

Sheet Metal Forming Simulation with IGA in LS-DYNA[®]

Stefan Hartmann

DYNAmore GmbH, Stuttgart, Germany

David J. Benson

Liping Li

Attila P. Nagy

Livermore Software Technology Corporation, Livermore, CA, USA

Abstract

In the last few years, numerous research work has been devoted to Isogeometric Analysis (IGA). IGA is a finite element technology in which computer-aided design (CAD) geometric description is invoked to perform numerical analysis. The most widely used mathematical description in CAD is non-uniform rational B-splines (NURBS) and therefore NURBS-based shell and solid finite elements have been implemented into LS-DYNA.

This paper describes the recent advances of the NURBS-based shell implementation in LS-DYNA to enable the IGA technology for the use in sheet metal forming applications. Necessary features like stress, strain, thickness, and history variable mapping from one stage to the other, the trimming of the formed part and other typical features used by the forming analysts have been enabled for the use with NURBS-shells. The new keywords will be explained and the multistage forming process will be analyzed by means of an example. A comparison with current state-of-the art methods is provided and further developments are outlined.

1 Introduction to Isogeometric Analysis

This section introduces the rather new finite element technology, called isogeometric analysis (IGA). The term IGA was introduced by Hughes et al. [1] in 2005, in analogy to the term “isoparametric”. While the standard “isoparametric” approach in finite element analysis indicates that the geometry representation as well as its deformed solution space is approximated using the identical shape functions, which are in general low order Lagrange polynomials. The “isogeometric” idea goes one step further and states that the geometry description used in the computer aided design (CAD) shall be used in the analysis as well. One of the largest initial motivations for developing IGA was the hope to better integrate the CAD-models with the subsequent finite element analysis in order to cut down significantly the labor time needed to reparametrize (mesh) the CAD geometry for doing the analysis. Although this particular possible advantage couldn’t be compellingly proven yet, the use of higher order shape functions, i.e. NURBS may yield better results while having the possibility of using larger element-sizes. Furthermore, the use of the IGA technology may help to reduce the discretization error that may result from the re-parameterization of the CAD design. A schematic comparison of the meshing procedure between standard finite elements and isogeometric analysis is shown in Fig.1. It 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 by doing mesh-refinement. With the IGA approach, the initial CAD geometry can be directly used for analysis and necessary mesh-refinements for enlarging the solution space will leave the geometry unchanged.

Various mathematical descriptions are used in the different CAD packages, but amongst them NURBS play a dominant role. That is why many researches in the area of IGA focus on NURBS and so does the IGA implementation in LS-DYNA.

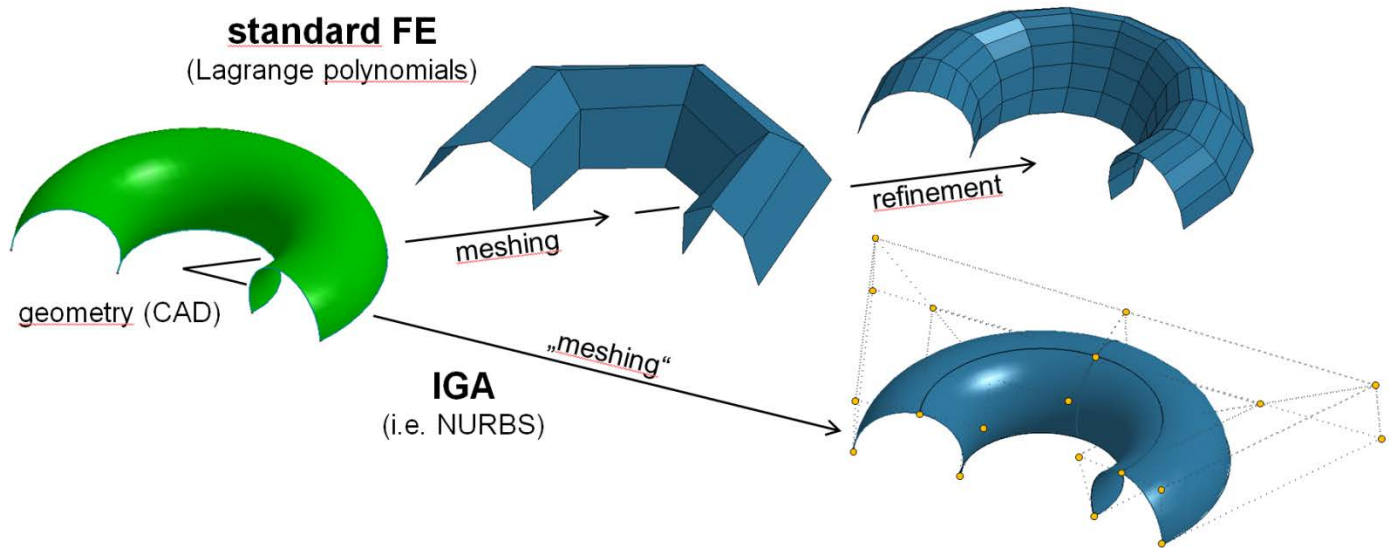


Fig.1: Comparison of meshing for standard finite elements and IGA

2 NURBS

Some basic properties of NURBS will be presented in order to be able to understand some significant differences of using NURBS instead of Lagrange polynomials for finite element analysis. As the few subsequent descriptions may only give a rough idea about NURBS, the interested reader is referred to the monograph by Piegl and Tiller [2].

2.1 B-Splines

Given the name Non-uniform rational B-Splines it is obvious that NURBS are built from B-Splines. B-Spline basis functions are constructed in a recursively manner, starting with a constant basis function and then increasing the order in every recursive step until the desired degree is reached (see Fig.2)

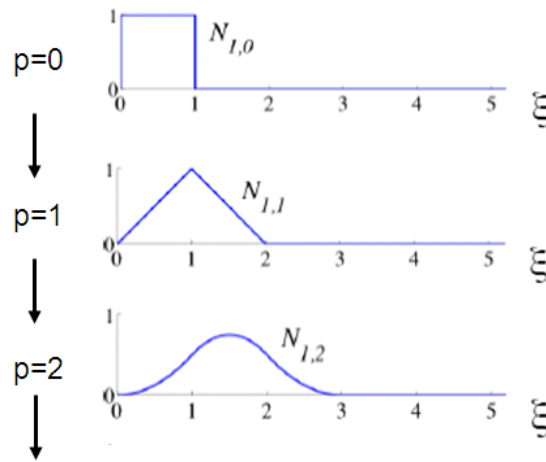


Fig.2: B-spline basis functions of order 0, 1 and 2 for uniform knot vector [3]

The recursion formula is given by

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

In here ξ_i is the i^{th} knot of the so-called “knot-vector” $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$, which is a non-decreasing set of coordinates in the parametric space. The degree of the basis functions is given with p and finally n represents the number of basis functions defined through the knot-vector. It has to be noted that B-spline basis functions are always and everywhere positive regardless of their degree, which is a significant difference compared to higher order Lagrange Polynomials (see Fig.3). Furthermore B-Spline basis functions constitute the important partition of unity property and exhibit a C^{p-1} -continuity along the internal element boundaries if no multiple knot values are present in the knot-vector.

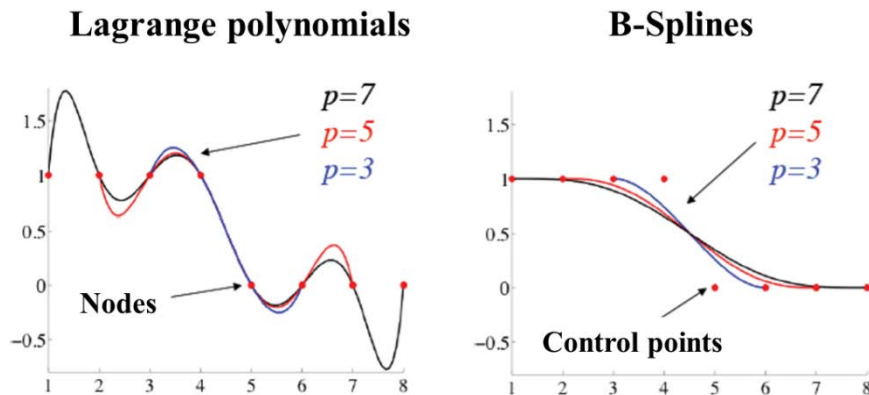


Fig.3: Comparison of Curves represented by Lagrange polynomials (left) and B-spline basis functions(right)given the same set of nodes and control points, respectively [3]

The construction of B-spline curves is similar to the way this is done using Lagrange polynomials, with one important difference. Instead of interpolating the curve through the nodal coordinates (see Fig.3 left), B-spline curves use so-called *control points*, which are used as coefficients of the B-spline basis functions. As can be seen in Fig.3 (right), these control points are most of the time not a part of the actual geometry which is due to the non-interpolatory nature of the B-Spline basis functions. A B-spline curve $C(\xi)$ is defined through a linear combination of the B-spline basis functions with the corresponding control points B_i .

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

The step from B-splines to NURBS is achieved by introducing an additional parameter to every control point which is called a weight. Using the weights w_i at the control points, the NURBS basis functions $R_i^p(\xi)$ are constructed as follows:

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

A NURBS curve is then defined in the same way as a B-spline curve, by substituting the B-spline basis functions in Equ. (2) with the NURBS basis functions in Equ. (3).

2.2 NURBS surfaces

The step from defining NURBS curves to NURBS surfaces is straight forward and can be easily extended to define NURBS solids. Starting with the univariate B-spline basis functions discussed in the preceding section, the necessary NURBS basis functions to finally describe a NURBS surface are constructed using a tensor product on these univariate basis functions and combine them with the weights at the control points. In mathematical terms the bi-variate NURBS basis functions are defined as follows:

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{W(\xi, \eta)} \text{ with } W(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j} \tag{4}$$

Given the bi-variate NURBS basis functions, NURBS surfaces are then constructed in a similar way as NURBS or B-spline curves through a linear combination of these basis functions with their corresponding control points. A typical NURBS surface is shown in Fig. 4.

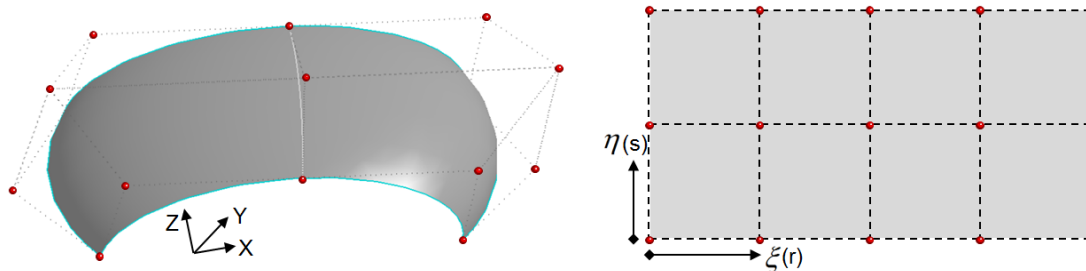


Fig.4: NURBS surface with the control points (red dots) in physical space (left) and in parametric space (right)

It can be seen, that in the parametric space a more or less regular grid of control points in a rectangular parametric space is defined. Mapping this into the physical space leads to some restrictions of defining any specific boundary of the surface or even digging holes into it. To circumvent this limitation, so-called trimmed NURBS are generally used in the CAD programs. Trimmed NURBS surfaces are defined by adding an unlimited number of so-called trimming curves, which actually define which part of the represented surface

shall be an actual part of the given geometry. Fig. 5 shows a trimmed NURBS surface that uses the exact same underlying representation of the NURBS surface shown in Fig. 4, just by adding two additional trimming curve definitions that specify the outer boundary of the actual geometry and a hole inside it.

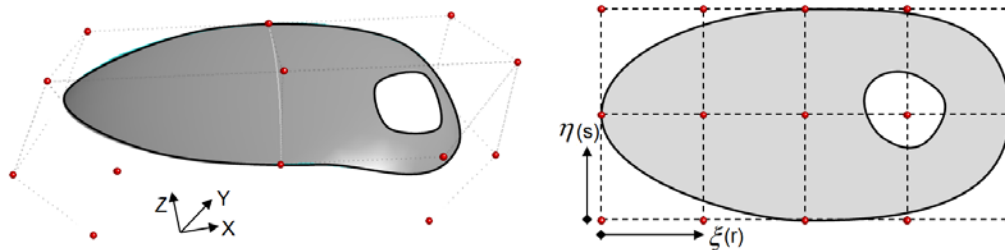


Fig.5: Trimmed NURBS surface with control points in physical space (left) and in parametric space (right)

Trimmed NURBS surfaces are a widely used standard in CAD programs, so the capability of doing isogeometric finite element analysis on trimmed NURBS surfaces is one of the key requirements. The NURBS shell implementation in LS-DYNA supports the analysis on trimmed surfaces.

3 NURBS shells in LS-DYNA

This section summarizes quickly the current possibilities of using NURBS surface representations for doing isogeometric finite element analysis in LS-DYNA.

Starting off a FEA using NURBS shells necessitates the appropriate definition of NURBS surfaces, which are called NURBS patches in LS-DYNA. The keyword to be used is ***ELEMENT_SHELL_NURBS_PATCH** or ***ELEMENT_SHELL_NURBS_PATCH_TRIMMED** in case of trimmed NURBS surfaces. Generally any material model available for standard shell elements may be used in combination with the NURBS shells. Currently there are five different NURBS shell formulations available, that differ basically in the way the normal of the shell is approximated. There are shell formulations based on the classical Kirchoff-Love shell theory as well formulations based on the shear-deformable Reissner-Mindlin theory. As basis functions with higher continuity across the element boundaries allow for element formulations that do not need any rotational degrees of freedom, also rotation free NURBS shells are available in LS-DYNA. More information on the implemented shell formulations can be found in the papers from Benson et al. [4], [5]. Likewise it is done for standard shell elements, each NURBS patch will be assigned to an appropriate ***PART**, that defines the material model to be used (***MAT_XXX**) and the section properties of the shell (***SECTION_SHELL**). For doing an analysis with NURBS shells, the parameter **ELFORM** in ***SECTION_SHELL** has to be set to **201**.

In the previous section about NURBS it became obvious, that the control points are not necessarily a part of the actual geometry. This fact makes it a little bit more complicated to actually apply necessary boundary conditions at the spot they should be. To do so, the keyword ***CONSTRAINED_NODE_TO_NURBS_PATCH** is available, which allows to define a massless node at any location on the actual NURBS surface and tie it to the NURBS patch. Having defined this particular location on the NURBS surface one may apply either Neumann or Dirichlet boundary conditions at this spot.

For dealing with contact boundary conditions basically two options are available. The first one is based on so-called *interpolation elements*. When doing an isogeometric analysis, LS-DYNA automatically creates a kind of background mesh consisting of standard bi-linear shell elements that are placed on the NURBS surface. The necessary *interpolation nodes* that are created for that are fully constrained to the underlying NURBS patch

description. Having this interpolation mesh in place, any standard penalty based contact formulation available in LS-DYNA can be used directly. The second possibility actually uses the real smooth description of the NURBS surface in the sense of a Node-To-Surface contact description. For this, the interpolation nodes on the slave side are projected onto the master surface which is described by the NURBS basis functions. This second approach can be activated by setting **IGACTC=1** in ***CONTROL_CONTACT**.

From an analysis perspective, the NURBS-based finite shell elements are available for explicit as well as for implicit analysis. They are supported in SMP (shared memory parallel) and MPP (massive parallel processing). Furthermore a conventional type of mass scaling [6] has been implemented for the NURBS shell elements as well as the possibility to treat them as rigid bodies.

4 Sheet metal forming with NURBS shells

Making the finite element analysis with NURBS shells in LS-DYNA a sound alternative in real world industrial-applications, numerous possibilities and features that have been developed over the last decades for standard finite elements must be made available for NURBS shells as well. In this section the focus is set on the recent advances of the NURBS shell implementation for the use in sheet metal forming application.

A more or less classical forming simulation consists of various steps that are generally analyzed within a multi-stage analysis. These stages may be deep-drawing, trimming, hemming, springback and others. No matter what the individual stage may look like, one major feature that needs to be supported by an analysis tool is to map the results achieved from one stage to the other. The results that are mainly mapped are the current stress and strain states, the equivalent plastic strain and the thickness variation due to the forming process. In LS-DYNA this is typically done via a so-called DYNAIN-file using the keyword ***INTERFACE_SPRINGBACK**. So once the analysis of one stage is finished, the required data is written out to this particular file and then in the next stage, this file is read back in in order to reinitialize the necessary values. To support this approach with NURBS shells two new keywords have been added to LS-DYNA, namely ***INITIAL_STRESS/STRAIN_SHELL_NURBS_PATCH** (see Fig.6).

***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					

Fig.6: Keyword to support data mapping between various stages

When defining the keyword ***INTERFACE_SPRINGBACK** for a particular analysis stage, LS-DYNA will automatically create a DYNAIN-file and write out for every NURBS-element (EID) the required solution values (stresses – SIGX, equivalent plastic strain – EPS and history variables - HISVx) at every integration point (NPLANE*NTHICK) using the keyword shown in Fig.6. The identical procedure can be made to map the strain values and the thickness variation.

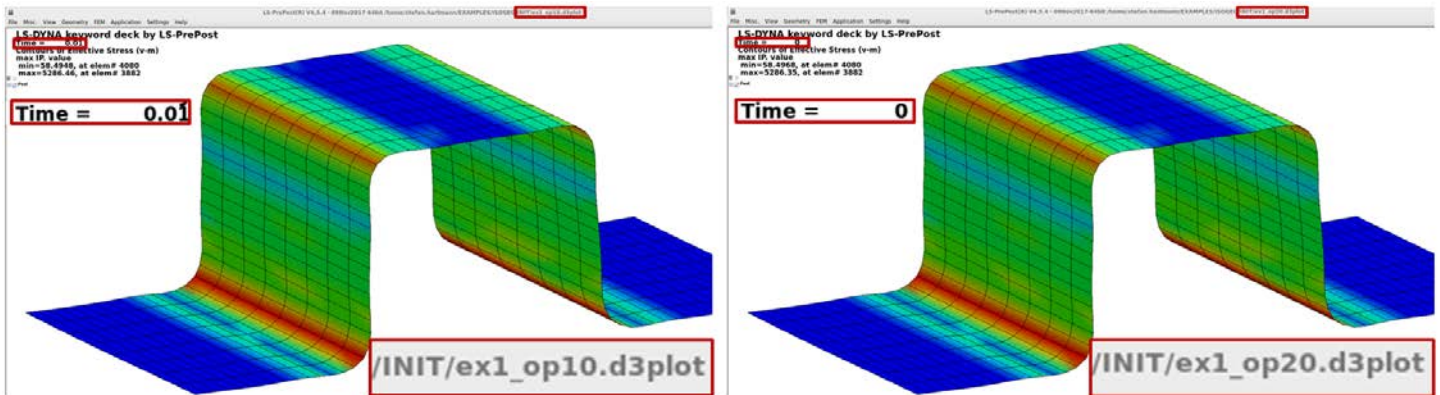


Fig.7: Von Mises Stress distribution: end of stage 1 (op10 - left) and beginning of stage 2 (op20 –right)

In Fig.7 you can see a little example that shows the von Mises Stress distribution at the end of the first stage and at the beginning of the second stage after reading back in the results file. Another important issue in forming applications is the trimming of the formed component. The support of this feature is currently under development for IGA shells and will be available soon (see Fig.8).

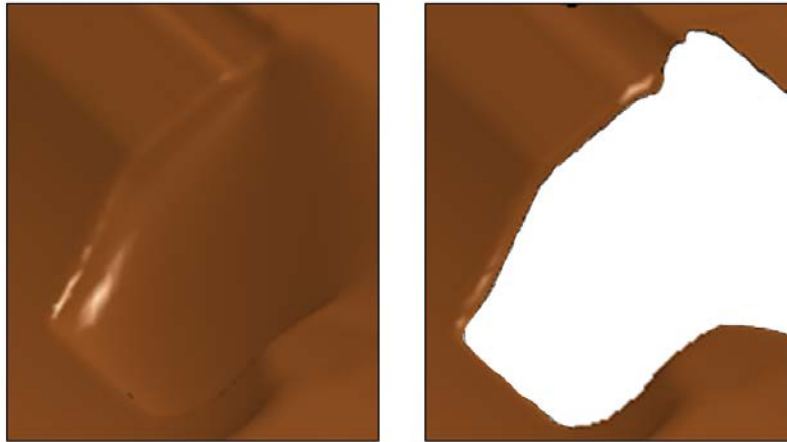


Fig.8: A deformed shell NURBS patch before (left) and after trimming (right)

5 Multistage forming process

Unfortunately the analysis of a multistage forming process to present the current capabilities and to compare it with respect to the current state-of-the art methods is still work in progress. These studies will be carried out until the date of the actual conference and presented there.

6 Conclusion

LSTC is working on adding more and more features that can be used together with Isogeometric NURBS shells in LS-DYNA. The current paper mentioned some recent advances that have been implemented in the context of forming analyses. Although the very important mapping ability via ***INTERFACE_SPRINGBACK** is now supported there is still some work to do in order to make the IGA technology available for general use in

industrial applications. One standard feature that is generally used in forming applications is the possibility to do adaptive mesh refinement in all the areas where this is needed. This fundamental topic hasn't been addressed yet for NURBS shells and will be another important step to be done.

References

- [1] Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: "Isogeometric Analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement", Computer Methods in Applied Mechanics and Engineering, 194: 4135-4195, 2005
- [2] Piegel, L., Tiller, W.: "The NURBS Book", Springer-Verlag (Berlin Heidelberg), 1995
- [3] Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.: "Isogeometric Analysis – Toward Integration of CAD and FEA", John Wiley & Sons, Ltd, 2009
- [4] Benson, D.J., Bazilevs, Y., Hsu, M.C., Hughes, T.J.R.: „Isogeometric shell analysis: The Reissner-Mindlin shell“, Computer Methods in Applied Mechanics and Engineering, 199, 276-289, 2010
- [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, 1367-1378, 2011
- [6] Hartmann, S., Benson, D.J.: “Mass scaling and stable time step estimates for isogeometric analysis”, International Journal for Numerical Methods in Engineering, 102, 2014, 671-687