About Isogeometric Analysis and the new NURBS-based Finite Elements in LS-DYNA

Stefan Hartmann[1], David J. Benson[2], David Lorenz[1]

[1] DYNAmore GmbH, Stuttgart, Germany

[2] Department of Structural Engineering, University of California, San Diego, USA

Summary:

In the context of isogeometric analysis many research activities have focused on the use of Non-Uniform Rational B-Splines (NURBS). These NURBS-based finite elements have been studied in depth and it has been shown, that they are particularly well suited for computational analysis leading to qualitatively more accurate results in comparison with standard finite elements based on Lagrange polynomials. Due to these motivating results, NURBS-based finite elements are currently implemented into LS-DYNA.

This work outlines the basic ideas of isogeometric analysis and gives a short introduction into NURBS basis functions. The new keyword *ELEMENT_NURBS_PATCH_2D available in LS-DYNA is presented together with various possible options, like shell theories with and without rotational degrees of freedom. Preliminary results on the performance of these new elements are studied by means of a sheet metal forming example discussed in the Numisheet conference 2005.

Keywords:

isogeometric analysis, NURBS, rotation-free shell elements

1 Introduction

The goal of integrating computer aided design (CAD) and finite element analysis (FEA) has led to a new computational method called *Isogeometric Analysis* (IGA). Its main idea is to use the same mathematical description for the geometry in the design (CAD) and the analysis (FEA). Much of the recent research on isogeometric analysis uses Non-Uniform Rational B-Splines (NURBS) as basis functions, as this geometrical representation is the most widely used in engineering design systems. It has been shown that NURBS-based finite elements are very well suited for computational analysis leading to qualitatively more accurate results in comparison with standard finite elements based on Lagrange polynomials. Due to these motivating results, NURBS-based finite elements are currently implemented into LS-DYNA.

The paper will be organized as follows:

In section 2 the basic ideas of isogeometric analysis will be presented followed by a short introduction into NURBS. Section 3 describes how to define a NURBS patch with the new keyword *ELEMENT_NURBS_PATCH_2D and explains the characteristics of interpolation-nodes and -elements. Preliminary results of the NURBS-based finite elements in LS-DYNA will be discussed by means of a sheet metal forming example taken from the Numisheet 2005 conference in section 4. Section 5 closes with some conclusions and an outlook.

2 Isogeometric analysis with NURBS

In this section a rough overview about isogeometric analysis will be given followed by a short introduction into NURBS. This outline is far from being complete and the interested reader is referred to the interesting textbook of Cottrell et al. [1] for isogeometric analysis and the monograph by Piegel and Tiller [2] for the details on NURBS.

2.1 Isogeometric analysis in short

2.1.1 Motivation

The initial motivation to come up with isogeometric analysis was the wish to reduce the amount of work spent to convert the geometry description from computer aided design (CAD) into a suitable mesh for finite element analysis (FEA). 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. This necessitates a re-parameterization of the CAD geometry which is labor cost intensive and results in most cases in a certain amount of discretization errors. The basic idea of isogeometric analysis is to use the same geometry description in design and analysis. 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 [3]. 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 structure interaction, shape and topology optimization, vibration analysis, buckling and others) where many studies were using NURBS as basis functions. A very interesting introduction to isogeometric analysis can be found in [1].

2.1.2 Definition

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 analogy to the well known term “isoparametric” in FEA, the name “isogeometric” is chosen. In Fig. 1 these two notions are given side by side.

ISOPARAMETRIC (in finite element analysis)
same approximation for the geometry and the deformation

GEOMETRIE	←→	DEFORMATION

ISOGEOMETRIC (CAD - FEA)
same description of the geometry in the design (CAD) and the anaylsis (FEA)

CAD	←→	FEA

Fig. 1: Isoparametric vs. Isogeometric

2.2 From B-splines to NURBS

Various studies have shown that NURBS basis functions are particularly well suited for FEA and therefore a rough overview on B-Splines and NURBS will be given in the following.

2.2.1 B-spline basis functions

B-spline basis functions (see Fig. 2) are constructed recursively until the desired polynomial degree of the functions is reached (see Equ. (1)).

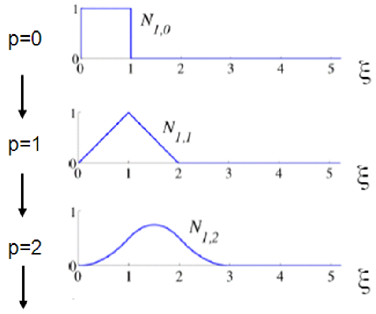


Fig. 2: B-spline basis functions of order 0, 1 and 2 for uniform knot vector [1]

A key ingredient for the recursion formula is the so-called "*knot-vector*" $\Xi = \left[\xi_1, \xi_2, ..., \xi_{n+p+1}\right]$, which is a non-decreasing set of coordinates in the parameter space, where ξ_i is the ith knot, p is the polynomial order and n is the number of basis functions.

$$\begin{aligned} &for\ p=0: \quad N_{i,0}(\xi) = \begin{cases} 1 & if\ \ \xi_i \le \xi < \xi_{i+1} \\ 0 & otherwise \end{cases} \\ &for\ \ p>0: \quad N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1(\xi)} + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1(\xi)} \end{aligned} \tag{1}$$

In contrast to the widely used Lagrange polynomials in FEA, B-spline basis functions are always positive regardless of their polynomial order. Furthermore they constitute the important partition of unity property and in absence of multiple knot values in the knot-vector they exhibit a C^{p-1}-continuity along the internal element boundaries (knot values).

2.2.2 B-spline curves

The creation of B-spline curves is similar to standard FEA. Instead of nodal coordinates in classical FEA, so-called *control points* $\boldsymbol{B}_i$ are used as coefficients of the B-spline basis functions. It is important to point out that the control points, in opposition to the nodal coordinates, are normally not a part of the actual geometry. This comes from the non-interpolatory nature of the B-spline basis functions. Taking a linear combination of the B-spline basis functions with the corresponding control points defines a B-spline curve $\boldsymbol{C}(\xi)$.

$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\boldsymbol{B}_i \tag{2}$$

A typical B-spline curve is shown in Fig. 3 together with the control points, the resulting control polygon and the B-spline basis functions.

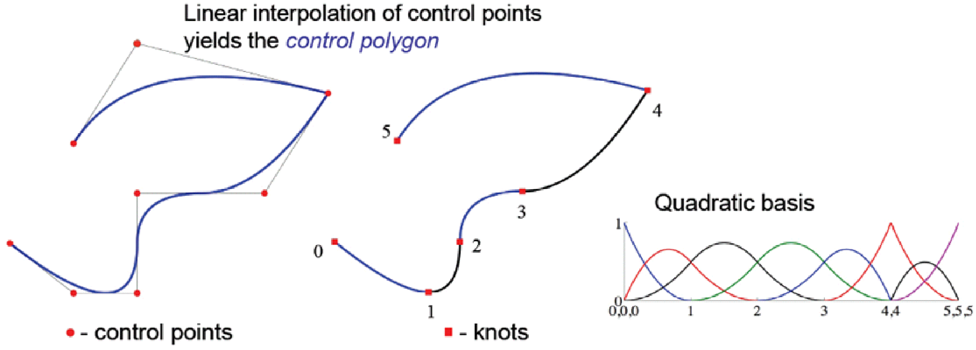


Fig. 3: B-spline curve with control points, knots and associated B-spline basis functions [1]

2.2.3 Refinements

Various types of refinements exist for B-spline curves without changing the initial curve, namely h-, p- and k-refinement. While h- and p-refinement can be directly compared to classical FEA, k-refinement displays an additional refinement possibility when using B-spline basis functions, which allows to generate a higher order of continuity.

2.2.4 NURBS

The main issue to go from B-splines to NURBS is the fact that a wide range of geometrical objects can be exactly represented with NURBS. A NURBS is constructed by a projective transformation of a B-spline. Technically this is achieved by introducing weights at the control points that allows more control over the actual shape of the NURBS. The NURBS basis functions $R_i^p(\xi)$ are computed by multiplying the B-spline basis functions with the appropriate weights w_i and dividing the product by the weighting function $W(\xi)$.

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)\,w_i}{W(\xi)} \quad with \quad W(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\,w_i \qquad (3)$$

A NURBS curve is then represented in the same way as a B-spline curve by taking a linear combination of the NURBS basis functions with the associated control points.

$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\,\boldsymbol{B}_i \qquad (4)$$

In Fig. 4 a two NURBS curves are shown that differ only in the value of the weight at the 9th control point.

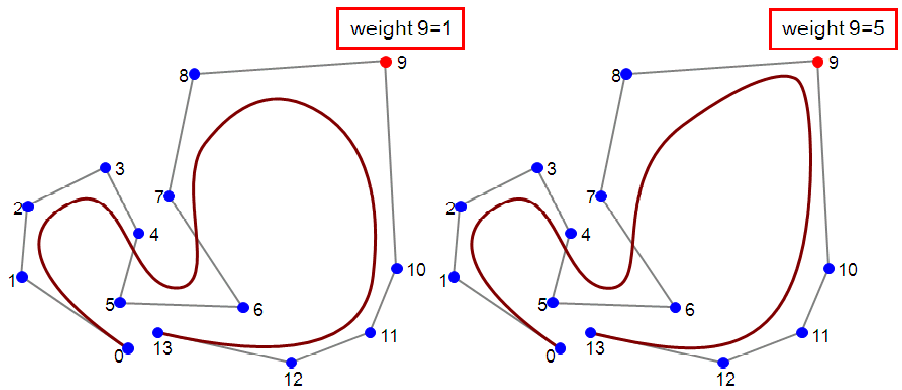


Fig. 4: NURBS curves with identical control points and knot-vectors - only difference weight

Up to now, only 1D curves have been discussed, whereas most of the applications as well in CAD as in FEA use 2D surfaces in space. The construction of NURBS surfaces is straightforward using a tensor product structure based on 1D basis functions to define the necessary basis functions.

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)\,M_{j,q}(\eta)\,w_{i,j}}{W(\xi,\eta)} \quad with \quad W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)\,M_{j,q}(\eta)\,w_{i,j} \qquad (5)$$

A NURBS surface is then again defined through a linear combination of these basis functions with the connected control points. Further details on the construction of NURBS surfaces, the derivatives of NURBS basis functions, useful algorithms and much more can be found in the monograph by Piegl and Tiller [2].

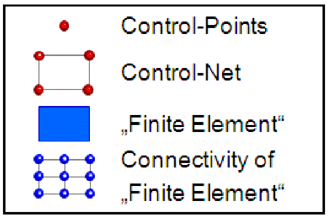
3 FEA with NURBS for thin-walled structures in LS-DYNA

The following section will introduce the new keyword *ELEMENT_NURBS_PATCH_2D and the present possibilities in LS-DYNA to use isogeometric analysis.

3.1 A typical NURBS-patch

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. 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 new keyword *ELEMENT_NURBS_PATCH_2D is depicted in Fig. 5 together with the resulting subdivision into “finite elements”.

```
*ELEMENT_NURBS_PATCH_2D
$---+--EID----+--PID----+--NPR----+----PR----+--NPS----+----PS----+----7----+----8
        11        12         4         2         5         2
$---+--WFL----+-FORM----+--INT----+-NISR----+-NISS----+IMASS----+----7----+----8
         0         0         1         2         2         0
$rk-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
       0.0       0.0       0.0       1.0       2.0       2.0       2.0
$sk-+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
       0.0       0.0       0.0       1.0       2.0       3.0       3.0       3.0
$net+---N1----+---N2----+---N3----+---N4----+---N5----+---N6----+---N7----+---N8
         1         2         3         4
         5         6         7         8
         9        10        11        12
        13        14        15        16
        17        18        19        20
```



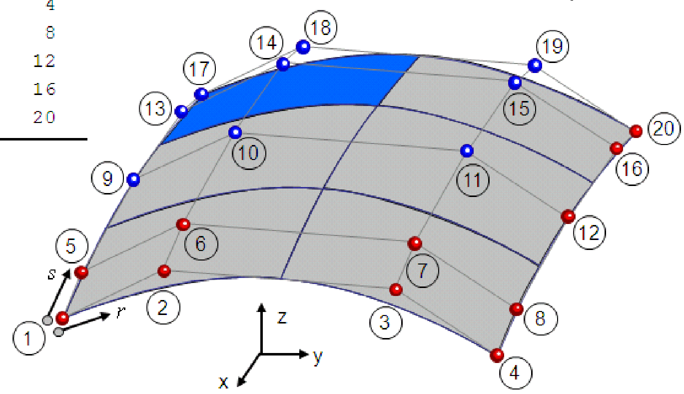


Fig. 5: The keyword *ELEMENT_NURBS_PATCH_2D in LS-DYNA

In this particular example, quadratic NURBS basis functions are chosen in local r- and s-direction. This leads to 9-noded quadratic “finite elements” for the analysis which are C^1-continuous along the element boundaries. Standard finite elements based on Lagrange polynomials are always C^0-continuous along the element boundaries. The higher continuity of the “NURBS-finite-elements” originates from the non-interpolatory nature of the NURBS-basis functions and leads to an overlap of control points of adjacent elements.

There is no limit in the size of a NURBS-patch definition within the keyword *ELEMENT_NURBS_PATCH_2D, such that the parameters NPR (number of control points in local r-direction), NPS (number of control points in local s-direction), PR (polynomial order in local r-direction) and PS (polynomial order in local s-direction) define the actual size of this keyword. The necessary knot-vectors are defined in the parameters RK and SK and the connectivity of the whole NURBS-patch follows afterwards.

3.2 Interpolation-nodes and -elements

Up to now, 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. 6 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.

```
...
$---+--WFL----+-FORM----+--INT----+-NISR----+-NISS----+IMASS
        0          0          1          2          2          0
```

- NISR/NISS – Number of Interpolation Elements per Nurbs-Element (r-/s-dir.)
  important for post-processing, boundary conditions and contact treatment

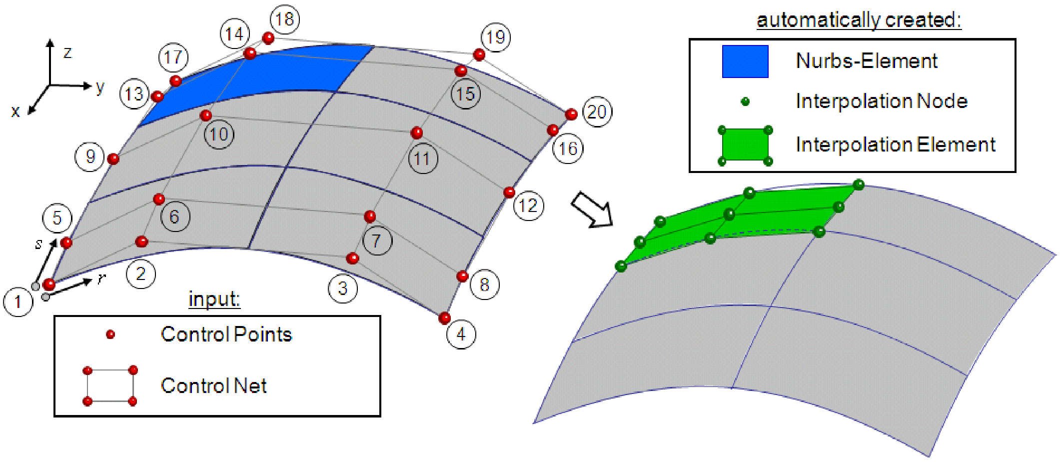


Fig. 6: Interpolation-nodes and -elements

3.3 Available shell formulations

At present four 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_NURBS_PATCH_2D, with the following code:

- 0: “shear deformable theory” with rotational degrees of freedom [4]
- 1: “shear deformable theory” without rotational degrees of freedom [5]
- 2: “thin shell theory” without rotational degrees of freedom
- 3: “thin shell theory” with rotational degrees of freedom

The shear deformable theory is based on the degenerated solid element formulation [6]. 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 [7] element,

$$x(\eta,\xi,\zeta)=\sum_i N_i(\eta,\xi)\left(x_i+\tfrac{h}{2}\zeta\hat{y}_i\right) \tag{6}$$

and the velocity is expressed as

$$\dot{x}(\eta,\xi,\zeta)=\sum_i N_i(\eta,\xi)\left(\dot{x}_i+\tfrac{h}{2}\zeta\dot{\hat{y}}_i\right). \tag{7}$$

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

$$\dot{\hat{y}}_i=\omega_i\times\hat{y}_i \tag{8}$$

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,

$$\dot{\hat{y}}_i=\sum_j\frac{\partial\hat{y}_i}{\partial x_j}\dot{x}_j\,. \tag{9}$$

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

$$x(\eta,\xi,\zeta)=\sum_i N_i(\eta,\xi)x_i+\tfrac{h}{2}\zeta\hat{n}(\eta,\xi). \tag{10}$$

Differentiating with time gives the velocity,

$$\dot{x}(\eta,\xi,\zeta) = \sum_i N_i(\eta,\xi)\dot{x}_i + \tfrac{h}{2}\zeta\dot{\hat{n}}(\eta,\xi). \tag{11}$$

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.

3.4 Analysis capabilities

Several analysis capabilities are already implemented for the new NURBS-based finite elements in LS-DYNA, like:

- implicit and explicit time integration
- eigenvalue analysis
- many material models from the LS-DYNA material library are available

4 Preliminary results – Numisheet-Benchmark 2005

Some preliminary results with the NURBS-based finite elements in LS-DYNA will be presented in the following by means of a sheet metal forming application.

4.1 Example setup

The Numisheet 2005 Benchmark on "*Forming of an Automotive Underbody Cross Member*" (BM2) [8] has been chosen to study the performance of the new NURBS-based finite elements in LS-DYNA. In Fig. 7 the setup of the forming process for the underbody cross member is shown.

The numerical simulation of this problem using fully integrated standard shell elements (ELFORM=16) with three possible steps of adaptive refinement (4mm → 2mm → 1mm) was chosen as reference solution.

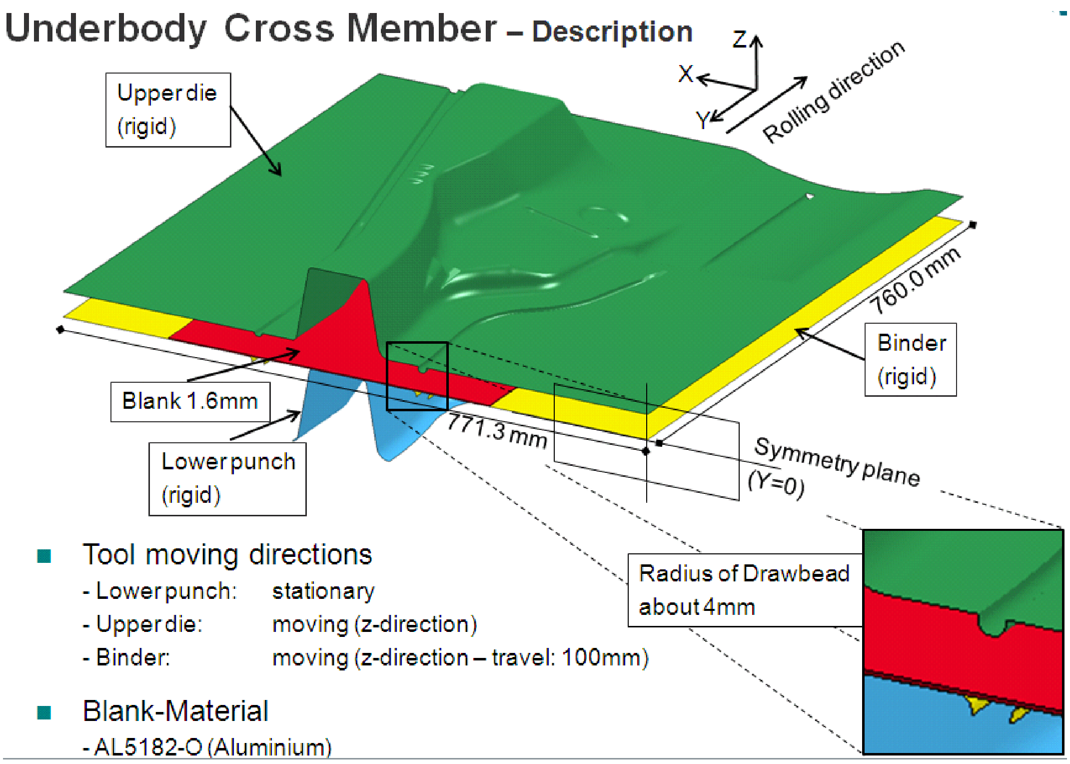


Fig. 7: Numisheet 2005 - BM2 setup

To compare the performance of the NURBS-based finite elements, this example was computed with different configurations of mesh sizes and polynomial orders. A summary of all analyzed configurations is given in Table 1. The model setup for the standard shell elements and the NURBS-based finite elements only differ in the discretization of the blank. All the numerical simulations were performed using the material model *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC (*MAT_037) with five integration points through the thickness of the shells. No thickness update was taken into account (istupd=0) and no mass scaling was considered. The computations were run on a Dual Core AMD Opteron, 2.2 GHz in SMP, double precision with ncpu=4.

The behavior of the NURBS-finite elements was compared with the results achieved with fully integrated standard shell elements (ELFORM=16).

	Fully integrated linear shell elements (ELFORM=16)	2D NURBS finite elements (FORM=2, rotation free)
Reference solution	Adaptivity (4mm → 2mm → 1mm)	--------------------
Discretizations	8mm 4mm 2mm	16mm 8mm 4mm
Polynomial order	P=1 (linear)	P=2 (quadratic) P=3 (cubic) P=4 (quartic) P=5 (quantic)

Table 1: Overview about different computations

4.2 Comparison

The quality of the results is compared on basis of the final draw-in of the blank at six different positions (see Fig. 8).

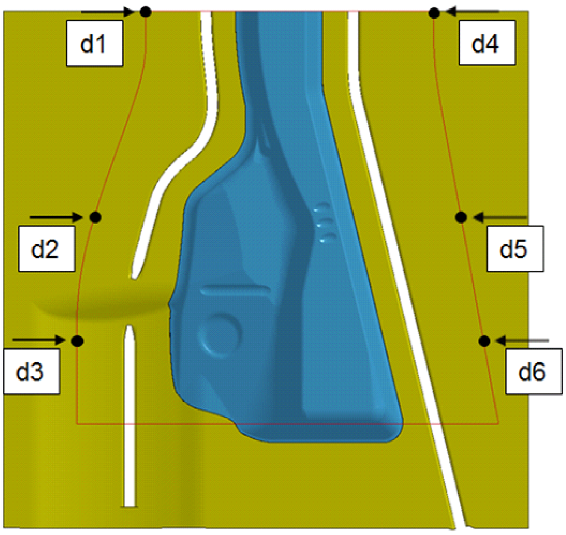


Fig. 8: Measured draw-in positions

The results for the draw-in length for all analyzed configurations are depicted in Fig. 9. Good results with respect to the reference solution (standard shell elements with adaptivity) are achieved with standard shell elements with an average element size of 2mm and with NURBS-based shell elements with an average element size of 4mm. Larger mesh sizes lead to a rather stiff behavior on the left side which is indicated by less draw-in. Furthermore it can be realized, that for this particular example, a higher polynomial order will not lead to better results. This can be explained by the fact, that the spacing of the control points will not change significantly when going to higher order polynomials and the maximum spacing of the control points to achieve qualitatively good results is dictated by the radius of the drawbead (approximately 4mm, see Fig. 7).

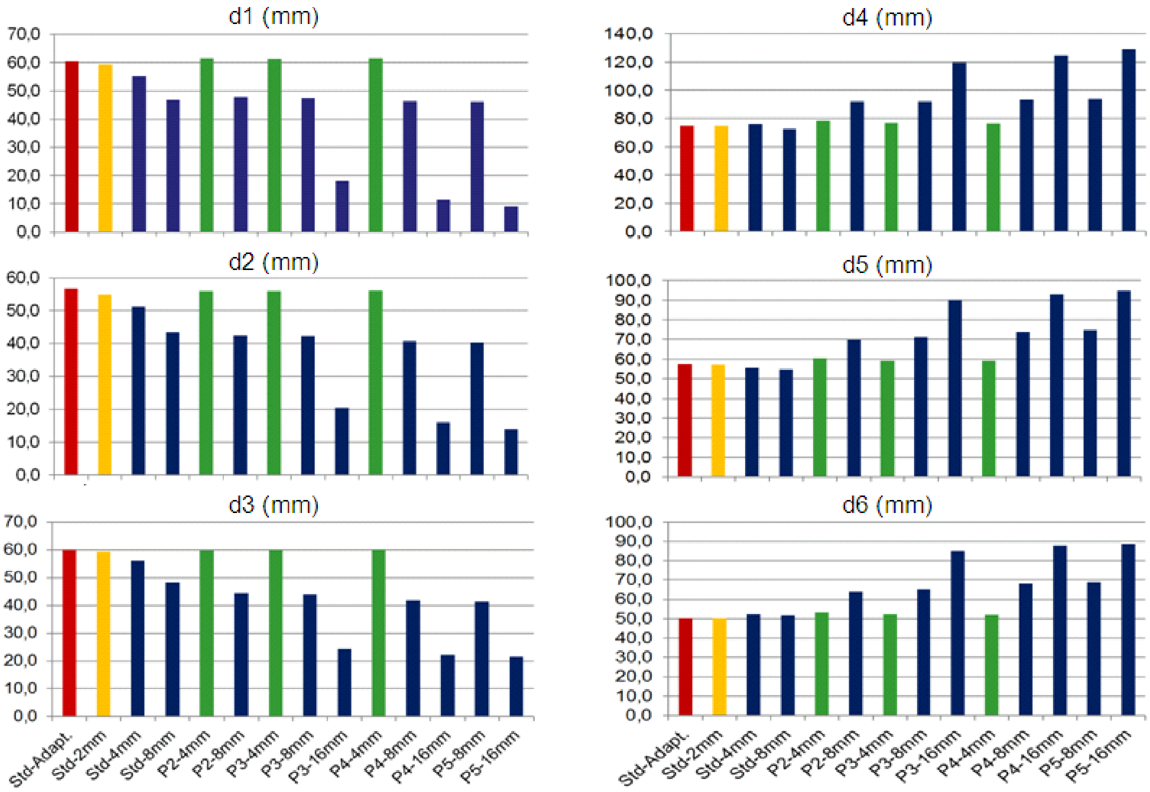


Fig. 9: Comparison of draw-in at measured positions

Besides the draw-in measurements the total contact forces in the interface between the upper die and the blank are compared. The results shown in Fig. 10 confirm the general behavior noticed in the draw-in comparisons. NURBS-based shell elements with an average mesh size of 4mm exhibit a qualitatively comparable result as standard linear shell elements with an average mesh size of 2mm.

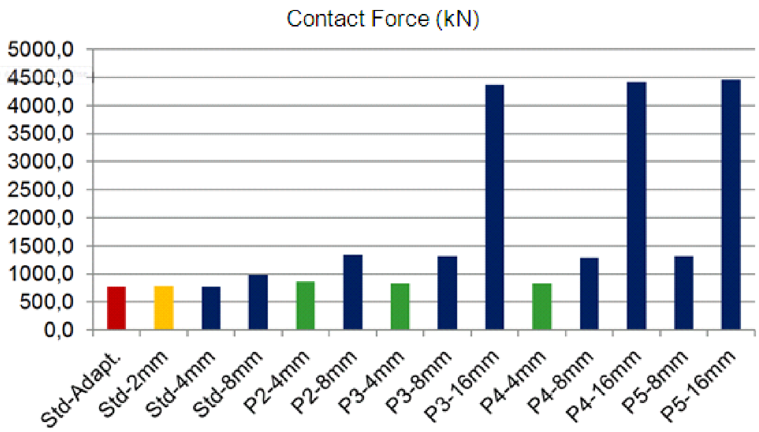


Fig. 10: Comparison of contact forces between upper die and blank

On top of the qualitative comparisons, the necessary CPU-time is one of the most important issues. The results in Fig. 11 show that the analysis with quadratic NURBS elements with a mesh size of 4mm (P2-4mm) runs about 30% faster than the one with linear standard shell elements with a mesh size of 2mm (Std-2mm). Furthermore it can be seen that raising the polynomial order of the elements will lead to an increase in computation cost of factor 2.5-3.0 (see for example P2-4mm → P3-4mm → P4-4mm in Fig. 11).

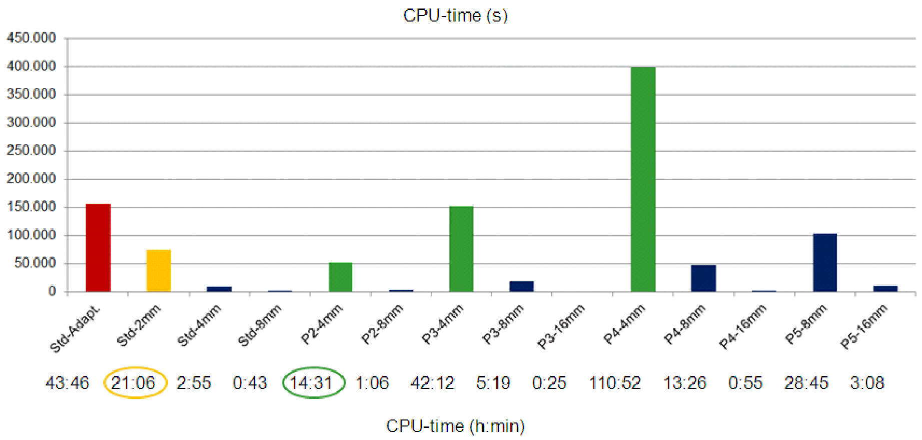


Fig. 11: Comparison of necessary CPU-time

4.3 Summary

This particular example has shown that NURBS shell elements will lead to qualitatively as good results as a comparable discretization with standard linear finite elements (e.g. P2-4mm ←→ Std-2mm) and that they are cost competitive. The same accuracy of the results can be obtained with 2/3 the CPU time. Note that this comparison is based on an early stage of the implementation of the NURBS-based finite elements in LS-DYNA where nearly no effort has been spent on CODE optimization whereas the implementation of the standard finite elements has been highly optimized during the last years. Therefore this result is even more promising.

5 Conclusion and Outlook

The basic ideas of isogeometric analysis have been introduced along with a rough overview about B-splines and NURBS. The new keyword *ELEMENT_NURBS_PATCH_2D that allows to perform isogeometric analysis with LS-DYNA was discussed. Preliminary results with the new NURBS-based finite elements have shown that the current implementation runs stable and that higher order accurate isogeometric analysis can be cost competitive even without a heavily optimized implementation.

A lot more studies need to be performed to figure out the areas in which NURBS-based finite elements could be the first choice in future FEA. Furthermore the present implementation needs to be optimized and parallelized to make them work in MPP. Many features available for standard finite elements in LS-DYNA need to be added for the NURBS-based finite elements in the near future.

6 References

[1] Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.: "Isogeometric Analysis – Toward Integration of CAD and FEA", John Wiley & Sons, Ltd, 2009

[2] Piegel, L., Tiller, W.: "The NURBS Book", Springer-Verlag(Berlin Heidelberg), 1995

[3] Hughes, T.J.R., Cotrell, J.A., Bazilevs, Y.: "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement", *Computer Methods in Applied Mechanics and Engineering*, 194, 2005, 4135-4195

[4] Benson, D.J., Bazilevs, Y., De Luycker, E., Hsu, M.-C., Scott, M., Hughes, T.J.R, Belytschko, T.: "A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to XFEM", *International Journal for Numerical Methods in Engineering*, 83, 2010, 765-785

[5] Benson, D.J., Bazilevs, Y.' Hsu, M.-C.' Hughes, T.J.R.: "A large deformation, rotation-free, isogeometric shell", *Computer Methods in Applied Mechanics and Engineering*, 200, 2011, 1367-1378

[6] Hughes, T.J.R.: "The Finite Element Method: Linear Static and Dynamic Finite Element Analysis", Prentice-Hall, 1987.

[7] Belytschko, T., Liu, W.K., Moran, B.: "Nonlinear Finite Elements for Continua and Structures", Wiley, 2000.

[8] "NUMISHEET 2005: On the Cutting Edge of Technology – The Numisheet 2005 Benchmark Study – Part B", Cao, J., Shi, M.F., Stoughton, T.B., Wang, C.-T., Zhang, L. (eds.), August 15-19, 2005, Detroit, MI, USA