the shell thickness. For example, consider the case where a linear viscoelastic material model, which does not have a failure option, is mixed with a composite model, which does have a failure option. Note: If the failure option includes failure based on the time step size of the element, element deletion will occur regardless of the value of FAILOPT.

-EQ.0: Element is deleted when the layers which include failure, fail.

-EQ.1: Element failure cannot occur since some layers do not have a failure option.

S: Coordinate of integration point in range -1 to 1.

WF: Weighting factor. This is typically the thickness associated with the integration point divided by actual shell thickness, that is, the weighting factor for the i th integration point $= \Delta t_i / t$ as seen in fig. 6-8.

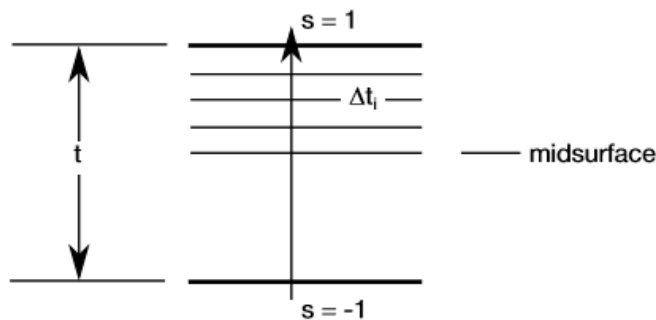


Figure 6-8: User defined shell integration rule. [28]

PID: Optional part ID if different from the PID specified on the element card. The average mass density for the shell element is based on a weighted average of the density of each layer that is used through the thickness. When modifying the constitutive constants through the thickness, it is often necessary to defined unique part IDs without elements that are referenced only by the user integration rule. These additional part IDs only provide a density and constitutive constants with local material axes (if used) and orientation angles taken from the PID referenced on the element card. In defining a PID for an integration point, it is okay to reference a solid element PID. The material type through the thickness can vary. [28]

6.2. Analysis

6.2.1 Impact on a carbon fiber plate

As stated before, the impact of a hemispheric bullet on a plate has been simulated in order to compare the performance of IGA composite modelling with FE analysis.

The model set up has already been presented and the plate is composed by 8 layers of T800 carbon fiber, laminated as shown in fig. 6-3. The tested NURBS patch have quadratic and cubic polynomial degree in both directions with a mesh size of 2.5, 5, 10 mm. (same size is chosen for FEA).

The results of IGA are not really satisfactory: the CPU runtimes are exponentially higher with respect to the FEA ones.

	RUNTIME (2CPUs)
FEA (10mm)	39 seconds
FEA (5mm)	4 minutes 38 seconds
FEA (2.5mm)	32 minutes 15 seconds

IGA (quadratic, 10mm)	2 minutes 53 seconds
IGA (quadratic, 5mm)	18 minutes 53 seconds
IGA (quadratic, 2.5mm)	2 hours 29 minutes
IGA (cubic, 10mm)	9 minutes 20 seconds
IGA (cubic, 5mm)	1 hour 7 minutes
IGA (cubic, 2.5mm)	8 hours 12 minutes

The results of the nodal displacement (measured on the plate center) and kinetic energy are similar for comparable mesh sizes, converging to the same values after refinement. An higher polynomial degree gives an advantage only in case of coarse mesh (10 mm), where the displacement peak at the beginning is smaller with respect to the other two models.

The quadratic IGA displacement peak has been cut for graphic reasons but it resulted 3 times higher than the FEA one.

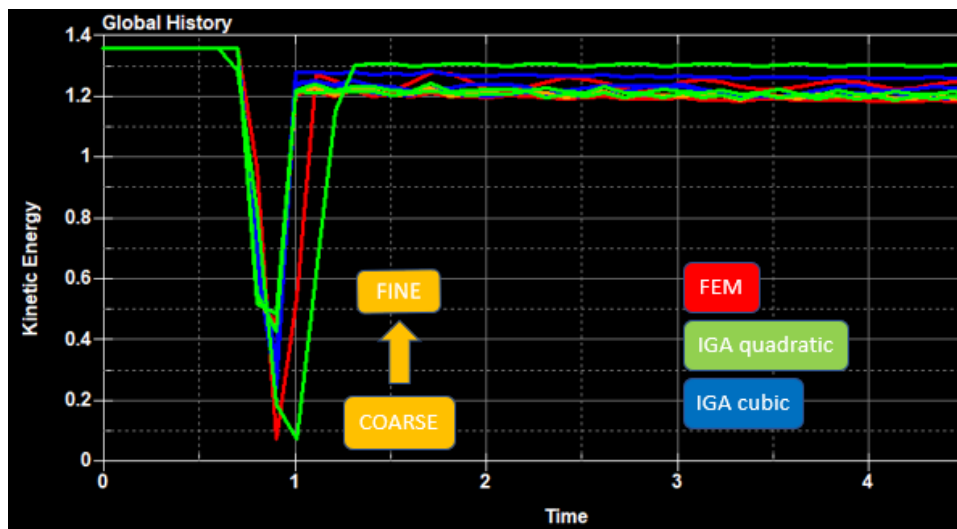


Figure 6-9: Kinetic energy plotted over time. IGA quadratic (green), IGA cubic (blue) and FEA (red).

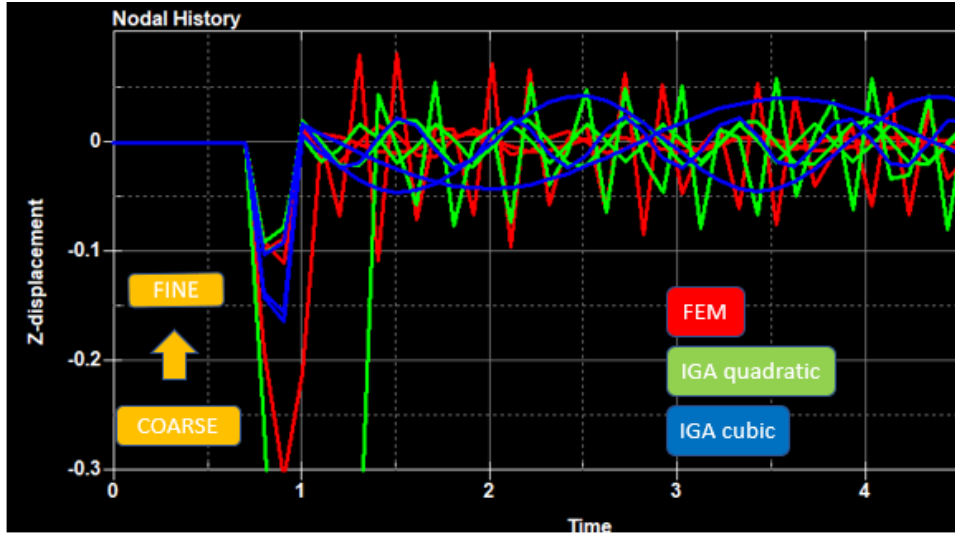


Figure 6-10: Z-displacement of the central node of the plate plotted over time. IGA quadratic (green), IGA cubic (blue) and FEA (red).

6.2.2 Three point bending of a sandwich panel

A sandwich panel composed by 4 plies of carbon fiber and a foam core has been modelled by means of the INTEGRATION_SHELL card. A 3 points bending simulation has been then performed, in order to compare the result with a FE analogue model, created using the composite tool.

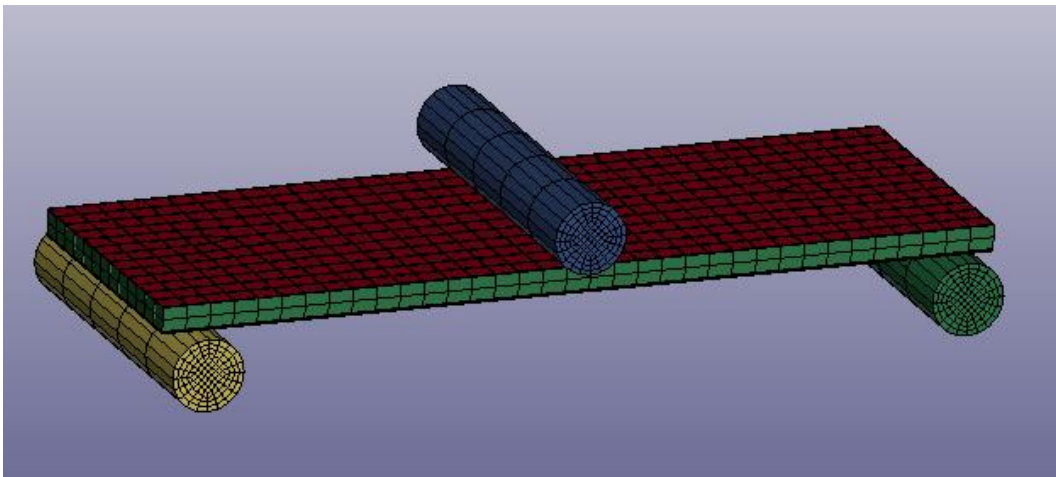


Figure 6-11: FE model with visible thickness.

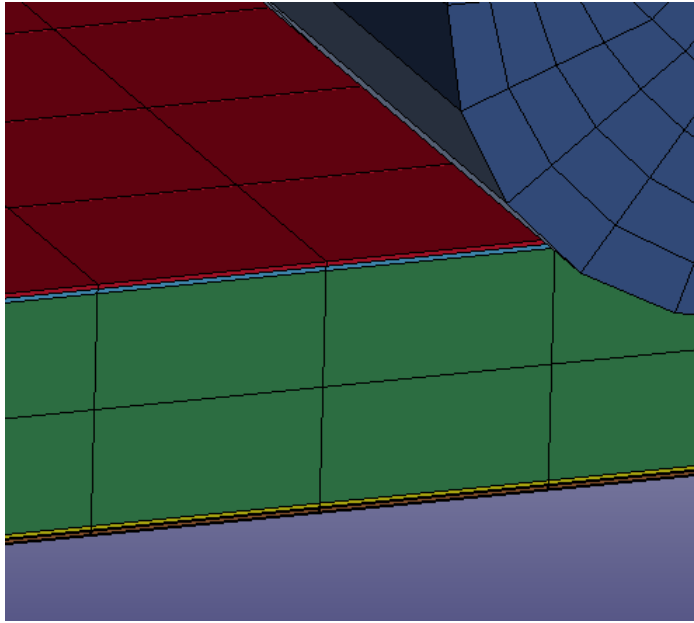


Figure 6-12: Particular of the plies.

A MAT 22 (COMPOSITE_DAMAGE) is used to model both carbon fiber and core, to avoid errors related to the material angle definition.

In the table below, the cpu runtime to simulate the first 75 seconds of the test is shown. In a second analysis the FEA termination is increased to 200 seconds, with a final cpu time of 55 minutes.

	RUNTIME (4CPUs)
IGA (quadratic, 10mm)	4 hours 41 minutes
FEA (10mm)	21 minutes

In this case, the cpu results are even worse than in the case before, with a runtime, for the isogeometric model, more than thirteen times higher with respect to FEA.

Moreover, processing the results, is noticed that in the isogeometric analysis, a sudden failure takes place at time $t=62$ ms, while in FEA this doesn't happen for the whole duration of the test.

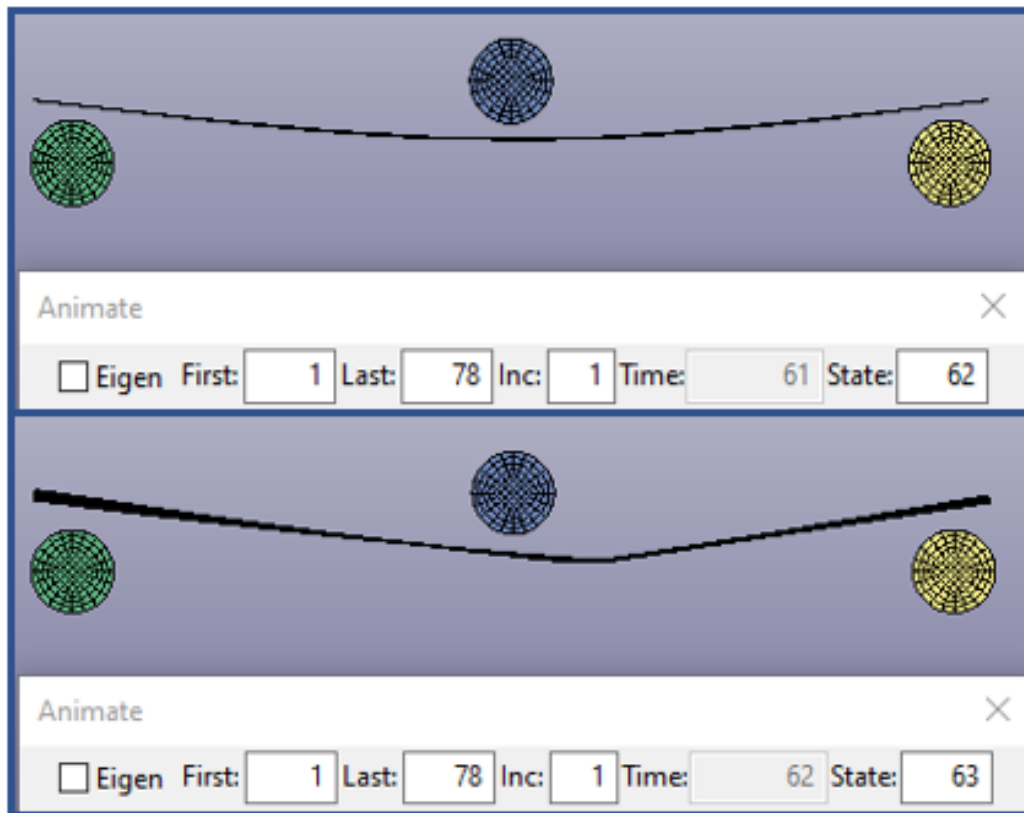


Figure 6-13: IGA - The simulation at time $t=61s$ (above) and $t=62s$ (below).

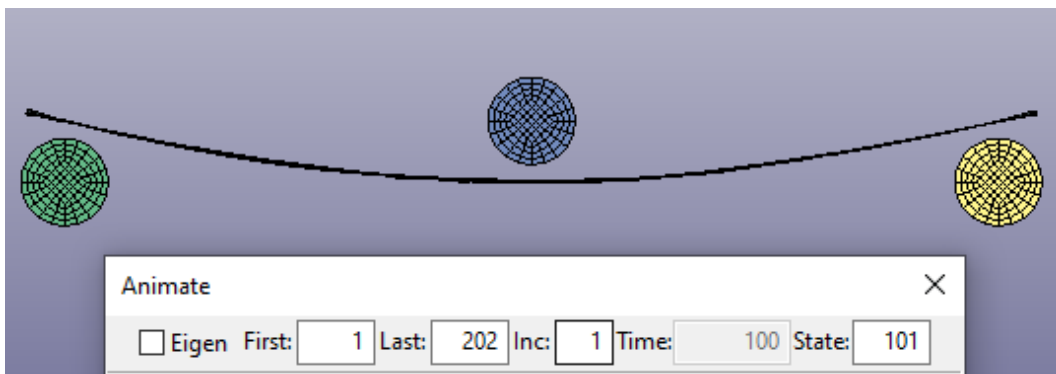


Figure 6-14: FEM – The simulation at time $t=100s$

The behaviour of the FE model is much more coherent with a real sandwich panel, as can be seen in the graph below. (Fig. 6-16 z-force plotted over time).

To have a better overview of the results, the FE analysis duration is increased to 200s.

The FE panel reaches some local peaks, while the NURBS model only one, after a perfectly elastic behaviour (the test rig has a constant velocity).

The peaks represent the failures of the carbon fiber, moreover the perfectly linear behaviour of the IG model makes arouse suspicions on the actual performance of the INTEGRATION_SHELL method.

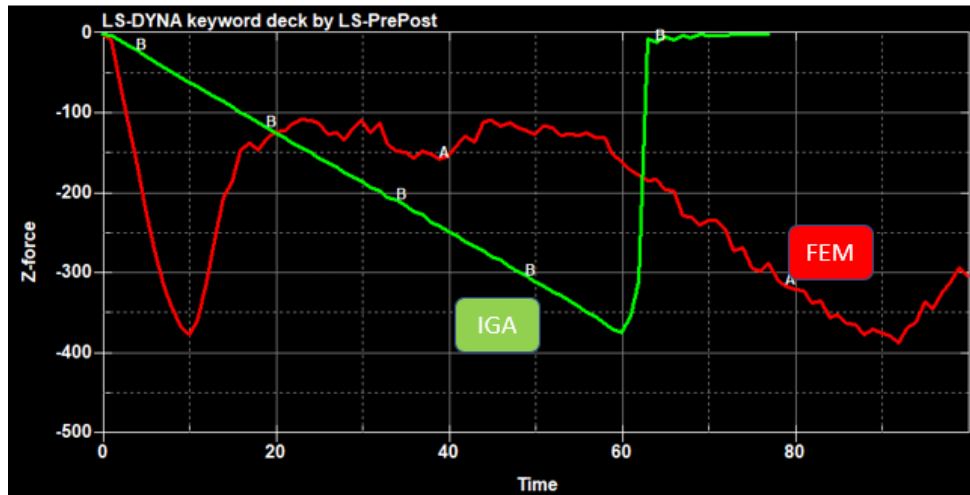


Figure 6-15: Z-force plotted over time, IGA (green) and FEA (RED).

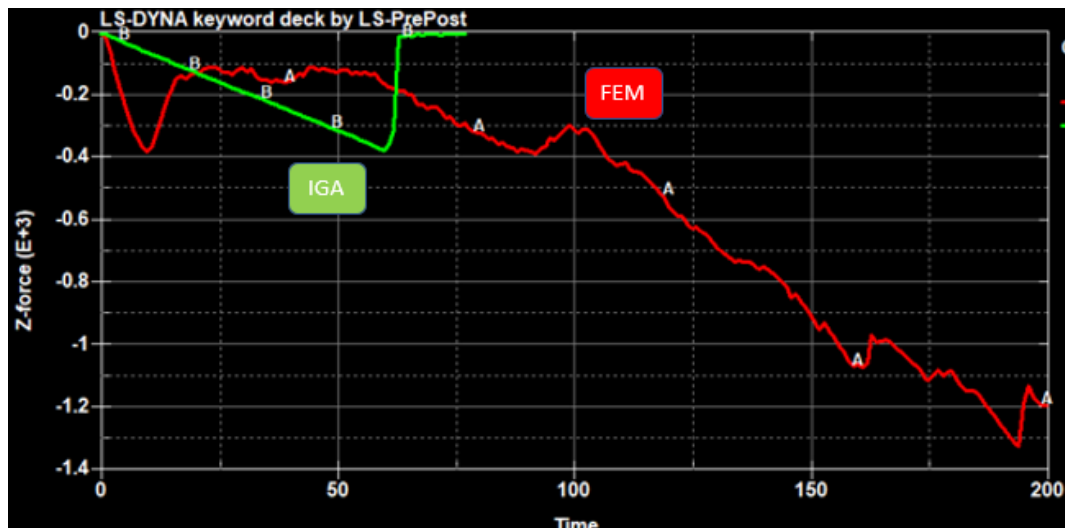


Figure 6-16: FE analysis termination time changed to 200s.

7. Sheet metal forming

7.1. State of the art

Many researches confirm that IGA has shown good stability and accuracy in sheet metal forming simulations.

This kind of simulations are part of a multi-stage process, therefore the analysis tool should be capable of mapping the results achieved from one step to another, no matter what the individual stage may be (deep-drawing, trimming, hemming, springback and others). The most important results to map are: current stress and strain states, the equivalent plastic strain and the thickness variations due to the forming process.

In LSDYNA this is typically done with a so called DYNAIN file, using the keyword INTERFACE_SPRINGBACK. The file is written during the simulation of one stage and is then read back in the next one, in order to reinitialize the necessary values. Two keywords have been added to LS-DYNA to perform this task in IGA: INITIAL_STRESS/STRAIN_NURBS_PATCH (keyword not available in LS-PrePost). [16]

***INITIAL_STRESS_SHELL_NURBS_PATCH**

	1	2	3	4	5	6	7	8
Card 1	EID	NPLANE	NTHICK	NHISV	LARGE			
Card 2	R	S	T					
Card 3	SIGXX	SIGYY	SIGZZ	SIGXY	SIGYZ	SIGZX	EPS	
Card 4	HISV1	HISV2	HISV3	HISV4	HISV5	HISV6	HISV7	HISV8
Card					

Figure 7-1: INITIAL_STRESS_NURBS_PATCH keyword. [60]

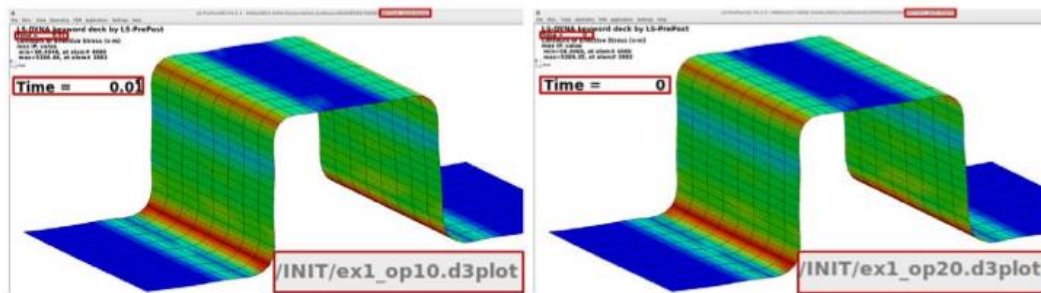


Figure 7-2: Von mises stress distribution at the end of stage one (left) and at the beginning of stage 2 (right). [16]

Another important feature that has been developed recently is the possibility to perform one step simulations with IGA, improving the performance of the crash analysis by including forming history into the model. Infact, many parts are manufactured from flat sheet metals in car design, and their mechanical properties variations, due to the forming process, can create big inaccuracies in the crash analyses.

In the one step method the part is assumed to deform from an initial flat blank, to its final shape, in only one step. It requires the mechanical properties of the part and its final geometry, subsequently an initial blank is optimized for minimum energy.

The one step analysis in IGA can be activated using the keyword `CONTROL_FORMING_ONESTEP`, setting the variable `OPTION` to 7. The output result are written in a *igaonestepresult* file that can be used exactly in the same way as a DYNAIN file. [31]

*CONTROL_FORMING_ONESTEP (0)								
1	<u>OPTION</u>	<u>TSLMAX</u>	<u>AUTOBD</u>	<u>TSLMIN</u>	<u>EPSMAX</u>	<u>unused</u>	<u>LCSDG</u>	<u>DMGEXP</u>
	7	1.0	0.3	1.0	1.0			
2	<u>FLATNAME</u>							

Figure 7-3: `CONTROL_FORMING_ONESTEP` keyword.

The IGA one step approach is demonstrated and compared to FEA in [31], with a couple of examples. The first example uses a single untrimmed NURBS patch while the second uses a TRIMMED one.

A comparison of the Von Mises stress is shown in fig. 7-4. The distribution is similar in the two models, while the maximum stress slightly differs, which is best explained by localization.

A comparison of the effective plastic strain fields is shown in fig. 7-5.

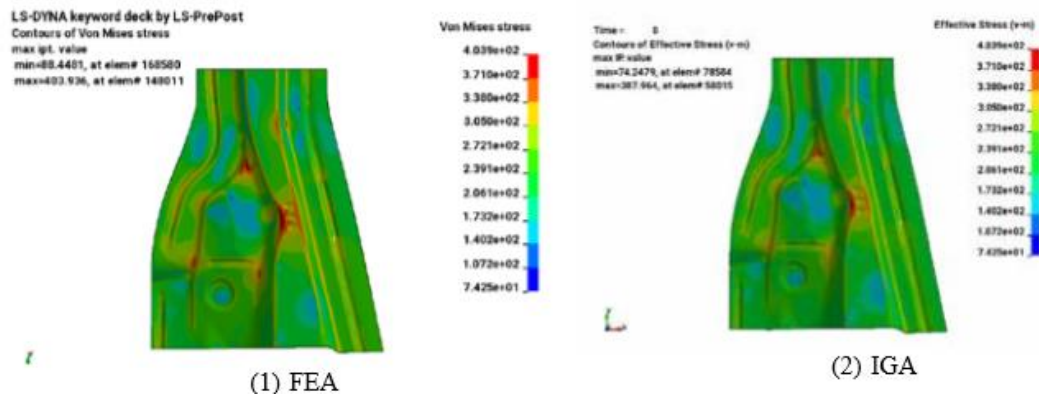


Figure 7-4: Comparison of the Von Mises stress obtained via the one-step approach, FEA (left) and IGA (right). [31]

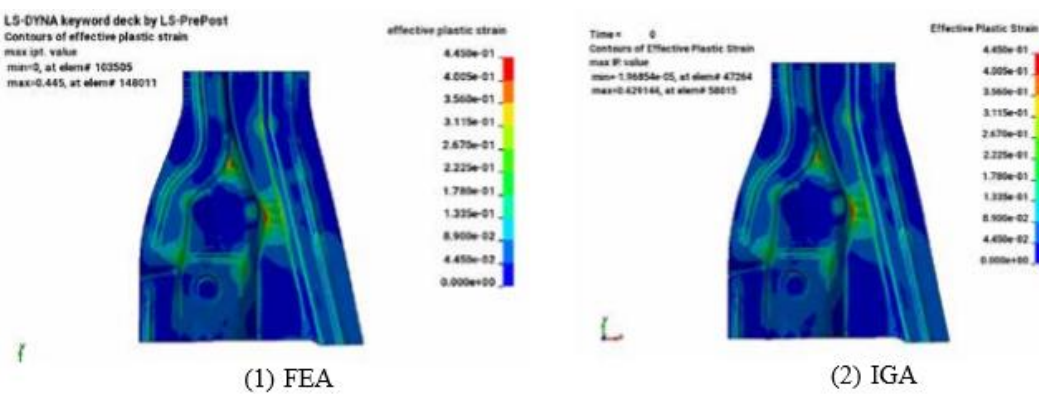


Figure 7-5: Comparison of the effective plastic strain fields obtained via the one-step approach, FEA (left) and IGA (right). [90]

Effective plastic strain fields of the second example (trimmed NURBS patch) are shown in Fig. 7-6. due to localization, the maximum strains are slightly different in the FE model as shown in Fig. 7-7.

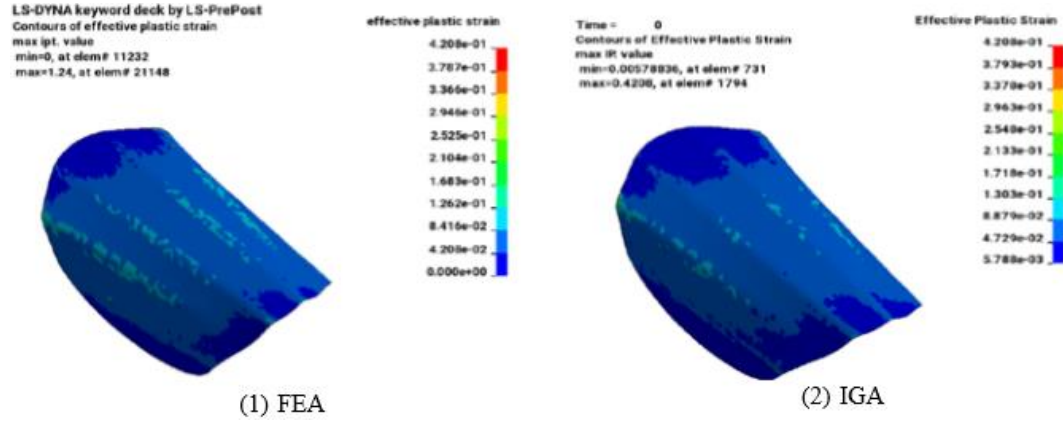


Figure 7-6: Comparison of the effective plastic strain fields obtained via the one-step approach, FEA (left) and IGA (right). Second example (trimmed NURBS patch). [31]

7.2. Multi-stage simulations.

A multi-stage simulation is presented in [16], proceeding with the following steps:

- 1 – Gravity
- 2 – Deep drawing
- 3 – Trimming
- 4 – Springback analysis

In each step the stress/ strain situation is mapped as already explained using INITIAL_STRESS/STRAIN_SHELL_NURBS_PATCH.

In stage 1, the blank is placed into the forming tool. An implicit static analysis has been carried out.

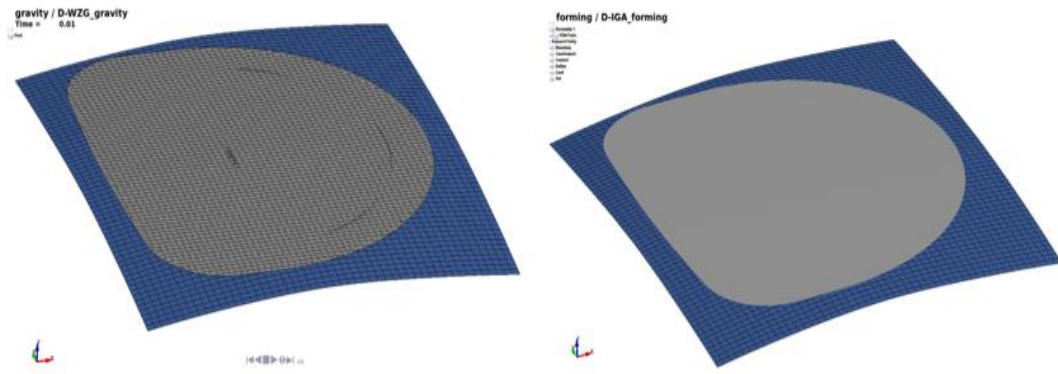


Figure 7-7: Stage 1 (Gravity). FEA (left) and IGA (right). Results are then mapped and loaded into the next step. [60]

In the stage 2, an explicit analysis has been carried out to simulate the deep drawing process.

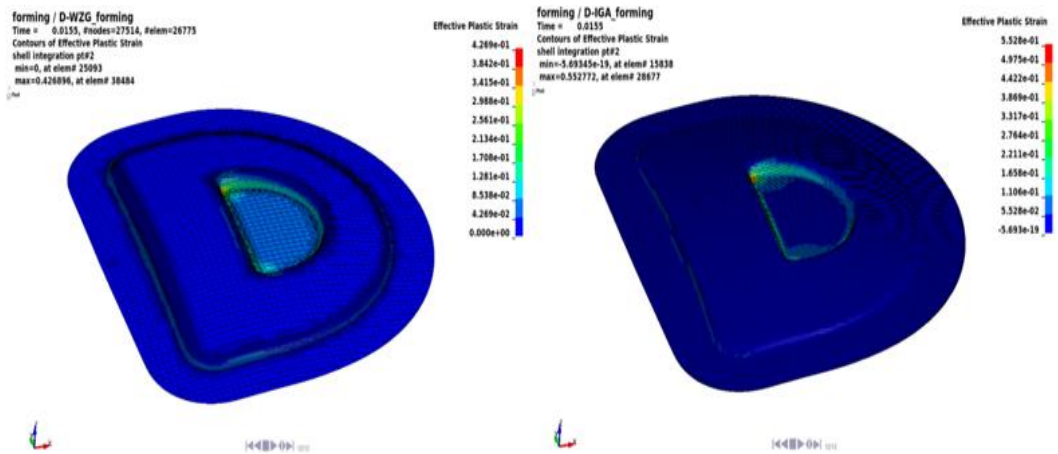


Figure 7-8: Effective plastic strain fields after the deep drawing process. FEA (left) and IGA (right). [16]

In stage 3, no analysis has been carried out, the part has been trimmed and the results mapped for the next stage.

To trim the part, the keyword `CONTROL_FORMING_TRIMMING` must be used together with the keyword `DEFINE_CURVE_TRIM_3D`. The latter is needed to define the trim loop, while the former receives a part set ID in input. Elements in

the part set will be automatically trimmed in the defined direction if they intersect the trim curves.

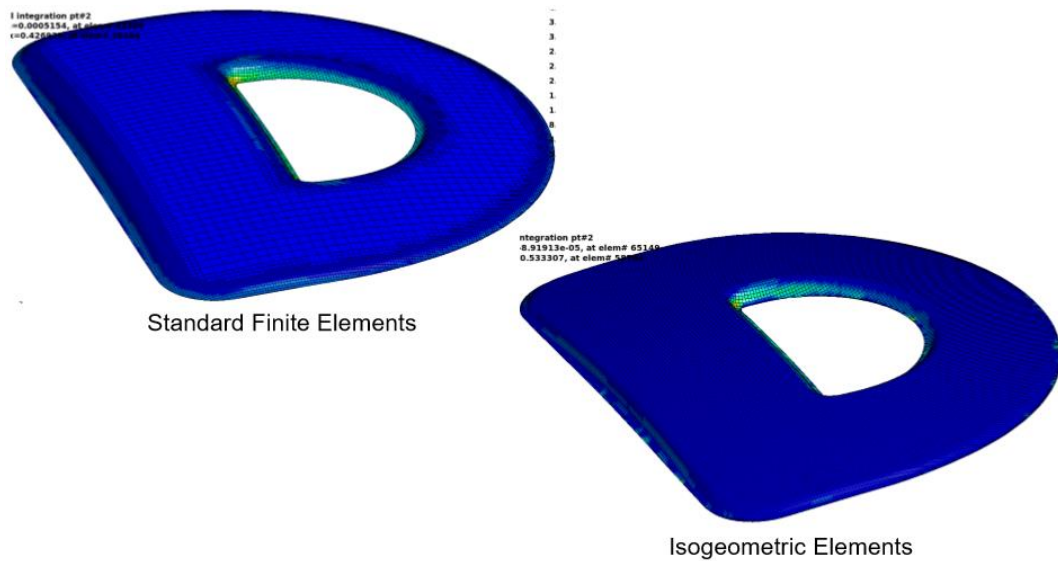


Figure 7-9: Effective plastic strain fields comparison after trimming, FEA (left) and IGA (right). [16]

In the stage 4, Springback analysis, an implicit static analysis must be carried out to find the equilibrium for final geometry. However, it is not available yet for IGA in LSDYNA.

8. CONCLUSIONS

In IGA, the exact geometry can be represented by utilizing a coarse mesh structure. This prevents unnecessary mesh refinements for geometrical representation. Thus, refinement techniques are only used for the solution accuracy. Moreover, mesh refinement is simply handled by reindexing the parametric space without interacting with the geometry contrarily to the refinement strategies of FE method that require interaction with the CAD system at each stage.

Good results have been obtained from simple shell models composed by one or a few NURBS patches, while many instabilities have been found in more complex multi-patch structures. The first source of problems seems to be the presence of trimmed NURBS patches that are subjected to wrong energy fluctuations even in basic analyses.

The runtimes are, in general, much higher with respect to FEA. This is probably due to the generation of the igaplot files and to the computational complexity induced by the p-refinement. Moreover the creation of additional files for IGA, makes the output folder much heavier than the equivalent FEA one (an interpolated mesh is needed in contacts and in the post-processing).

The analysis of solid elements is possible only in case of small deformations and simple shapes: stamping, crashes and penetrations are impossible to be set up. Elements erosion is not practicable, considering the IGA element definition.

Composite modelling showed a good behaviour in the first simple example, while the 3 point bending simulation has underlined the limits of the through thickness integration method, that after many trials hasn't actually worked. Moreover, other weak points in in composites IGA are the limits given by the element definition (a whole patch must have a single lay up) and the gigantic CPU times.

IGA is well implemented in LS-DYNA for what concerns sheet metal forming analysis, with the availability of the full multi stage process (except for springback analysis) and the one step approach.

LIST OF FIGURES

Figure 1-1: Estimation of the relative time costs of each component of the model generation and analysis process at Sandia National Laboratories. [13]	3
Figure 1-2: Manufacturing time in relation with the number of parts of a model [General Dynamics & Electric Boat Corporation]	4
Figure 1-3: : Comparison of meshing for standard FE and IGA.[16].....	5
Figure 1-4: Thin shell structures exhibit significant imperfection sensitivity: (a) faceted geometry of typical finite element meshes introduces geometric imperfections and (b) buckling of cylindrical shell with random geometric imperfections [14].....	7
Figure 2-1: Example of cubic Bézier curve with its control polygon [wiki]......	10
Figure 2-2: the evaluation of a point on that curve at $\xi = 0.5$ with the de Casteljau algorithm.....	11
Figure 2-3: Bernstein polynomials of different degree plotted over the domain $[0,1]$, i.e. the basis functions of a Bézier curve. (a) For the depicted curve in fig. 2-1 (a) with degree 3; (b) the functions of a curve with seven control points ($p = 6$) [24].	11
Figure 2-4: Dependencies between results of basis functions for computing a cubic basis function [17].....	14
Figure 2-5: Basis functions for order 0 and 1 for uniform knot vector = $\{0,1,2,3,4,5\}$ [21].....	15
Figure 2-6: Comparison of quadratic finite element shape functions and B-spline basis functions [23]......	16
Figure 2-7: Quadratic basis functions drawn for non-uniform open knot vector $\Xi = [0\ 0\ 0\ 0.2\ 0.4\ 0.4\ 0.6\ 0.8\ 1\ 1\ 1]$. [23]	17
Figure 2-8: A Quadratic B-Spline curve example [23].....	20
Figure 2-9: Basis functions of knot vectors (a) $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ and b) $\mathcal{H} = \{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$ [23].....	21

Figure 2-10: An Example B-Spline surface [22].	21
Figure 2-11: control net $\mathbf{Bi,j}$ [22].	22
Figure 2-12: Knot insertion refinement technique [18].	25
Figure 2-13: Order elevation refinement technique [18].	27
Figure 2-14: k-refinement sequence comparison (a) Base case of one linear element. (b) Classic p-refinement approach: knot insertion followed by order elevation results in seven piecewise quadratic basis functions that are C_0 at internal knots (c) New k-refinement approach: order elevation followed by knot insertion results in five piecewise quadratic basis functions that are C_1 at internal knots. [14].	29
Figure 2-15: An Example of projective transformation of (a) Control points (b) Curves [14].	31
Figure 2-16: An Example for NURBS surface (a) Control net for toroidal surface (b) Toroidal surface [14].	33
Figure 2-17: Rhinoceros 3D environment.	34
Figure 2-18: Car body design on Siemens NX.	35
Figure 2-19: Airbus A380 modelling in CATIA V5.	36
Figure 3-1: Screenshot from LS-PrePost showing the results of an LS-DYNA simulation of a car impacting a rigid wall at 120 kph. [Wikipedia]	37
Figure 3-2: The keyword *ELEMENT_NURBS_PATCH_2D in LS-DYNA [15]	41
Figure 3-3: Interpolation nodes and elements [15]	42
Figure 3-4: Data cards: title card, card 1, card 2 [28]	45
Figure 3-5: knot vector cards for r-direction. Variables \mathbf{RKm} are the values of the univariate knot vector in r -direction [28]	48
Figure 3-6: knot vector cards for r-direction. Variables \mathbf{SKm} are the values of the univariate knot vector in r -direction [28]	48
Figure 3-7: Control Grid cards, variables \mathbf{Nk} are the control point IDs, defined via *NODE, to define the control grid - LT.0: Control point with rotational DOFs for FORM = 4 /-4; [28].	49
Figure 3-8: Control weight cards, variables \mathbf{Wk} are the control weights of the surface patch. [28].	49

Figure 3-9: Trimming loop title card. [28]	49
Figure 3-10: Variables C1 are Trimming curve ID pointing to a curve defined using *DEFINE_NURBS_CURVE. A unique number has to be chosen. [28]...	50
Figure 3-11: *DEFINE_NURBS_CURVE card. [28].....	51
Figure 3-12: Knot vector card. The knot vector of length N+P+1 is given below requiring $\text{ceil}[(N+P+1)/8]$ cards in total. The knot vector has to be normalized to the [0,1] interval. [28]	51
Figure 3-13: Control point cards. The spatial coordinates of the control points and the control weights are listed on N cards. Control weight entries are disregarded unless WFL = 1 on Card 1. [28]	52
Figure 3-14: Defining a quadratic NURBS curve using the *DEFINE_NURBS_CURVE keyword. [28].....	53
Figure 3-15: A NURBS volume example. [29]	54
Figure 3-16: ELEMENT_SOLID_NURBS_PATCH, Cards 1 and 2. [28].....	55
Figure 3-17: Example of definition of ELEMENT_SOLID_NURBS_PATCH. [28].....	56
Figure 3-18: Bi-quadratic NURBS patch and interpolation elements dependent on the parameter NISR and NISS. [30]	57
Figure 3-19: NURBS_TIED keyword interface in LS-PrePost.....	59
Figure 3-20: Mapping procedure to define the B-Rep edge elements on the master curve. A B-Rep edge element is defined between any two consecutive points (either black or red) shown in the right part of the figure. [27]	61
Figure 4-1: Keyword input editor. Variables RK, SK and N are defined for the NURBS patch on the left.	63
Figure 4-2: Single patch CAD surface imported in LS-PrePost.....	63
Figure 4-3: NURBS single patch element creation in LS-PrePost	64
Figure 4-4: Multi patch Isogeometric part. Four SHELL_NURBS_PATCH elements are created, with four NURBS_TIED_EDGE_TO_EDGE contacts. In order to form a single part, the NURBS patch elements were created using the same part ID.....	65
Figure 4-5: H-refinement (left) and P-refinement (right) interfaces.....	66
Figure 4-6: K-refinement interface of LS-Prepost.....	67

Figure 4-7: The same NURBS patch element, after creation (left) and after refinement (right).....	67
Figure 4-8: Example of trimmed NURBS patch. The geometry imported from the CAD is already trimmed, the element created by the editor is a SHELL_NURBS_PATCH_TRIMMED and 16 NURBS_CURVE have been automatically defined (used as trimming loops).....	68
Figure 4-9: simple parallelepiped NURBS element created from the keyword input form, on the right the 3 knot vectors variables.	69
Figure 4-10: CAD imported basic surface (left), revolution solid created in LS-PrePost (right).....	70
Figure 4-11: Solid NURBS element creation with the 3D NURBS editor.....	70
Figure 4-12: Sweeping surface and curve (left), solid NURBS element created with the editor (right).....	71
Figure 4-13: Solid bullet model, before and after refinement	71
Figure 4-14: The three versions of the plate.	72
Figure 4-15: Simple IGA model, impact of and emispheric bullet on a plate.....	73
Figure 4-16: From left to right: Both geometries are turned on, only NURBS patch turned on(red), interpolated mesh turned on (green).	74
Figure 4-17: Displacement contour plot in case the NURBS patch is not turned off.....	74
Figure 4-18: The three models are represented at time $t=0$	75
Figure 4-19: Maximum displacements of the plates during the impacts.	75
Figure 4-20: Maximum stresses suffered by the plates.	76
Figure 4-21: Kinetic energy plot: flat plate.	76
Figure 4-22: Kinetic energy plot: bulged plate.....	77
Figure 4-23: Kinetic energy plot: bulged/trimmed plate.	77
Figure 4-24: Kinetic energy of the bullet.....	78
Figure 4-25: Internal energy of the plate	78
Figure 4-26: Kinetic energy of the plate.....	78
Figure 5-1: Impactor (blue) and crashbox (red).....	80
Figure 5-2: Model boundary conditions	80
Figure 5-3: NURBS patch deformation in the post processor.	82

Figure 5-4: IGA- quadratic patch - Resultant displacement at time $t=30\text{ms}$. Mesh size decreases rom left to right: 20, 10, 5 mm.	82
Figure 5-5: IGA- cubic patch - Resultant displacement at time $t=30\text{ms}$. Mesh size decreases rom left to right: 20, 10, 5 mm.	82
Figure 5-6: FEM - Resultant displacement at time $t=30\text{ms}$. Mesh size decreases rom left to right: 20, 10, 5 mm.....	83
Figure 5-7: Kinetic energy plot for all the models. IGA quadratic (green), IGA cubic (blue) and FEM (red), mesh size from coarse to fine in the direction of the arrow.	83
Figure 5-8: Spotwelded s-rail isogeometric model.	84
Figure 5-9: Particular of the spotweld beam elements	84
Figure 5-10: Three different levels of refinement (20, 10, 5 mm) in IG model (left) and FE model (right).....	85
Figure 5-11: Spotwelds failure at time $t=10\text{ ms}$, mesh size 20mm, IGA (left) and FEM (right).	86
Figure 5-12: Spotwelds failure at time $t=10\text{ ms}$, mesh size 10mm, IGA (left) and FEM (right).	87
Figure 5-13: Spotwelds failure at time $t=10\text{ ms}$, mesh size 5mm, IGA (left) and FEM (right).	87
Figure 5-14: Kinetic energy plot. A, B, C curves represent the FE model, from coarse to fine mesh, D, E, F curves represent IG model from coarse to fine mesh.	88
Figure 5-15: Spotwelds failure at time $t=10\text{ms}$, only two beam elements didn't fail. In this model a SPOTWELD contact is used.	88
Figure 5-16: Automotive crash structure.	89
Figure 5-17: FE model (up), IG model (low).	89
Figure 5-18: Example of NURBS contact errors.	90
Figure 5-19: FEM solution.	90
Figure 5-20: Simplified geometry.....	91
Figure 5-21: FE model (up), IG model (down).	91
Figure 5-22: Particular of the badly trimmed patch (the problem is spotted on both crashboxes).	91

Figure 5-23: IGA solution.....	92
Figure 5-24: FEM solution.	92
Figure 5-25: Particular of the kinetic energy plot. FEM (red) and IGA (green). ..	92
Figure 5-26: Set up of the implicit solution via CONTROL_IMPLICIT_GENERAL card.....	93
Figure 5-27: Stress distribution in the instant of max stress. Implicit (left) and explicit (right).	94
Figure 5-28: Displacement of the excited node. Explicit (red) and implicit (green).	94
Figure 5-29: Stress of the most stressed interpolated element. Explicit (red) and implicit (green).	95
Figure 5-30: CONTROL_IMPLICIT_EIGENVALUE card.	95
Figure 5-31: The first six modes in IG and FE model.	96
Figure 5-32: Plot of the frequency over one hundred modes.	96
Figure 5-33: Wheel rim IGA model (left) and geometry (right).	97
Figure 5-34: Example of the warning messages appeared for all the trimmed NURBS patches.	97
Figure 5-35: Solid NURBS model.....	98
Figure 5-36: Plot of the frequency over twenty modes.	98
Figure 5-37: The first six modes in IG and FE model.	99
Figure 6-1: MAT-022 keyword, AOPT with A and D directions vector are highlighted.	100
Figure 6-2: AOPT values explanation. [28].....	102
Figure 6-3: Definition of composite layers from SECTION_SHELL keyword..	103
Figure 6-4: Composite interface with graphical rendering of the ply thickness..	104
Figure 6-5: Ply orientation definition.	104
Figure 6-6: ELEMENT_SHELL_COMPOSITE keyword.....	105
Figure 6-7: INTEGRATION_SHELL keyword.	105
Figure 6-8: User defined shell integration rule. [28]	106
Figure 6-9: Kinetic energy plotted over time. IGA quadratic (green), IGA cubic (blue) and FEA (red).	108

Figure 6-10: Z-displacement of the central node of the plate plotted over time. IGA quadratic (green), IGA cubic (blue) and FEA (red).	109
Figure 6-11: FE model with visible thickness.	109
Figure 6-12: Particular of the plies.	110
Figure 6-13: IGA - The simulation at time $t=61s$ (above) and $t=62s$ (below).....	111
Figure 6-14: FEM – The simulation at time $t=100s$	111
Figure 6-15: Z-force plotted over time, IGA (green) and FEA (RED).....	112
Figure 6-16: FE analysis termination time changed to 200s.	112
Figure 7-1: INITIAL_STRESS_NURBS_PATCH keyword. [60].....	113
Figure 7-2: Von mises stress distribution at the end of stage one (left) and at the beginning of stage 2 (right). [16]	114
Figure 7-3: CONTROL_FORMING_ONESTEP keyword.....	114
Figure 7-4: Comparison of the Von Mises stress obtained via the one-step approach, FEA (left) and IGA (right). [31]	115
Figure 7-5: Comparison of the effective plastic strain fields obtained via the one-step approach, FEA (left) and IGA (right). [90]	115
Figure 7-6: Comparison of the effective plastic strain fields obtained via the one-step approach, FEA (left) and IGA (right). Second example (trimmed NURBS patch). [31].....	116
Figure 7-7: Stage 1 (Gravity). FEA (left) and IGA (right). Results are then mapped and loaded into the next step. [60]	117
Figure 7-8: Effective plastic strain fields after the deep drawing process. FEA (left) and IGA (right). [16].....	117
Figure 7-9: Effective plastic strain fields comparison after trimming, FEA (left) and IGA (right). [16].....	118

REFERENCES

- [1] P. Bezier. D'efinition num'erique des courbes et surf'aces i. Automatisme, XI:625– 632, 1966.
- [2] P. Bezier. D'efinition num'erique des courbes et surf'aces ii. Automatisme, XII:17– 21, 1967.
- [3] P. Bezier. Numerical Control: Mathematics and Applications. Wiley, 1972.
- [4] S. Bernstein. D'emonstration du th'eorem de weierstrass fonde'e sur le calcul des probabilit'es. Harkov Soobs. Matem ob-va, XIII:1–2, 1912.
- [5] I. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Quarterly of Applied Mathematics, IV:45–99, 1946.
- [6] H. Curry and I. Schoenberg. On po'lya frequency functions iv: The fundamental spline functions and their limits. Journal d'Analyse Mathematique, XVII:71–107, 1966.
- [7] R.F. Riesenfeld. Application of B-spline Approximation to Geometric Problems of Computer Aided Design. PhD thesis, Syracuse University, 1972.
- [8] K.J. Vesprille. Computer-aided Design Applications of the Rational B-spline Approximation form. PhD thesis, Syracuse University, 1975.
- [9] T.N. Sederberg, J.M. Zhengs, A. Bakenov, and A. Nasri. T-splines and tnurccss. ACM Transactions on Graphics, III:477–484, 2003.
- [10] T.W. Sederberg, T.L. Cardon, G.T. Finigan, N.S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. ACM Transactions on Graphics, III:276–283, 2004.

- [11] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," *Bulletin of American Mathematical Society*, vol. 49, pp. 123, 1943.
- [12] R. Clough, "The Finite Element Method, in Plane Stress Analysis," in *Proc. 2nd A.S.C.E. Conf: on Electronic Comp.*, Pittsburgh, PA, 1960.
- [13] M. F. Hardwick, R. L. Clay, P. T. Boggs, E. J. Walsh, A. R. Larzelere, and A. Altshuler, "DART system analysis," *Sandia National Laboratories, Tech. Rep. SAND2005-4647*, 2005.
- [14] T. J. 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*, 2005.
- [15] Stefan Hartmann , David J. Benson, David Lorenz, *About Isogeometric Analysis and the new NURBS-based Finite Elements in LS-DYNA*, 2011
- [16] S. Hartmann, D. J. Benson, L. Li, A.P. Nagy, I. Yeh: *Sheet metal forming simulations with IGA in LS-DYNA*. (2018)
- [17] L. Piegl and W. Tiller, *The NURBS Book*, Springer, 1997.
- [18] J. A. Cottrell, T. J. Hughes and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, Wiley, 2009.
- [19] E. Cohen, T. Lyche, and R. Riesenfeld. Discrete b-spline and subdivision techniques in computer aided geometry design and computer graphics. *Computer Graphics and Image Processing*, XIV:87–111, 1980.
- [20] L. Ramshaw. Blossoming: a connect-the-dots approach to splines. Technical report, Digital Systems Research Center, Palo Alto, CA, 1987.
- [21] L. Ramshaw. Blossoms are polar forms. *Computer Aided Geometric Design*, VI:323–358, 1989.
- [22] S. B. Raknes, *Isogeometric Analysis and Degenerated*, Trondheim: Norwegian University of Science and Technology, 2011.

- [23] Özdoğan, Yasin, Analysis of metal forming by using isogeometric elements, 2018.
- [24] M. Schwedler, C. Könke, E. Rank, K. Beucke, Integrated structural analysis using isogeometric finite element methods, 2016.
- [25] D. F. Rogers. An introduction to NURBS: with historical perspective. Morgan Kaufmann Publishers Inc., 2001.
- [26] P. E. Bézier. How Renault uses numerical control for car body design and tooling. Technical report, SAE Technical Paper, 1968.
- [27] S. Hartmann, L.F. Leidenger, D. J. Benson, L. Li, A.P. Nagy, M. Pigazzini: Enabling the Analysis of Topologically Connected Multi-Patch Trimmed Nurbs Shells in LS-DYNA, 2019.
- [28] LIVERMORE SOFTWARE TECHNOLOGY CORPORATION (LSTC), LS-DYNA® KEYWORD USER'S MANUAL VOLUME I, 2018.
- [29] S. Hartmann, D. J.Benson, L. Li, A.P. Nagy, M. Montanari, N. Petrinic: Recent Developments in Isogeometric Analysis with Solid Elements in LS-DYNA, 2018.
- [30] S. Hartmann, A. P. Nagy, D. J. Benson, Forming simulations with NURBS shells in LS-DYNA, 2018.
- [31] L. Li, X. Zhu, J. Zheng, D. J. Benson, A.P. Nagy, S. Hartmann, A one step simulation approach using isogeometric shells in LS-DYNA, 2018.
- [32] M.G. Cox. The numerical evaluation of b-splines. Technical report, National Physics Laboratory DNAC 4, 1971.
- [33] C. De Boor. On calculation with b-splines. Journal of approximation theory, VI:50–62, 1972.
- [34] C. De Boor. A practical guide to Splines. Springer-Verlag, 1978.