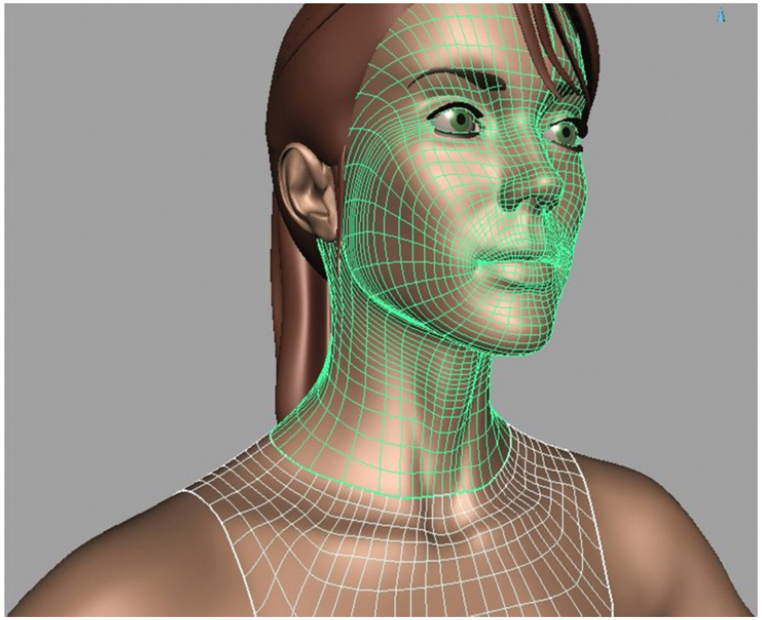
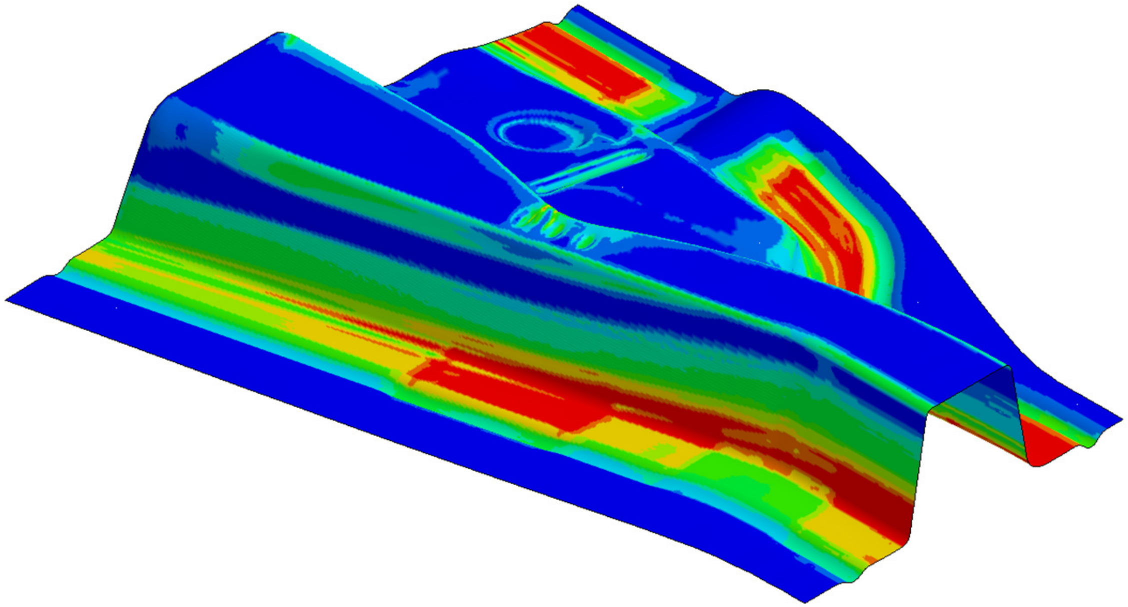
Recent developments in LS-DYNA for Isogeometric Analysis



T.J.R. Hughes



Stefan Hartmann

DYNAmore GmbH, Industriestr. 2, D-70565 Stuttgart, Germany

Some slides borrowed from:

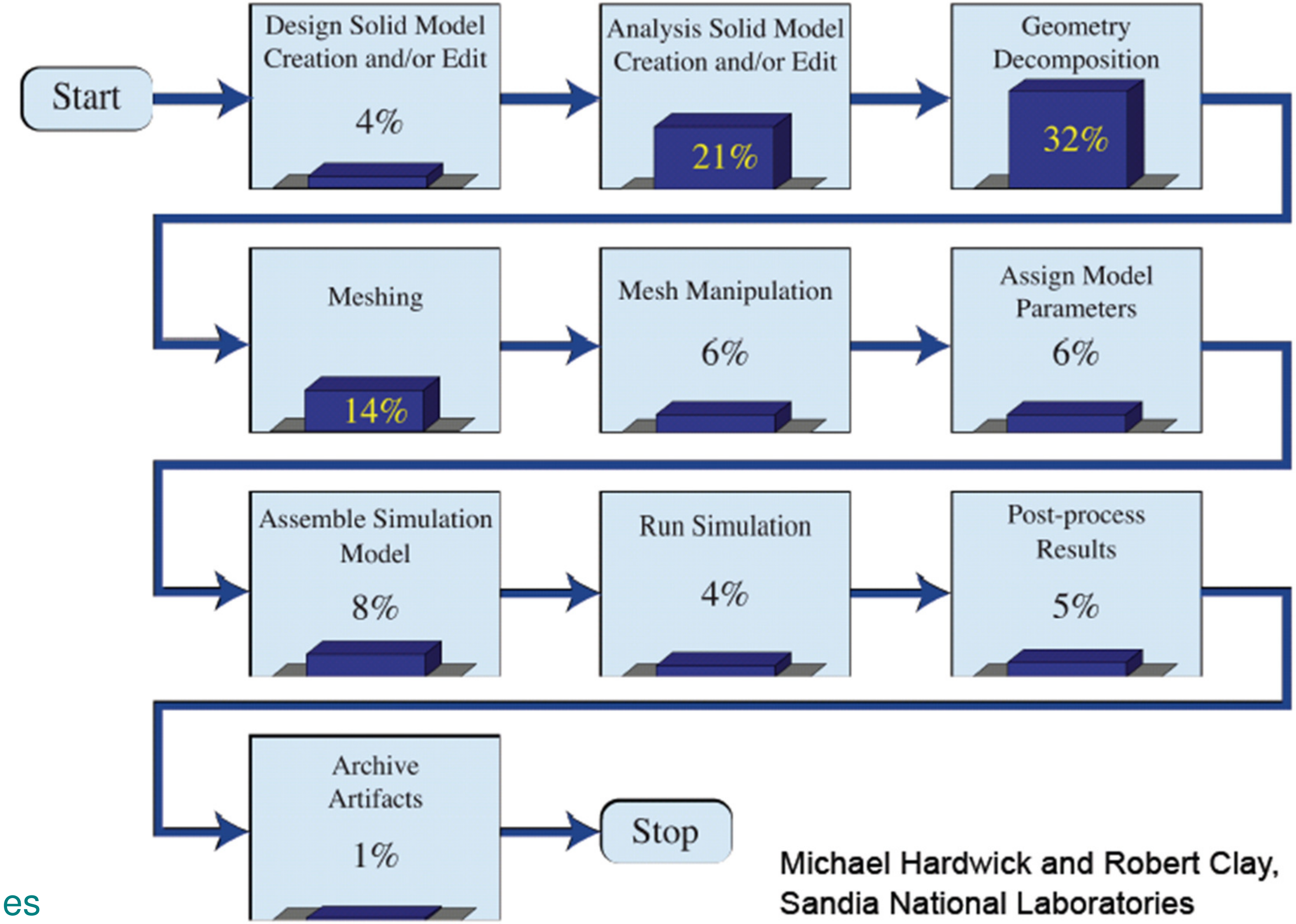
T.J.R. Hughes: Professor of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin
D.J. Benson: Professor of Applied Mechanics, University of California, San Diego

Outline

- Isogeometric Analysis
  - motivation / definition / history
- From B-splines to NURBS (T.J.R. Hughes)
  - basis functions / control net / refinements
- *generalized elements* in LS-DYNA (D.J. Benson)
  - basic idea / shell formulations / interpolation-nodes / interpolation-elements
- NURBS-based finite elements in LS-DYNA
  - *ELEMENT_NURBS_PATCH_2D
- Example: Underbody Cross Member (Numisheet 2005)
  - description / comparison of results / summary
- Summary and Outlook

Isogeometric Analysis – motivation (… at the beginning)

- reduce effort of geometry conversion from CAD into a suitable mesh for FEA



Michael Hardwick and Robert Clay,
Sandia National Laboratories

T.J.R. Hughes

Isogeometric Analysis - definition

- ISOPARAMETRIC (FE-Analysis)
  use same approximation for geometry and deformation
  (normally: low order Lagrange polynomials ---- in LS-DYNA basically only linear elements)

GEOMETRY	←→	DEFORMATION

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

CAD	←→	FEA

- common geometry descriptions in CAD
  - NURBS (Non-Uniform Rational B-splines) → most commonly used
  - T-splines → enhancement of NURBS
  - subdivision surfaces → mainly used in animation industry
  - and others

Isogeometric Analysis - history

- start in 2003
  - summer: Austin Cotrell starts as PhD Student of Prof. T.J.R. Hughes at the University of Texas, Austin
  - autumn: first NURBS-based FE-code for linear, static problems provides promising results, the name „ISOGEOMETRIC“ is used the first time

- 2004 up to now: many research activities to various topics
  - non-linear structural mechanics
    → shells with and without rotational DOFs
    → implicit gradient enhanced damage
    → XFEM
  - shape- und topology-optimization
  - efficient numerical integration
  - turbulence and fluid-structure-interaction
  - acoustics
  - refinement strategies
  - …

- January 2011: first thematic conference on *Isogeometric Analysis*
  - “Isogeometric Analysis 2011: Integrating Design and Analysis“, University of Texas at Austin

From B-splines to NURBS

- B-spline basis functions
  - constructed recursively
  - positive everywhere (in contrast to Lagrange polynomials)
  - shape of basis funktions depend on: knot-vector and polynomial degree
  - knot-vector: non-decreasing set of coordinates in parameter space
  - normally $C^{(P-1)}$-continuity

→ e.g. lin. / quad. / cub. / quart. Lagrange: → C^0 / C^0 / C^0 / C^0

→ e.g. lin. / quad. / cub. / quart. B-spline: → C^0 / C^1 / C^2 / C^3

Example of a uniform knot-vector: $\Xi = \{0,1,2,3,4,\ldots\}$

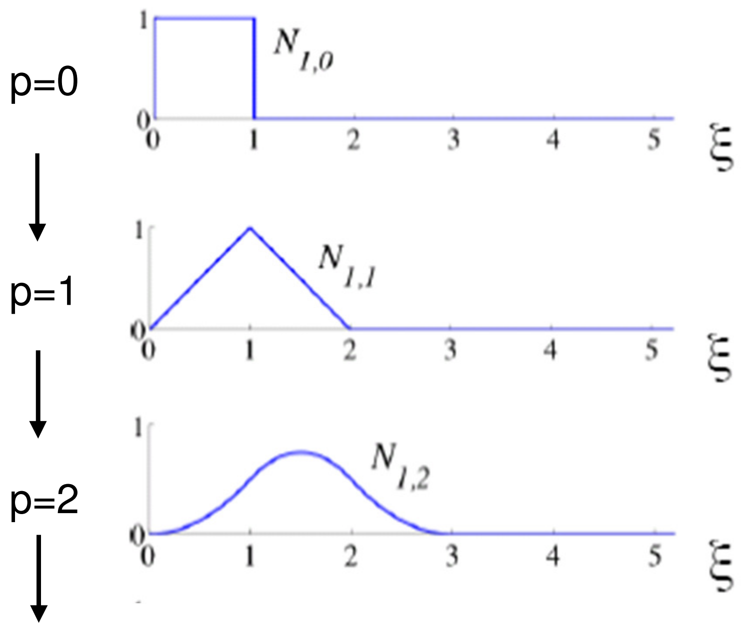
$p = 0:$

$$N_{i,0}(\xi) = \begin{cases} 1 & if\ \ \xi_i \le \xi < \xi_{i+1} \\ 0 & otherwise \end{cases}$$

$p > 0:$

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

T.J.R. Hughes

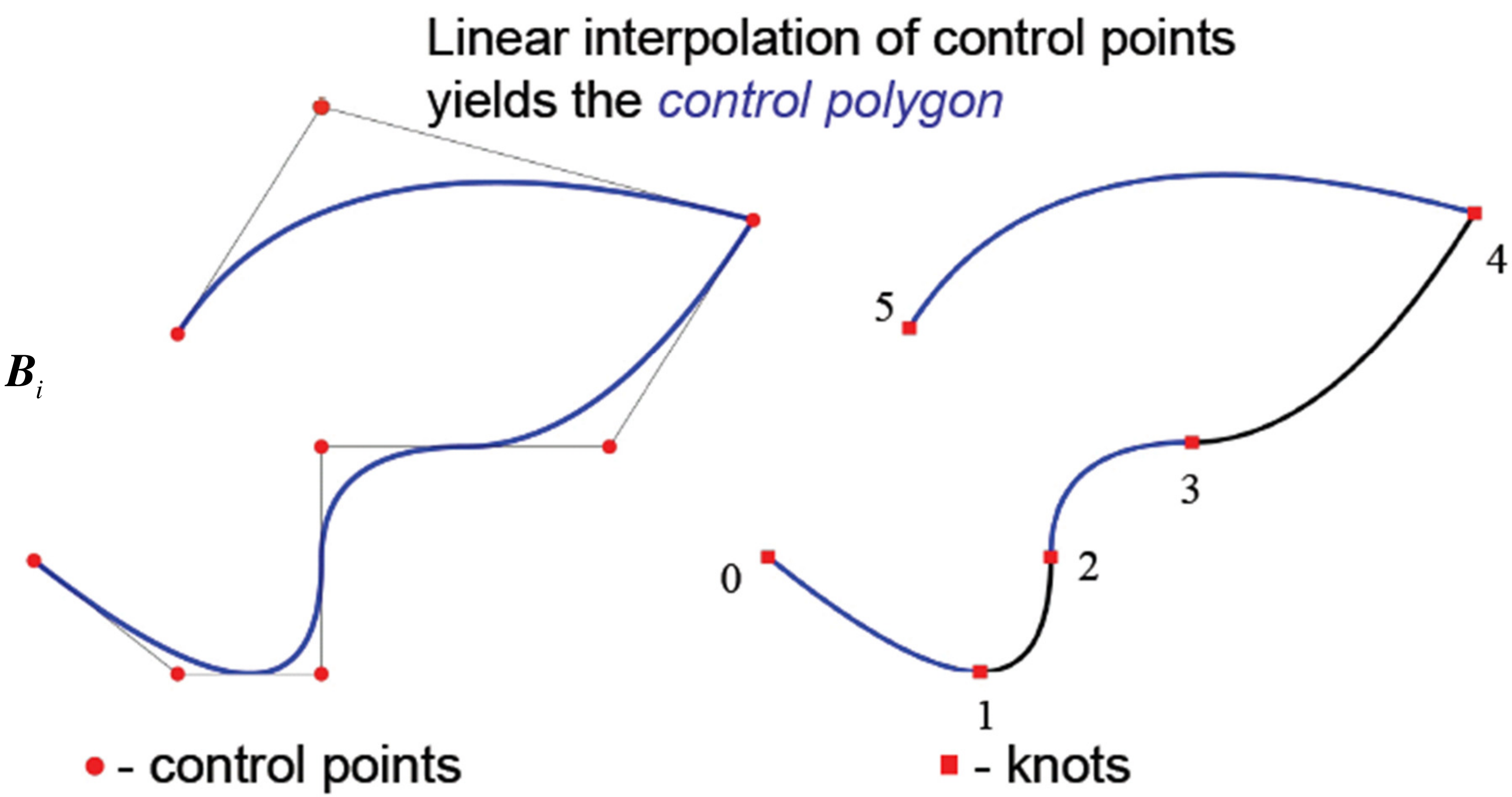


From B-splines to NURBS

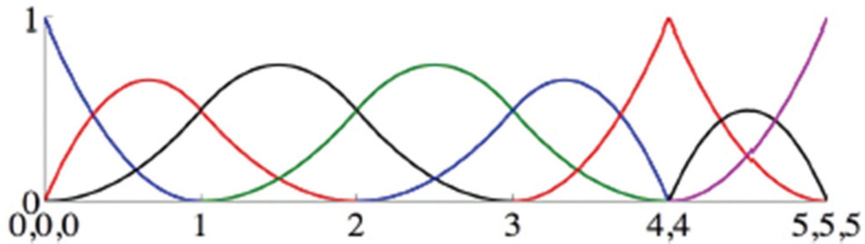
- B-spline curves
  - control points $\boldsymbol{B}_i$ / control polygon (control net)
  - knots

linear combination:

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



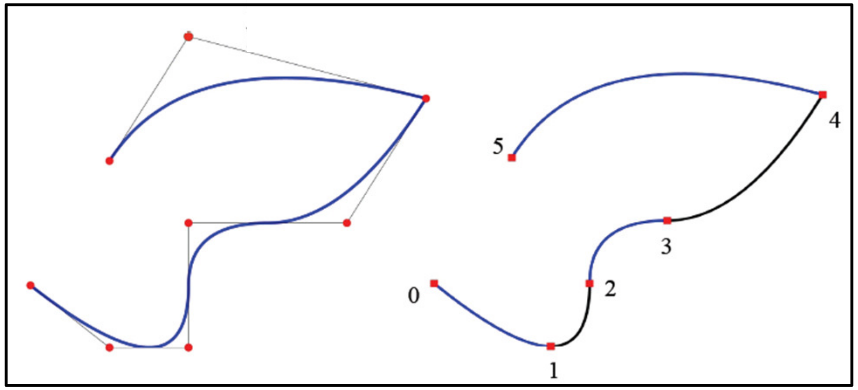
Quadratic basis



T.J.R. Hughes

From B-splines to NURBS

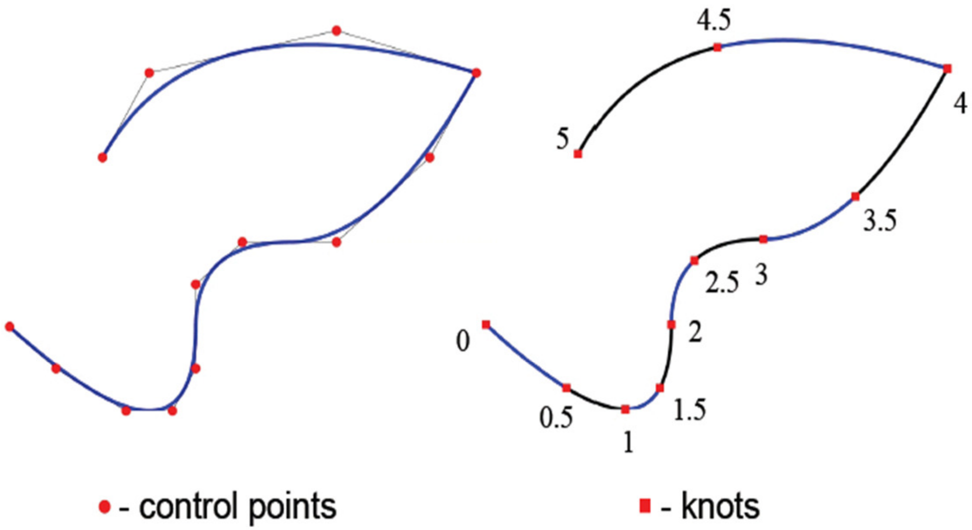
- B-splines
  - refinement posibilities



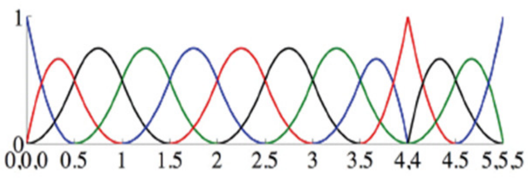
h-refinement

→ number of elements increases

→ order of polynomial remains the same



Quadratic basis

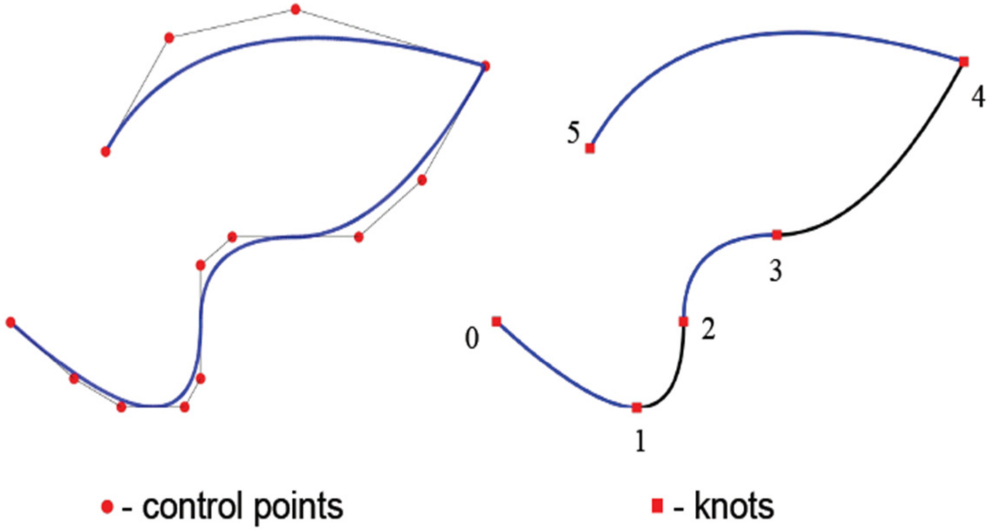


T.J.R. Hughes

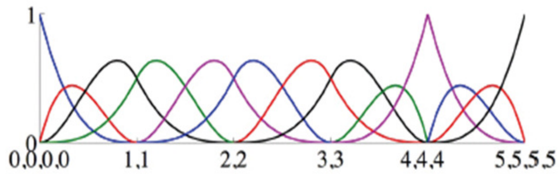
p-refinement

→ number of elements remains the same

→ order of polynomial increases



Cubic basis



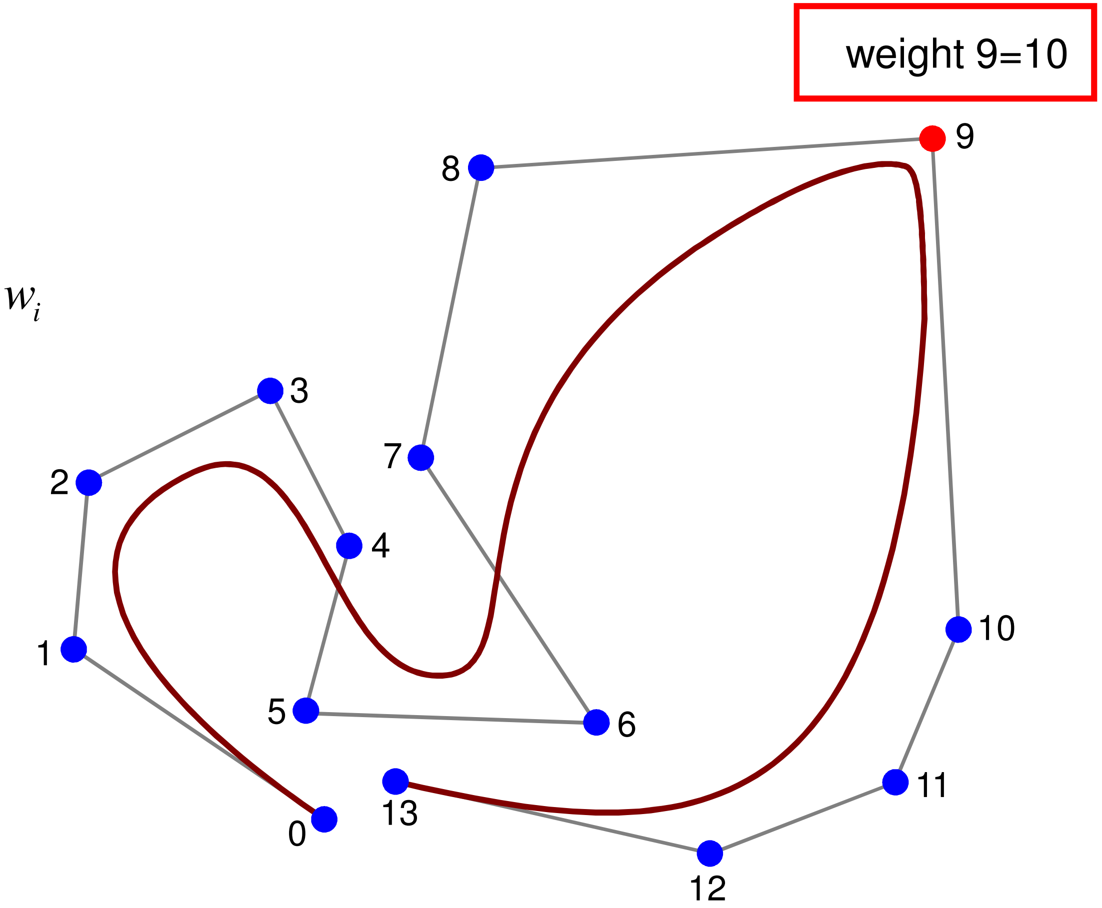
From B-splines to NURBS

- NURBS – Non-Uniform Rational B-splines
  - weights at control points leads to more control over the shape of a curve
  - projective transformation of a B-spline

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)\, w_i}{W(\xi)}$$

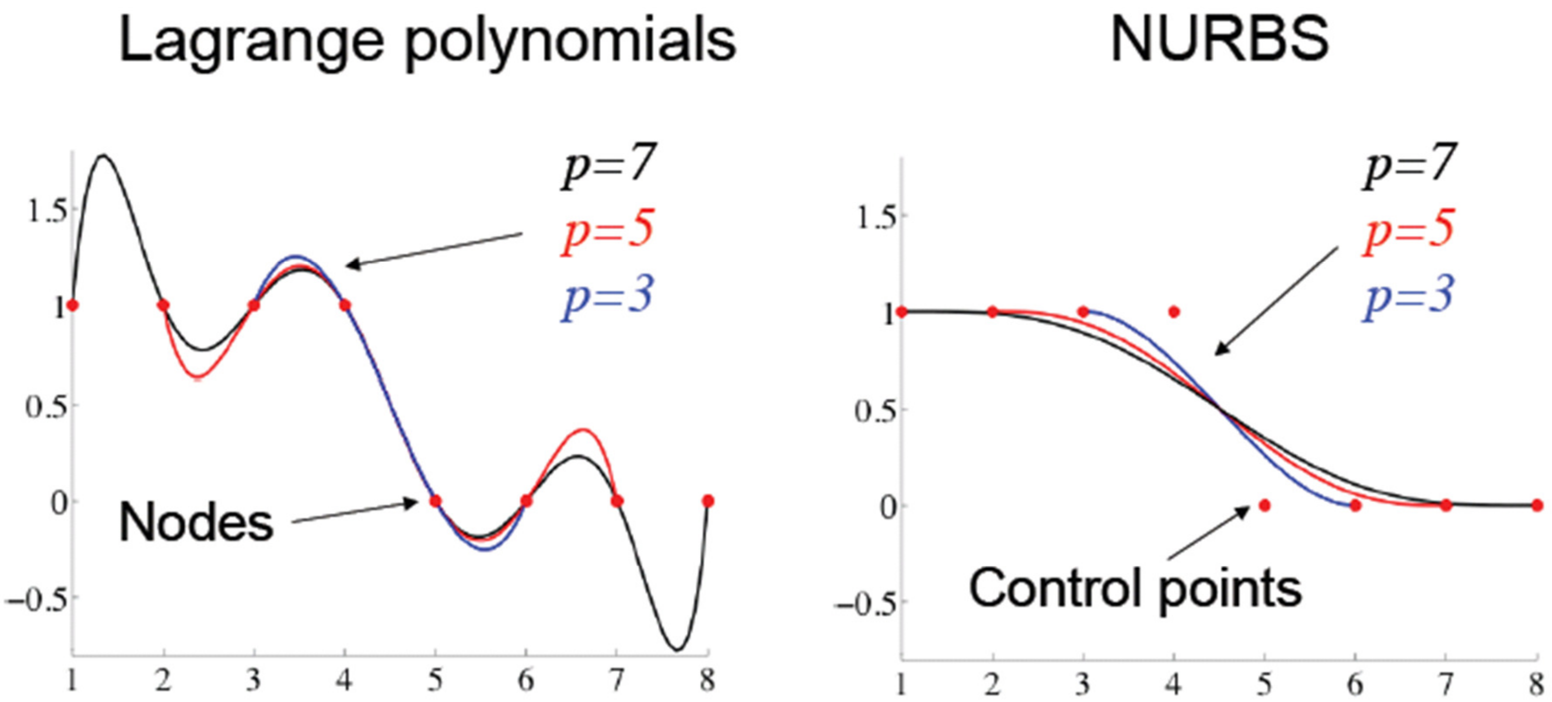
$$with:\ W(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\, w_i$$

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



From B-splines to NURBS

- smoothness of Lagrange polynomials vs. NURBS



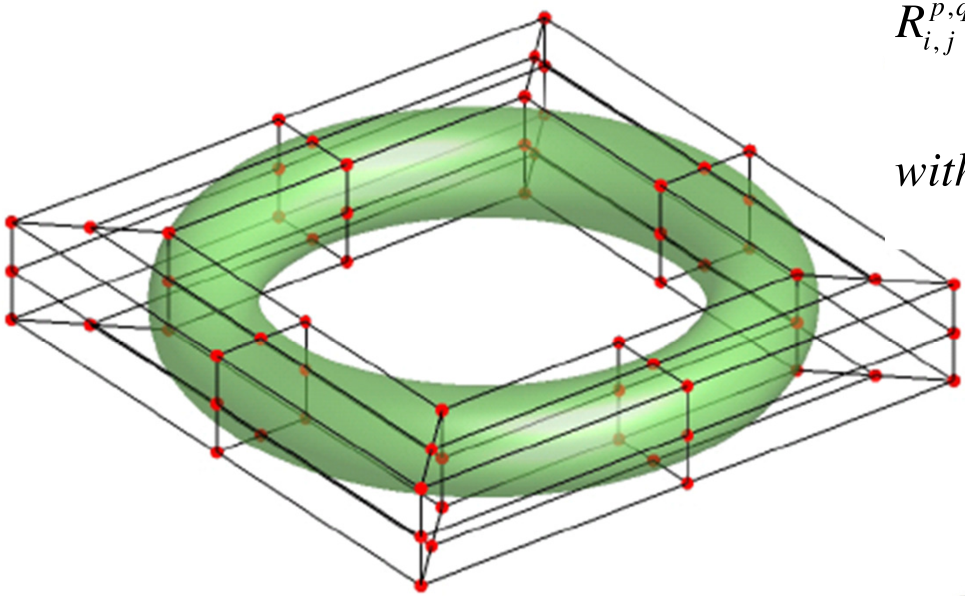
T.J.R. Hughes

From B-splines to NURBS

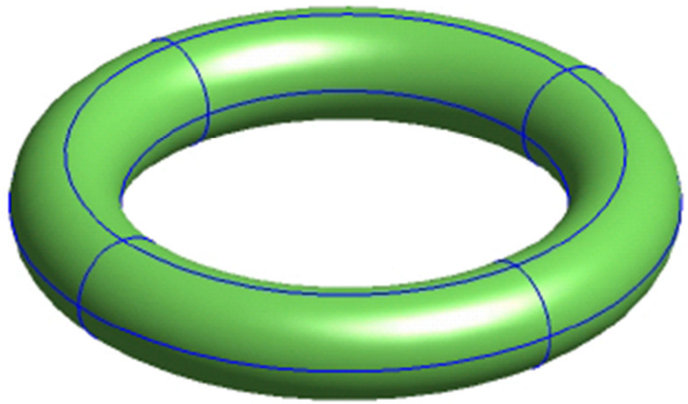
- NURBS – surfaces (tensor-product of univariate basis)

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{W(\xi,\eta)}$$

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



Control net



Mesh

T.J.R. Hughes

From B-splines to NURBS - summary

- B-spline basis functions
  - recursive
  - dependent on knot-vector an polynomial order
  - normally $C^{(P-1)}$-continuity
  - „partition of unity“ (like Lagrange polynomials)
  - refinement (h/p and k) without changing the initial geometry → adaptivity
  - control points are normally not a part of the physical geometry (non-interpolatory basis functions)

- NURBS
  - B-spline basis functions + control net with weights
  - all mentioned properties for B-splines apply for NURBS

generalized elements in LS-DYNA

- basis functions
  - active research as well in the field of CAD as in computer animation
- implementation of finite elements for specific basis functions
  - time consuming
  - software might become obsolete once new types of basis functions appear
- wish: possibility for fast prototyping of new elements

⇩

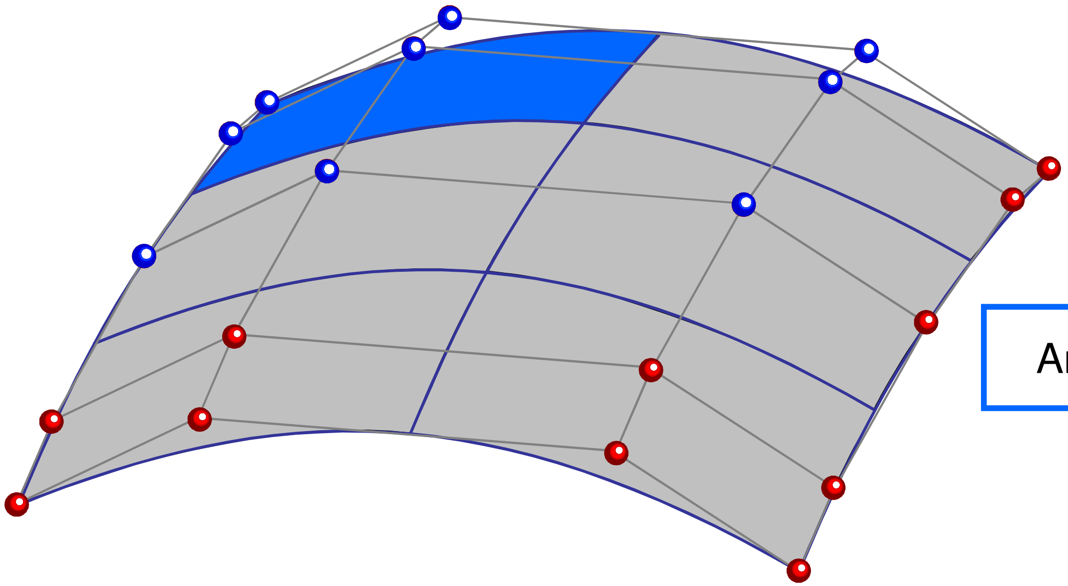
- ***generalized elements*** (***shells*** and *solids*)
  - elements are formulated with the help of *generalized coordinates*
  - implementation is independent of utilized basis functions
  - new basis functions can be used immediately → rapid prototyping of elements
  - properties of *generalized elements* are defined exclusively in the input deck:
    → values of shape functions and its derivatives at generalized coordinates and integration points
    → values of integration weights
  - generalized formulation allows the use of different types of basis functions
    → Lagrange polynomials (standard FEA) / NURBS / T-splines / subdivision surfaces / …

D.J. Benson

generalized elements in LS-DYNA - visualization

- generalized coordinates (coorespond to control points in NURBS)
  - are normally not part of the physical geometry
- LS-PrePost
  - displays only elements with linear basis functions
    → right now it is able to display and modify NURBS
- *interpolation elements*
  - linear elements to visualize results of generalized elements
  - are used for contact treatment
- *interpolation nodes*
  - nodes on physical geometry to define interpolation elements
  - displacements of interpolation nodes follow a linear function depending on displacements of generalized coordinates

generalized elements in LS-DYNA



*DEFINE_ELEMENT_GENERALIZED_SHELL
*ELEMENT_GENERALIZED_SHELL

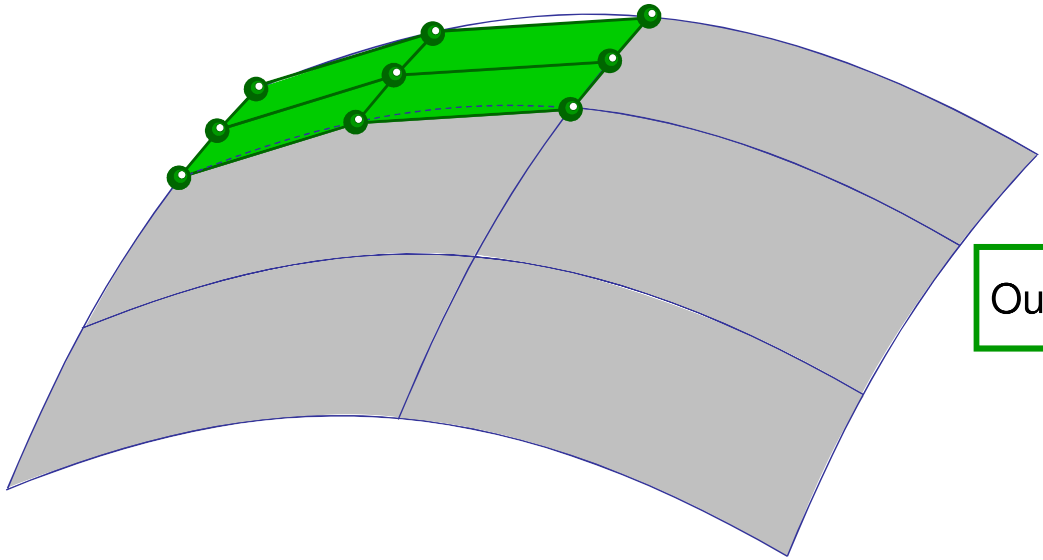
Analysis

- *DEFINE_ELEMENT_GENERALIZED_SHELL
  - element ID / number of IPs / number of generalized coordinates
  - for each IP: weights & values of all basis functions and derivatives at IPs
  - for each generalized coordinate: values of all basis functions and derivatives at control points

→ e.g.: a typical 9-noded element with 9 IPs necessitates approx. **172 lines** in input deck!

- *ELEMENT_GENERALIZED_SHELL
  - connectivity of an element (here: blue control points)

generalized elements in LS-DYNA



Output & BC

*CONSTRAINED_NODE_INTERPOLATION
*ELEMENT_INTERPOLATION_SHELL

- *CONSTRAINED_NODE_INTERPOLATION
  - for each interpolation node: number of control points of which the position of this interpolation node is dependent
  - IDs of control points and weighting factors
    → displacement of interpolation node will be interpolated linearly depending on the displacements of control points
  - will be used for contact treatment at the moment
- *ELEMENT_INTERPOLATION_SHELL
  - dependency to control points with appropriate weighting factors

generalized elements in LS-DYNA – shell formulations

- shear deformable & thin shell theory
  - with rotational DOFs
  - without rotational DOFs

	shear deformable shell theory	thin shell theory
displacement field: $x(\eta,\xi,\zeta)$	$\sum_i N_i(\eta,\xi)\left(x_i + \frac{h}{2}\zeta\,\hat{y}_i\right)$	$\sum_i N_i(\eta,\xi)x_i + \frac{h}{2}\zeta\,\hat{n}(\eta,\xi)$
velocity field: $\dot{x}(\eta,\xi,\zeta)$	$\sum_i N_i(\eta,\xi)\left(\dot{x}_i + \frac{h}{2}\zeta\,\dot{\hat{y}}_i\right)$	$\sum_i N_i(\eta,\xi)\dot{x}_i + \frac{h}{2}\zeta\,\dot{\hat{n}}(\eta,\xi)$
unit orientation vector (shell normal):	$\hat{y}$	$\hat{n}$
time derivative of unit orientation vector:	with /without rotational DOFs $\dot{\hat{y}}_i = \omega_i \times \hat{y}_i$	$\dot{\hat{y}}_i = \sum_j \frac{\partial \hat{y}_i}{\partial x_j}\dot{x}_j$	with / without rotational DOFs

generalized elements in LS-DYNA

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

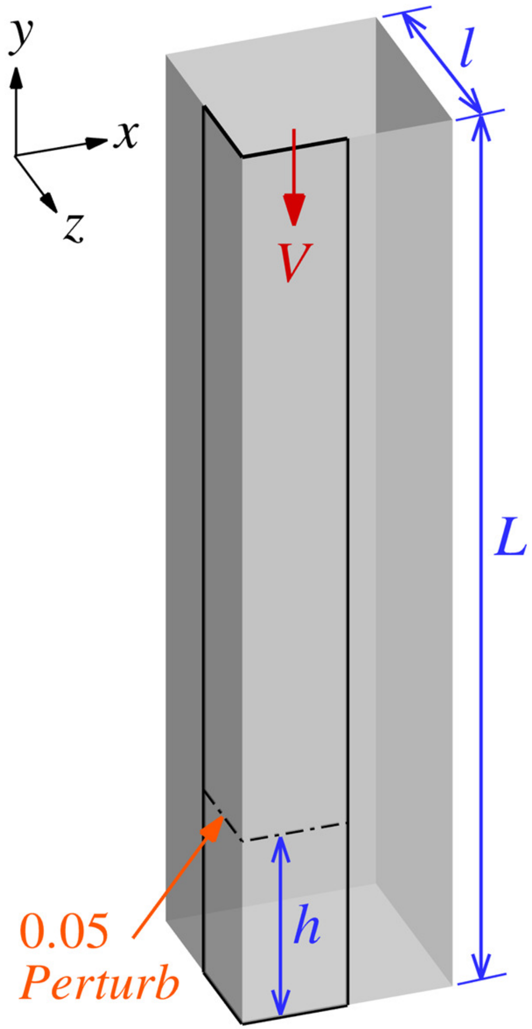
- some boundary conditions are implemented via *interpolation elements*
  - contact treatment
  - pressure distribution (not fully tested yet)

- time step control via *„maximum system eigenvalue“*
  - D.J. Benson: *Stable Time Step Estimation for Multi-material Eulerian Hydrocodes*, CMAME,191-205 (1998)

- generalized ***solids*** are implemented as well
  - „standard“ displacement elements

generalized elements in LS-DYNA - Example

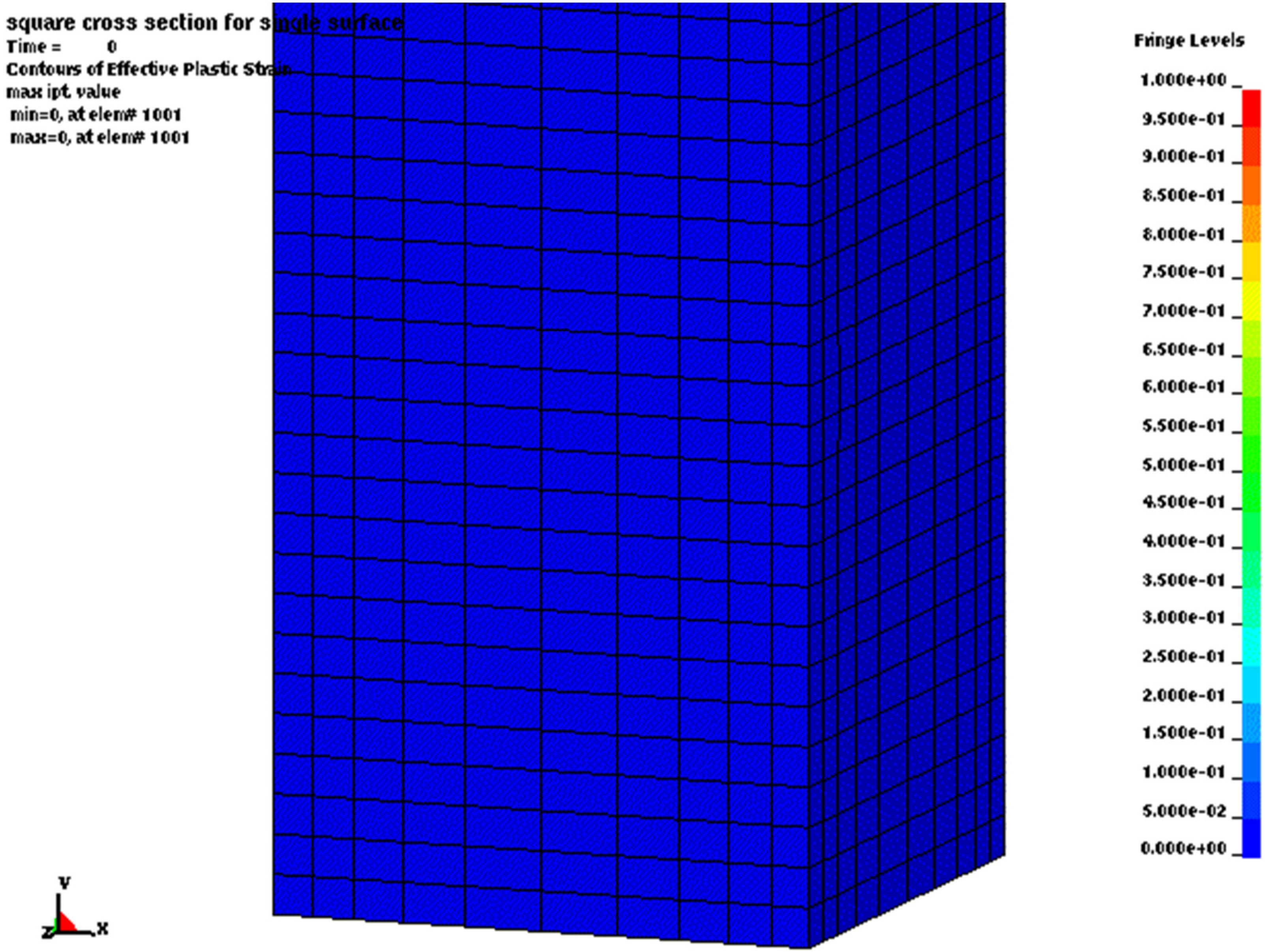
- buckling of a square tube (using NURBS basis functions)



D.J. Benson

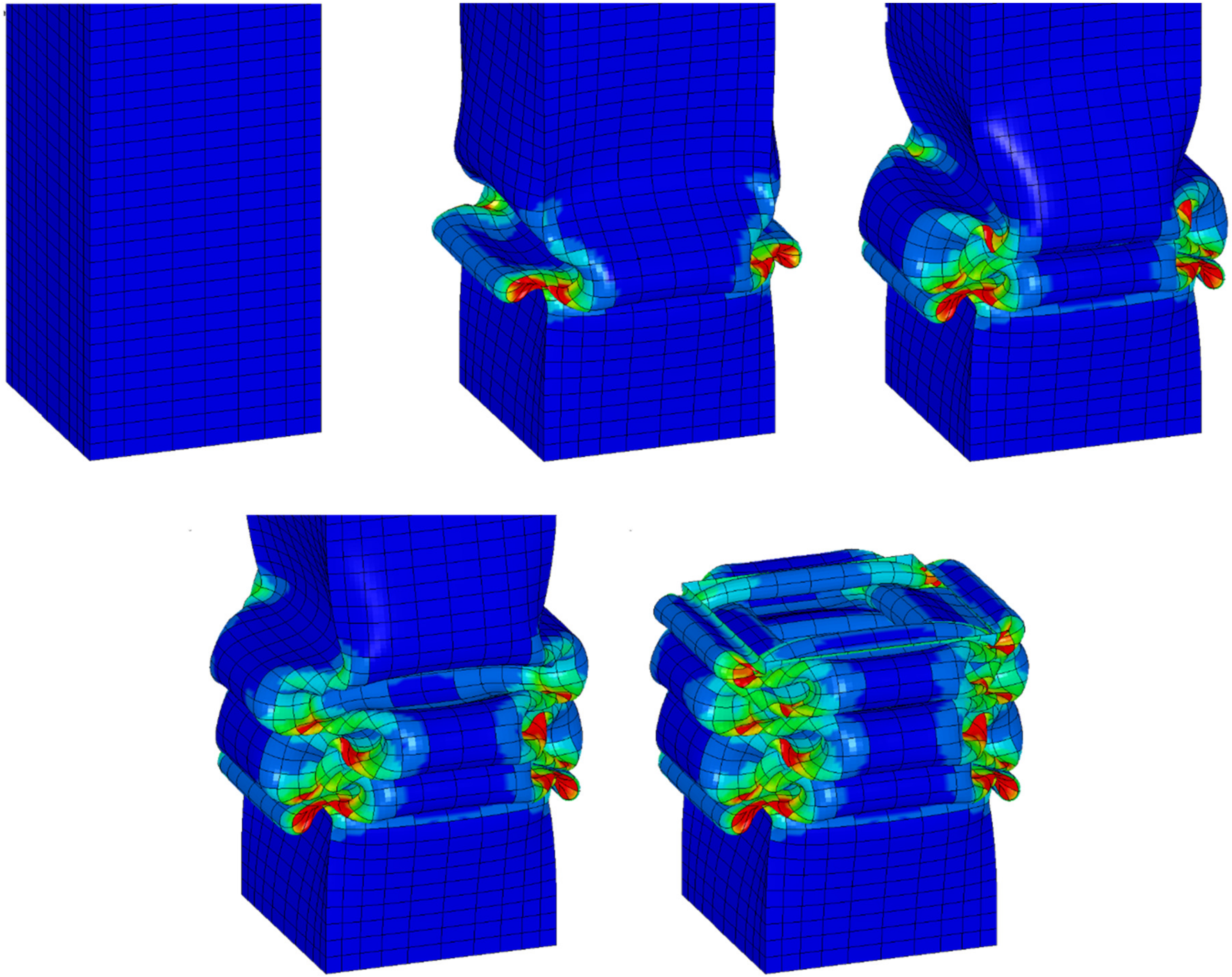
- standard benchmark for automobile crashworthiness
- quarter symmetry to reduce cost
- perturbation to initiate buckling mode
- J_2 plasticity with linear isotropic hardening
- mesh:
  - 640 quartic (P=4) elements.
  - 1156 control points.
  - 3 integration points through thickness.

- buckling of a square tube (NURBS-elements: p=4)



D.J. Benson

- buckling of a square tube (NURBS-elements: p=4)

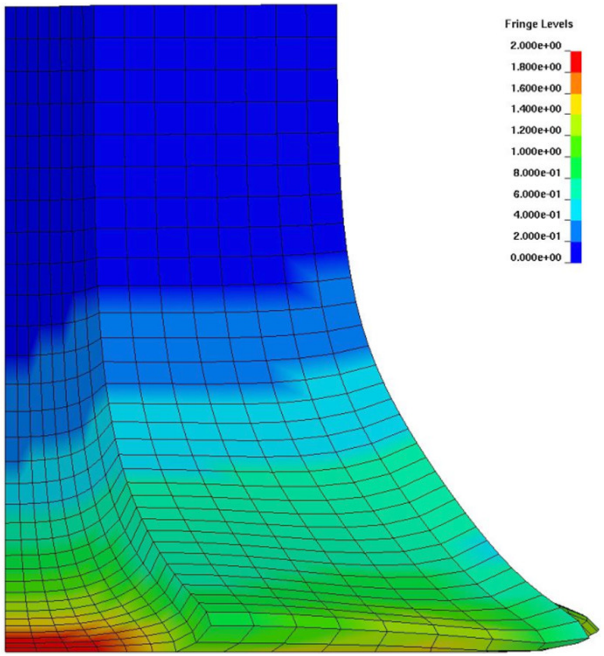


D.J. Benson

generalized elements in LS-DYNA – solid elements

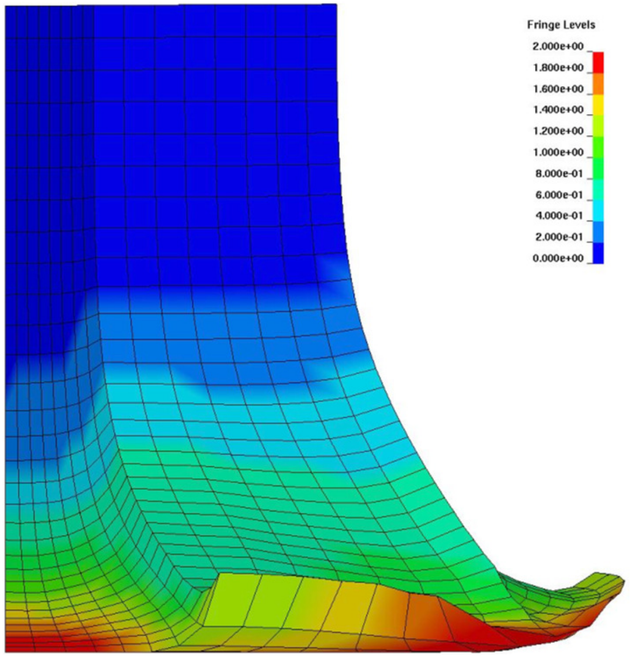
- Taylor Bar Impact

1-Pnt Hex



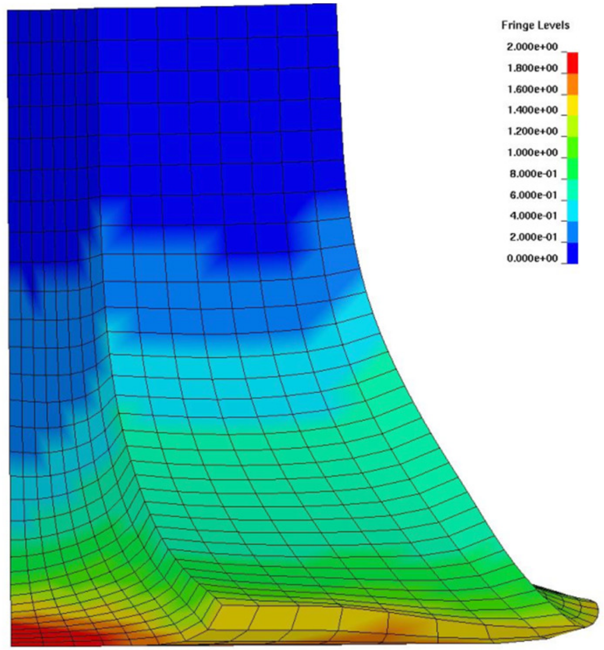
Standard LS-DYNA element

27 Node Quadratic



Generalized Element

Quadratic NURBS



Generalized Element

Formulation	# Nodes/ CP	Peak Plastic Strain	# Time Steps
1-Pnt Hex	2677	2.164	**2136**
Quad. Lagr.	2677	2.346	**3370**
Quad. NURBS	648	2.479	**954**

D.J. Benson

generalized elements in LS-DYNA - summary

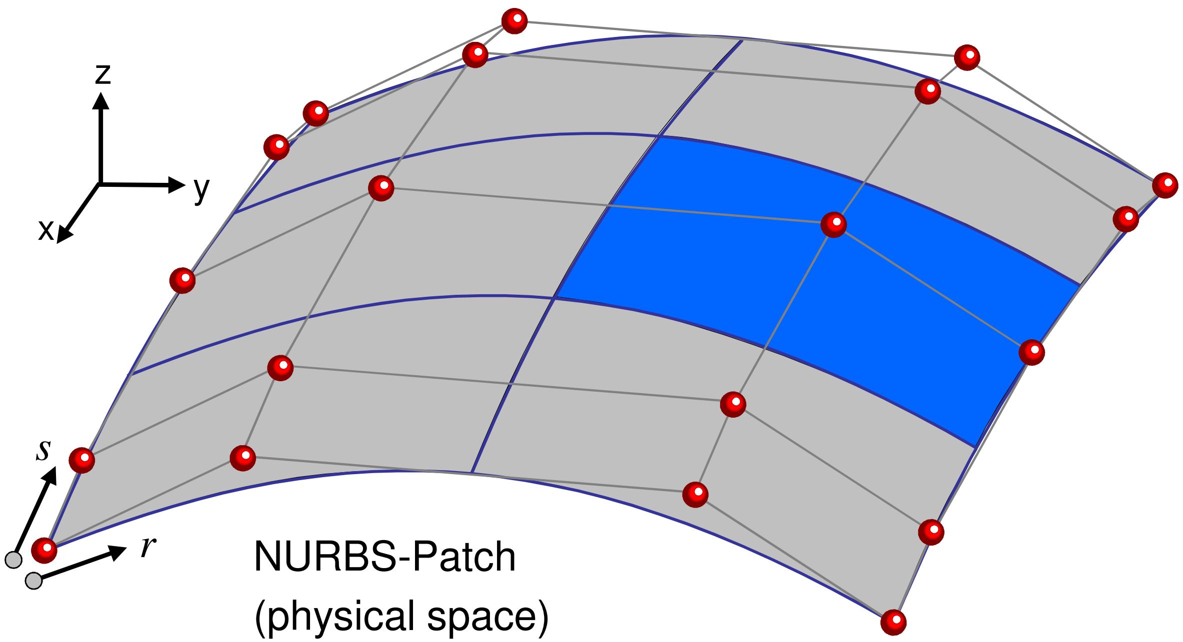
- fast prototyping of new elements
  - all the information in the input deck
  - no limitation on number of control points and integration points per element
  - no restriction to special types of basis functions
  - interesting for research
  - rather difficult to create input deck → not usable for industry
  - good results with NURBS → decision to implement NURBS-based finite elements in LS-DYNA

- generalized shells
  - shear deformable and thin shell theories implemented
  - with and without rotational DOFs

- generalized solids
  - „standard" displacement formulation

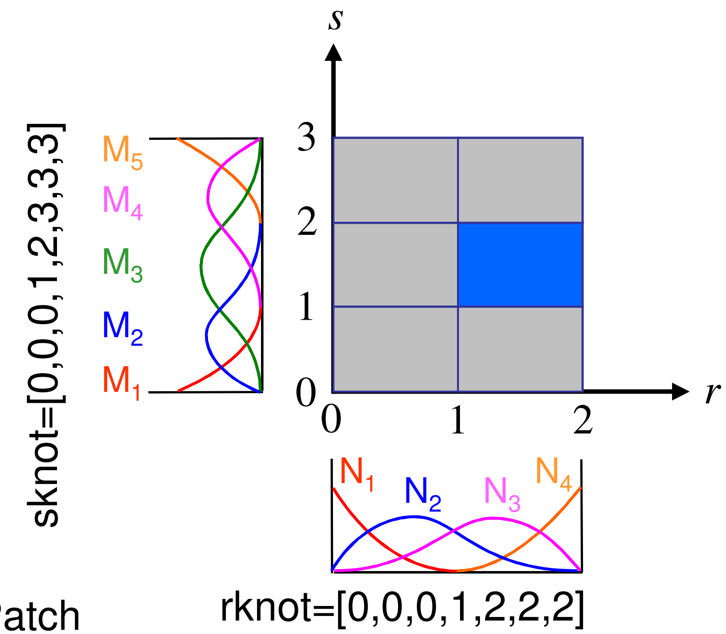
NURBS-based finite elements in LS-DYNA

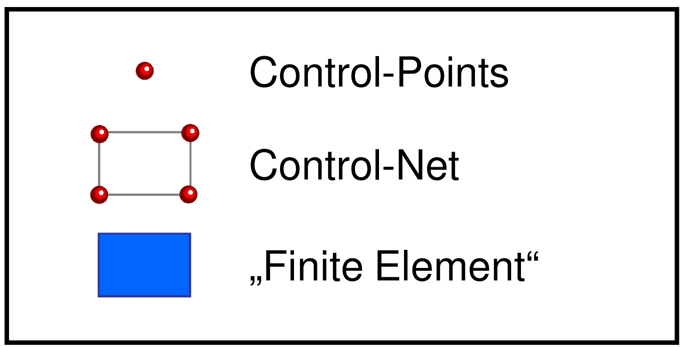
- A typical NURBS-Patch and the definition of elements
  - elements are defined through the knot-vectors (interval between different values)
  - shape functions for each control-point



NURBS-Patch
(physical space)

polynomial order:
- quadratic in r-direction (pr=2)
- quadratic in s-direction (ps=2)

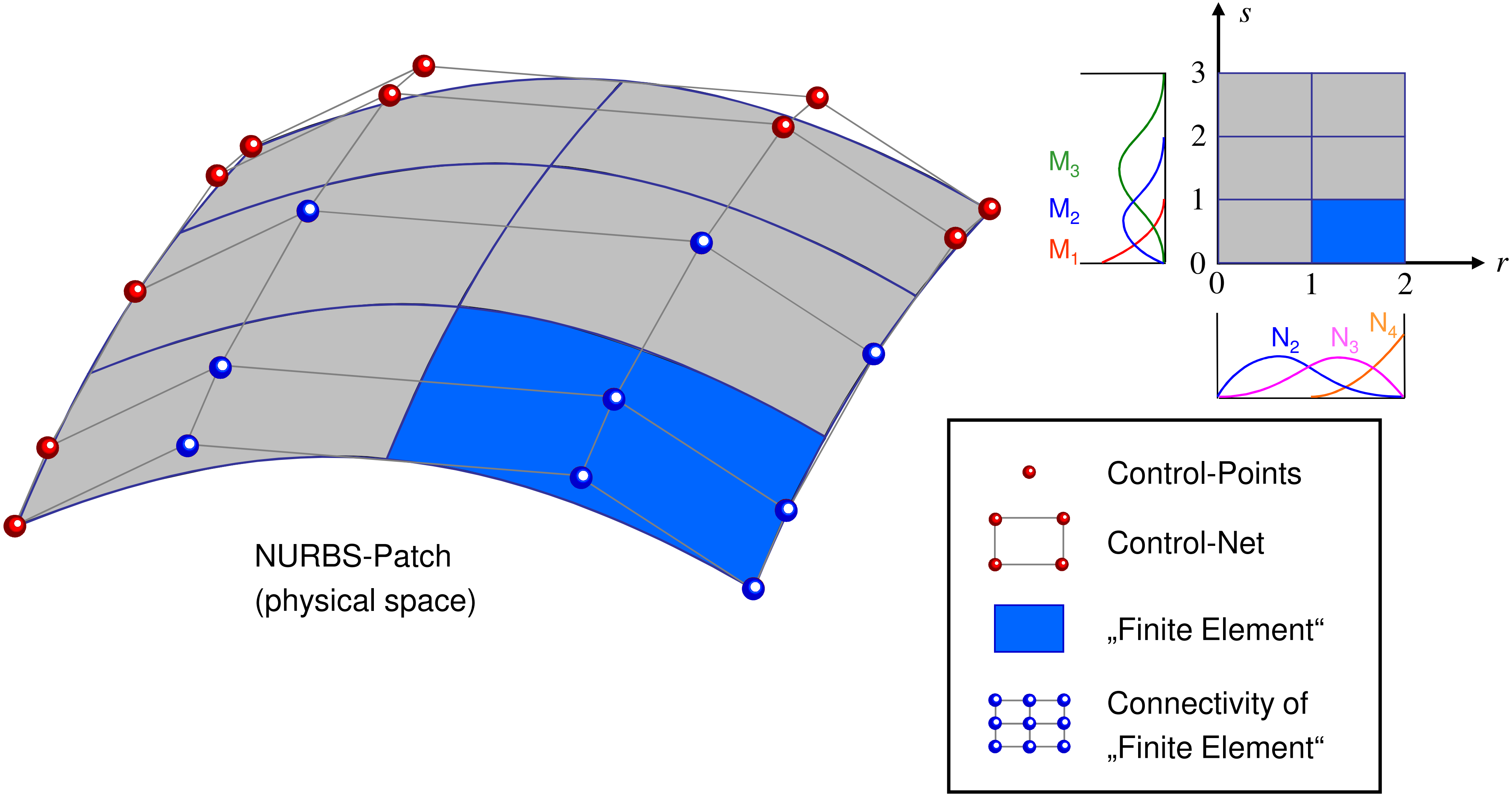




NURBS-Patch
(parameter space)

NURBS-based finite elements in LS-DYNA

- A typical NURBS-Patch – Connectivity of elements
  - Possible „overlaps“ (→ higher continuity!)
  - Size of „overlap“ depends on polynomial order (and on knot-vector)

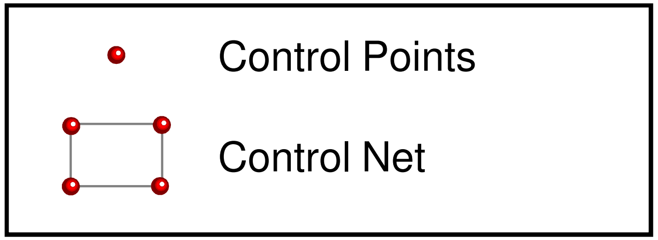


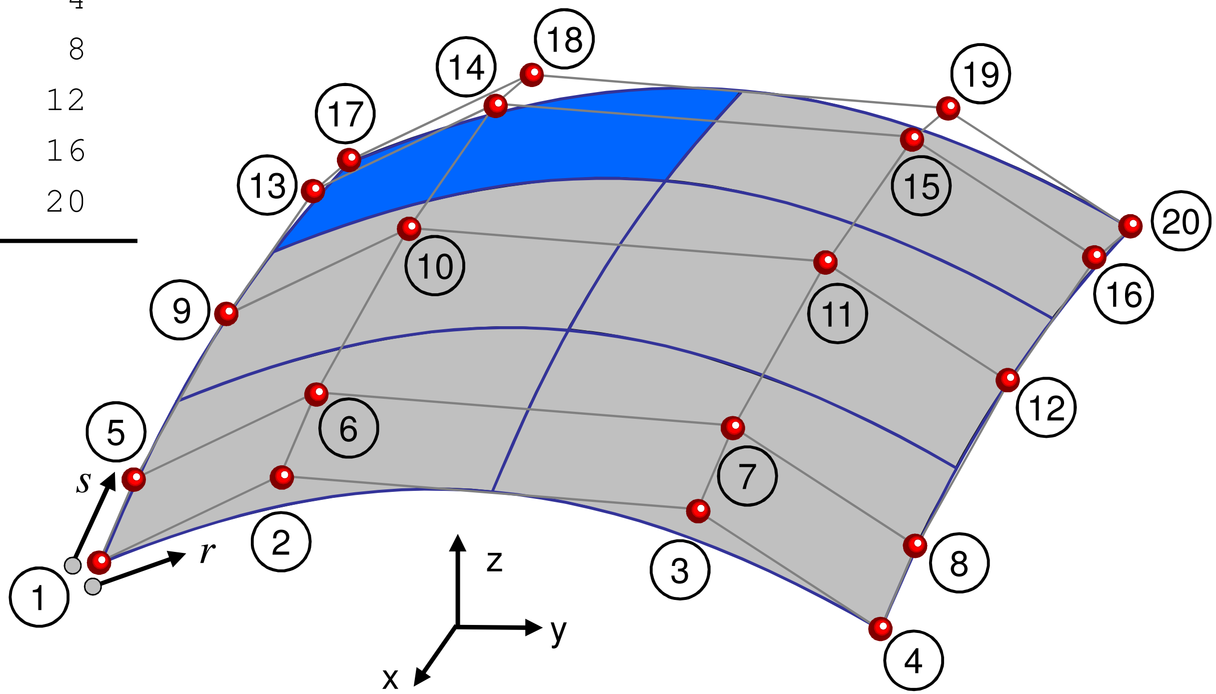
NURBS-based finite elements in LS-DYNA

- New Keyword: *ELEMENT_NURBS_PATCH_2D
  - definition of NURBS-surfaces
  - 4 different shell formulations with/without rotational DOFs (→ generalized shells)
- Pre- and Postprocessing
  - work in progress for LS-PrePost … current status (lspp3.1beta)
    → visualization of 2D-NURBS-Patches
    → import IGES-format and construct *ELEMENT_NURBS_PATCH_2D
    → modification of 2D-NURBS geometry
    → … much more to come!
- Postprocessing and boundary conditions (i.e. contact) currently with
  - interpolation nodes
  - interpolation elements
- Analysis capabilities (→ generalized shells)
  - implicit and explicit time integration
  - eigenvalue analysis
  - other capabilities (e.g. geometric stiffness for buckling) implemented but not yet tested
- LS-DYNA material library available (including umats)

NURBS-based finite elements in LS-DYNA

```
*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
```





NURBS-based finite elements in LS-DYNA

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

- WFL – Flag for weighting factors for control points
  - 0: All weights are 1.0 (no need to define them → B-splines)
  - 1: define weights for control points

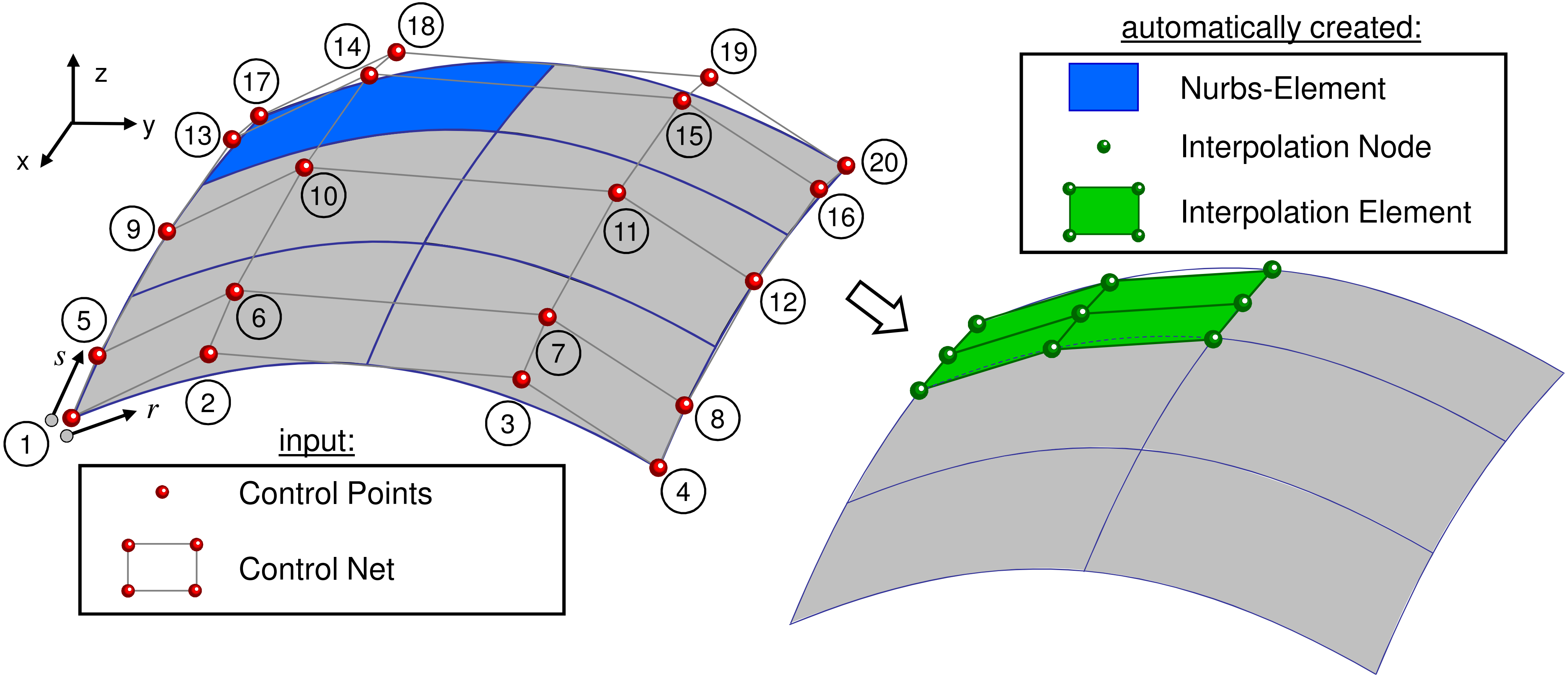
- FORM – Shell formulation to be used
  - 0: "shear deformable theory" with rotational DOFs
  - 1: "shear deformable theory" without rotational DOFs
  - 2: "thin shell theory" without rotational DOFs
  - 3: "thin shell theory" with rotational DOFs

- INT – In-plane integration rule
  - 0: reduced (Gauss-)integration (NIP=PR*PS)
  - 1: full (Gauss-)integration (NIP=(PR+1)*(PS+1)
  - ?: "Half-Point-Rule" (→ A. Reali)

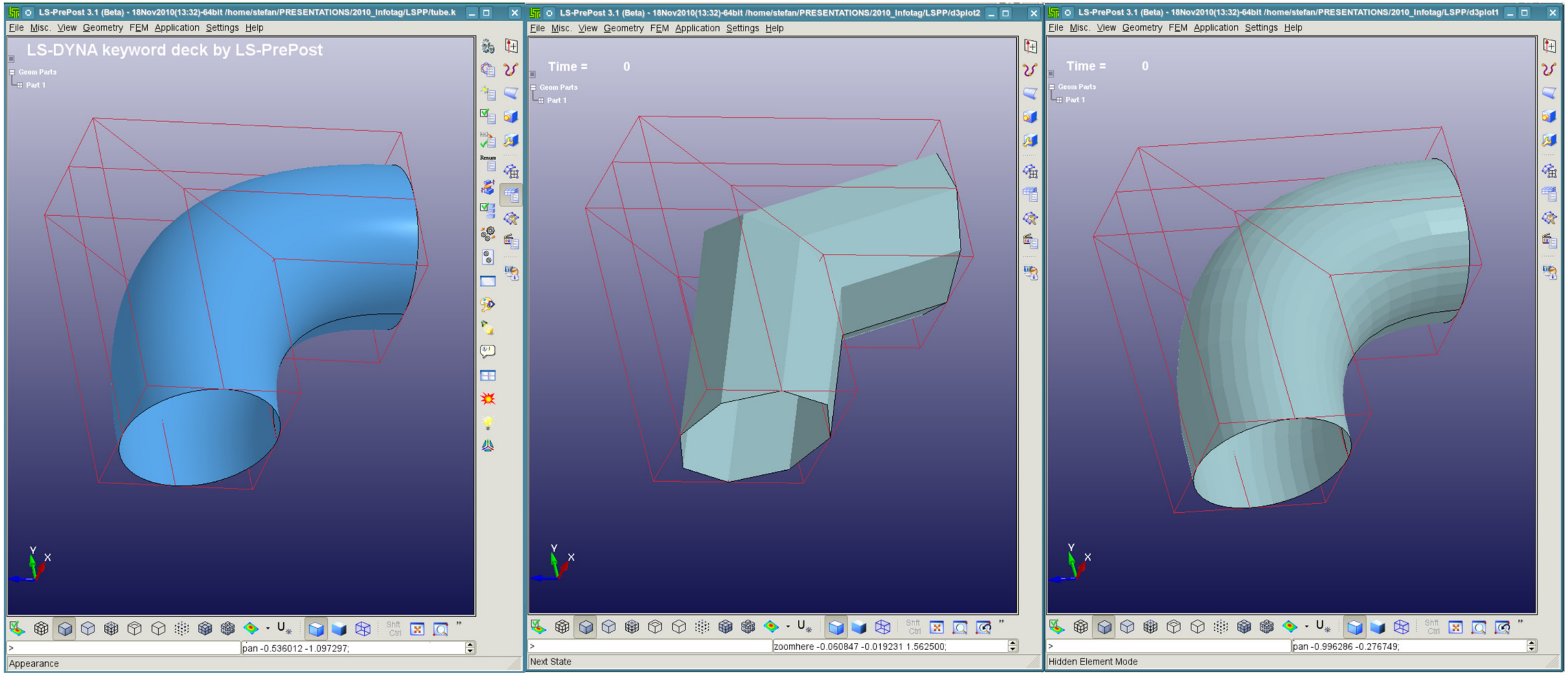
NURBS-based finite elements in LS-DYNA

```
...
$---+--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



NURBS-based finite elements in LS-DYNA



LSPP: Preprocessing

- control-net
- nurbs surface

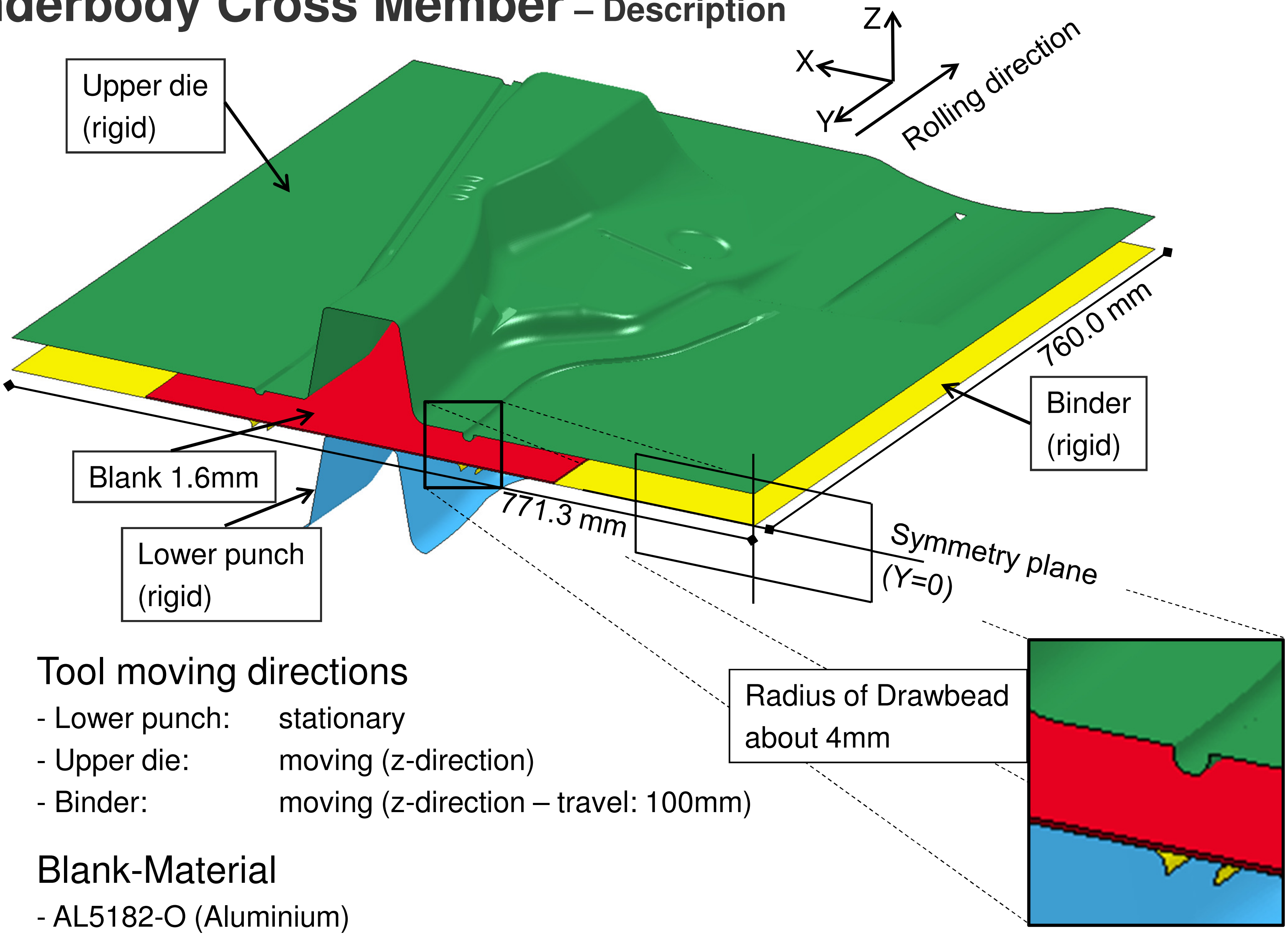
LSPP: Postprocessing

- Interpolation nodes/elements

nisr=niss=2

nisr=niss=10

Underbody Cross Member – Description



- Tool moving directions
  - Lower punch: stationary
  - Upper die: moving (z-direction)
  - Binder: moving (z-direction – travel: 100mm)
- Blank-Material
  - AL5182-O (Aluminium)

Underbody Cross Member – Simulation models

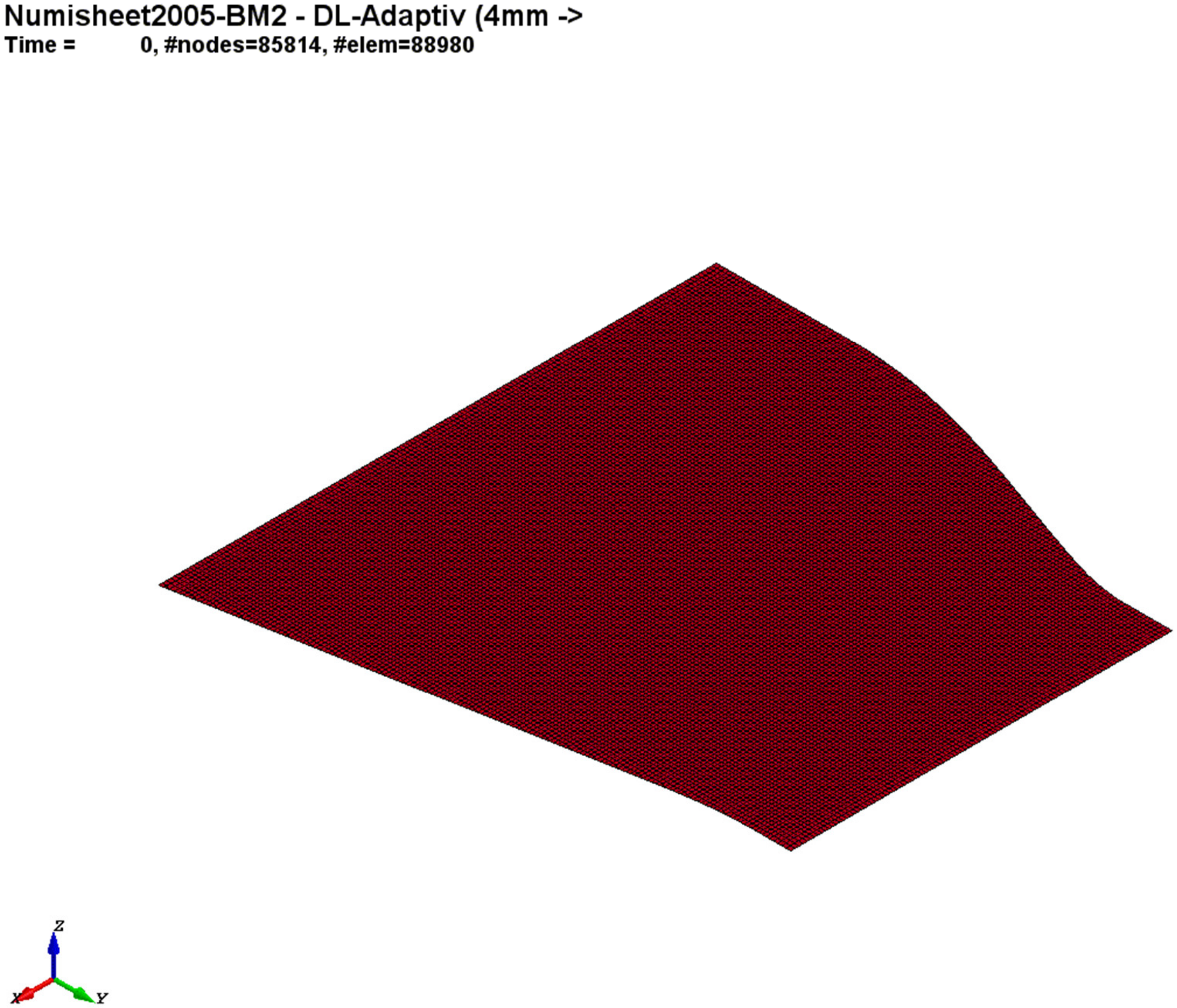
- identical for all
  - material model: *MAT_TRANSVERSELY_ANISOTROPIC_ELASTIC_PLASTIC (*MAT_037)
  - nip=5 number of integration points through the thickness
  - istupd=0 no thickness update
  - imscl=0 no “selective” mass scaling (no mass scaling at all!)
  - SMP, double precision, ncpu=4 (Dual Core AMD Opteron, 2.2 GHz)

- standard elements
  - ELFORM=16: fully integrated (4-noded) shell-elements with assumed strain formulation
  - discretizations: with adaptivity (mesh size: 4mm → 2mm → 1mm) as reference solution
    without adaptivity: mesh-sizes: 2mm; 4mm; 8mm

- 2D-NURBS elements
  - Formulation: FORM=2 (rotation free formulation)
  - Integraion rule: INT=0 (reduced integration)
  - Polynomial: p2 (quadratic), p3 (cubic), p4 (quartic), p5 (quintic)
  - discretizations: mesh-sizes: 4mm; 8mm; 16mm
  - number of interpolation elements/ NURBS-elements: NISR=PR; NISS=PS

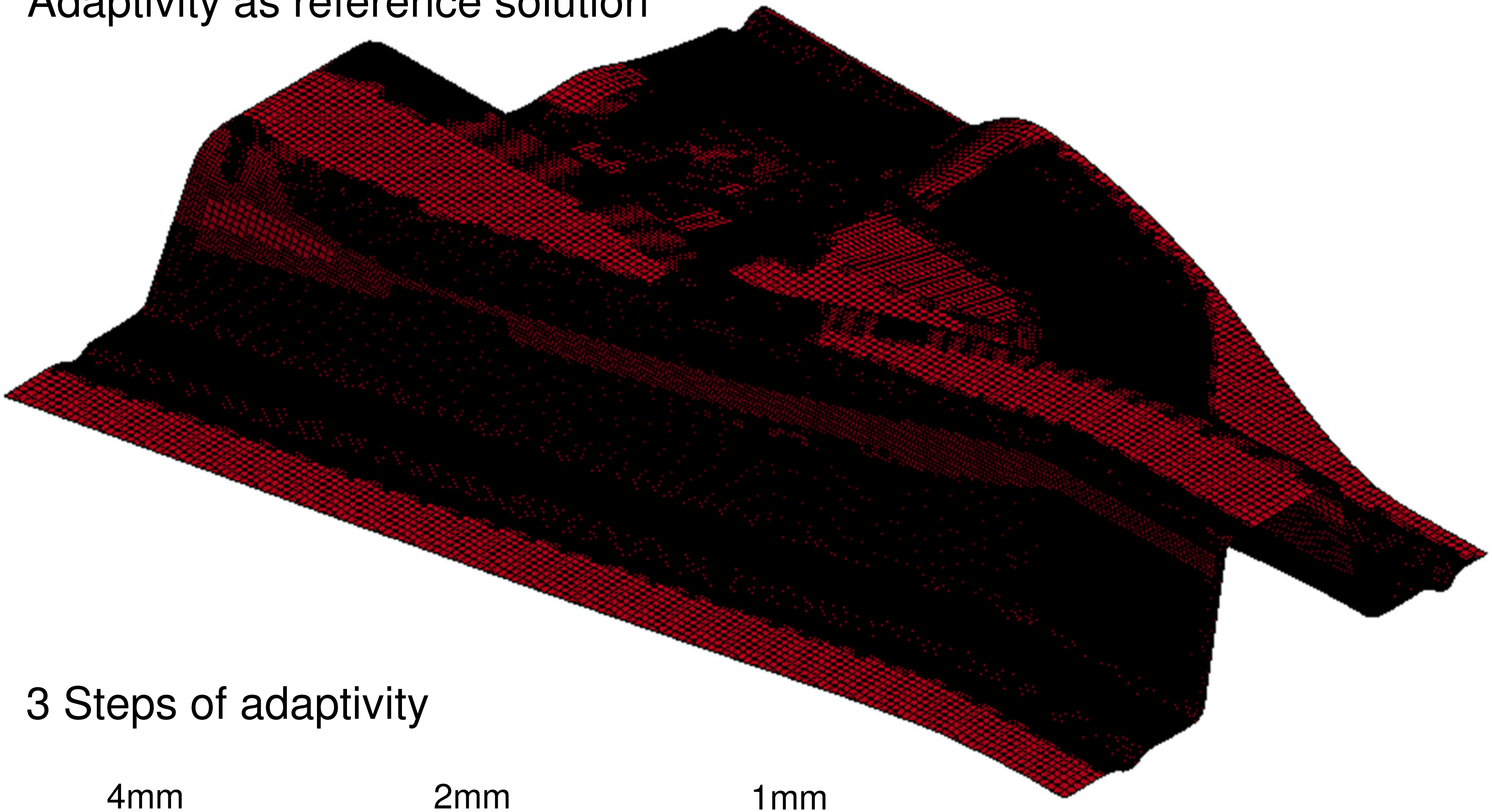
Underbody Cross Member – Standard-Elements

- Adaptivity as reference solution

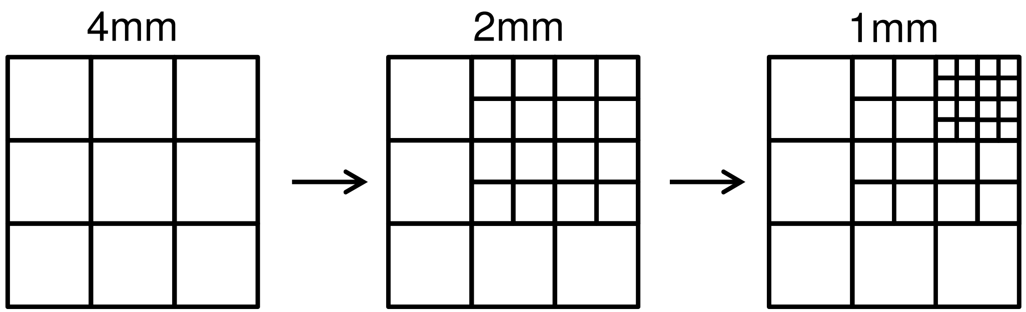


Underbody Cross Member – Standard-Elements

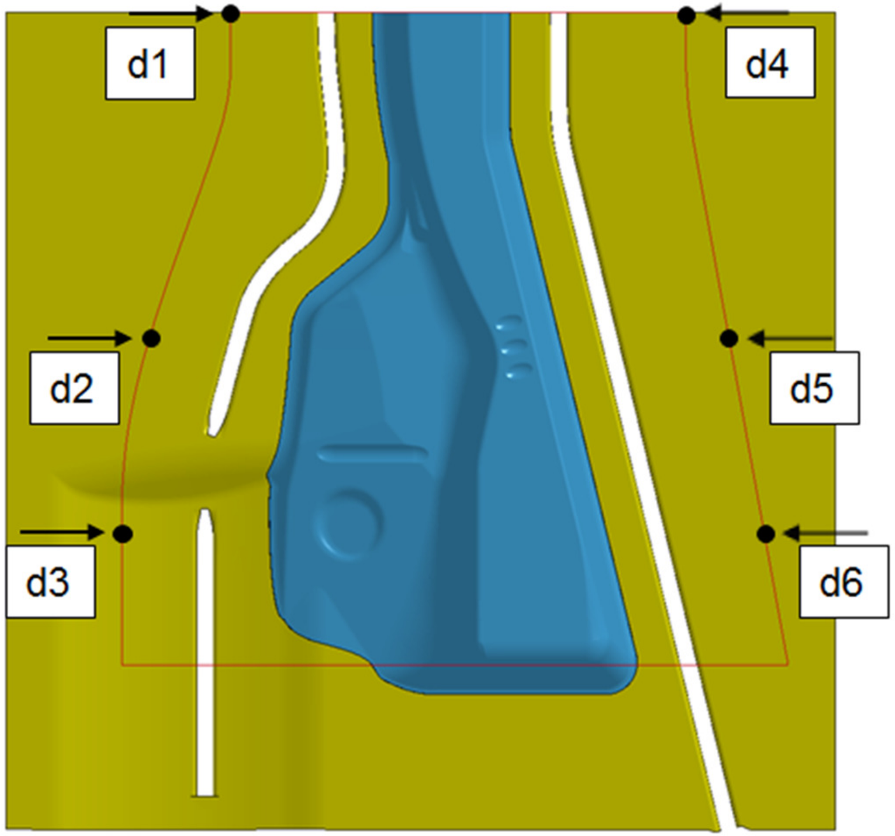
- Adaptivity as reference solution

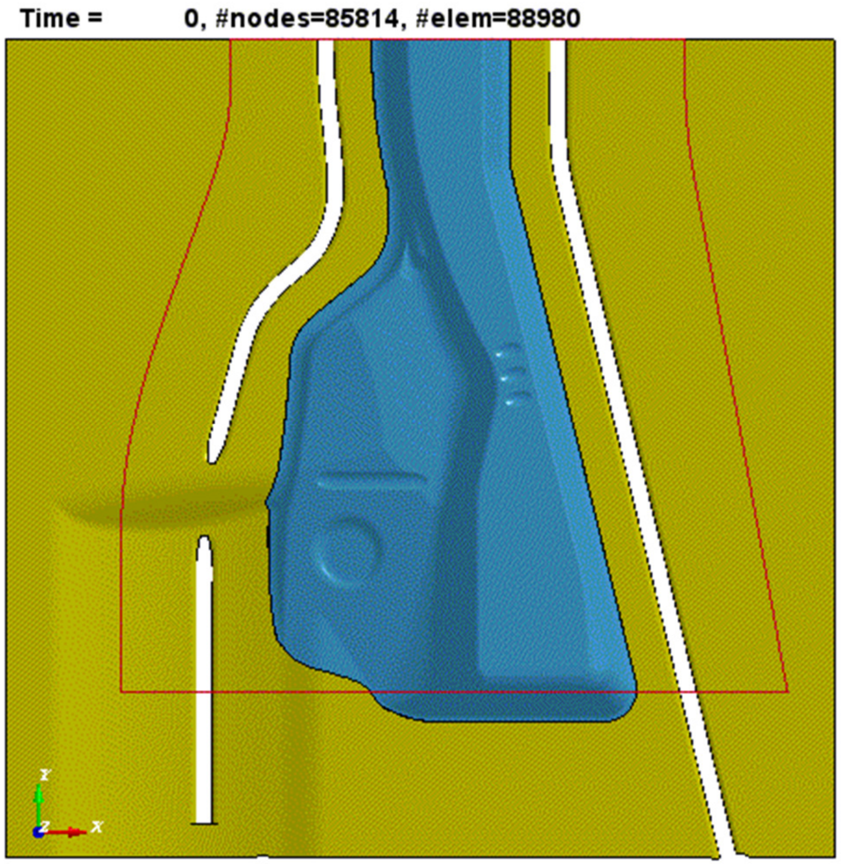


- 3 Steps of adaptivity



Underbody Cross Member – Draw-in

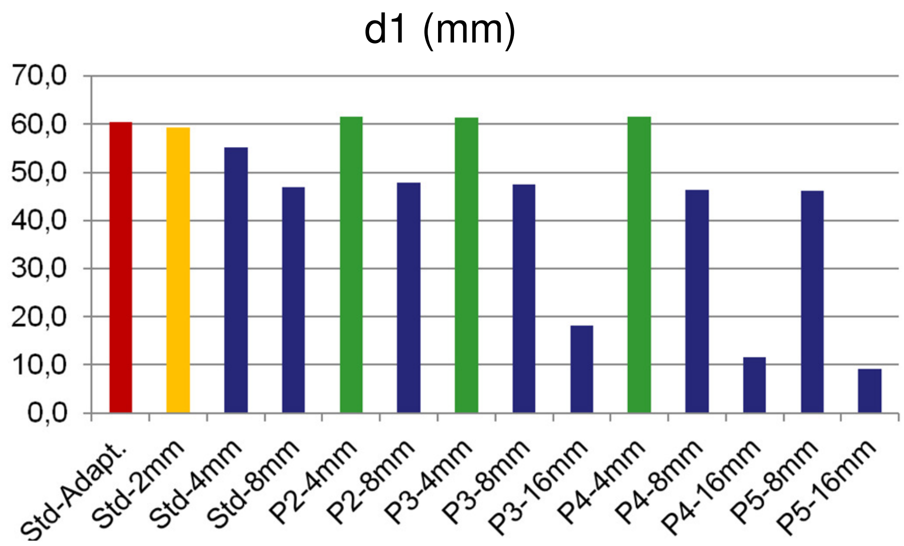


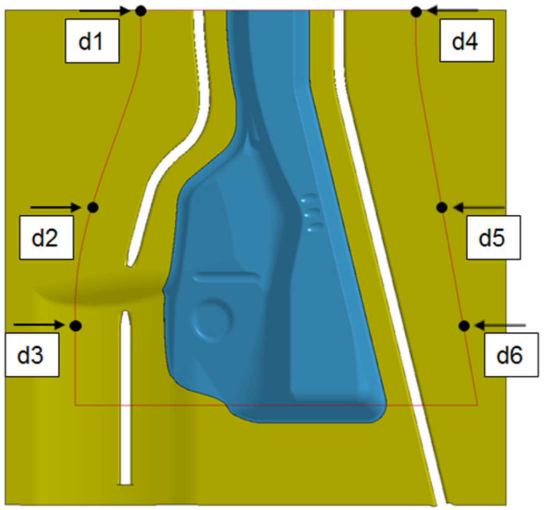


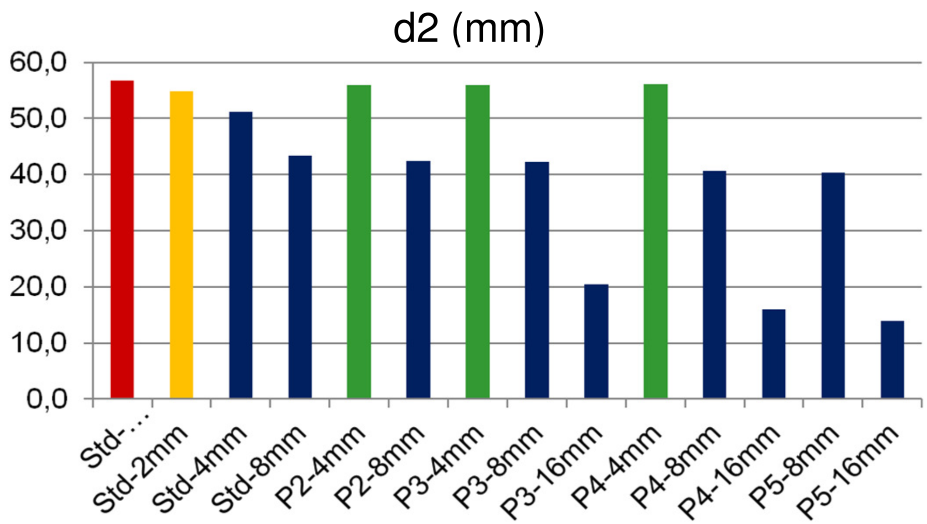
- Results → Use "Adaptiv" as reference solution

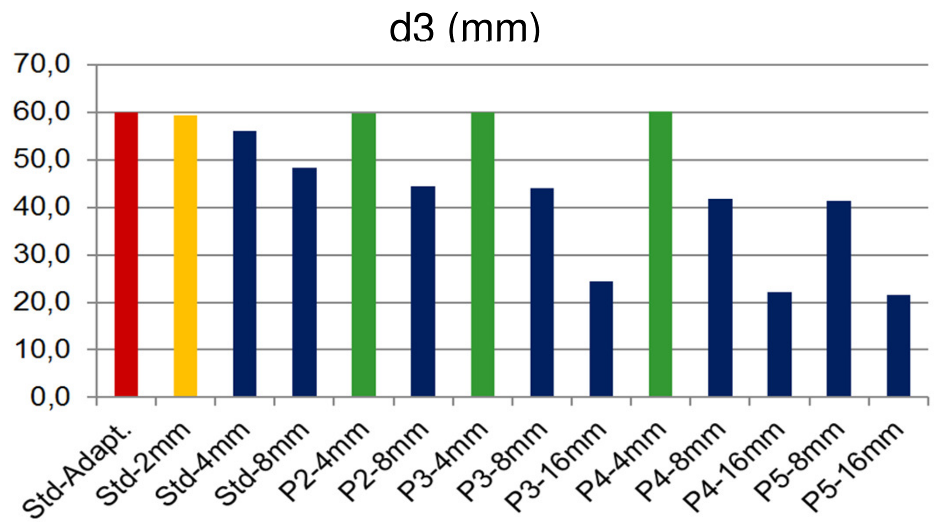
	d1(mm)	d2(mm)	d3(mm)	d4(mm)	d5(5mm)	d6(6mm)
Benchmark	62.2	51.8	56.0	73.7	57.6	47.8
Adaptiv	60.5	56.8	59.9	74.8	57.5	50.4

Underbody Cross Member – Draw-in



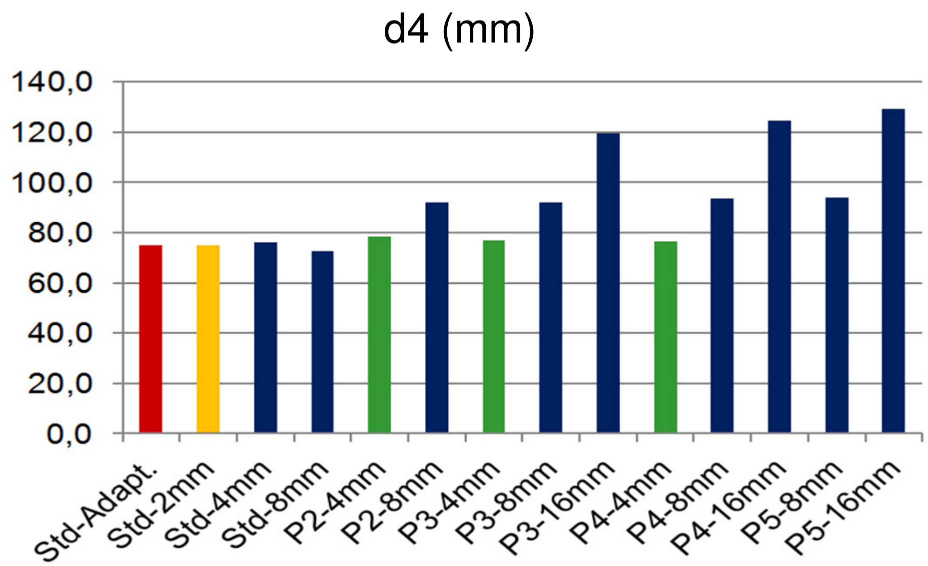


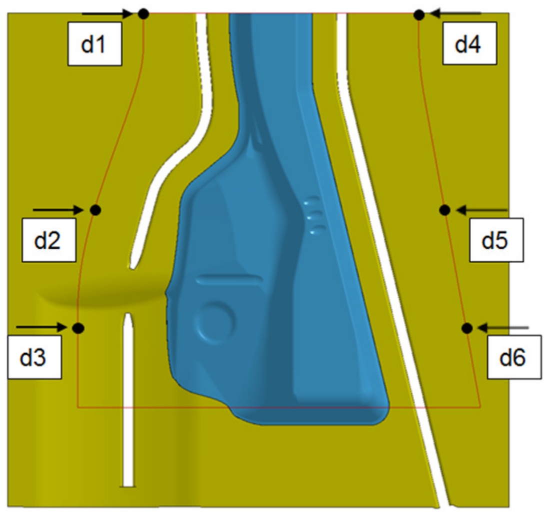


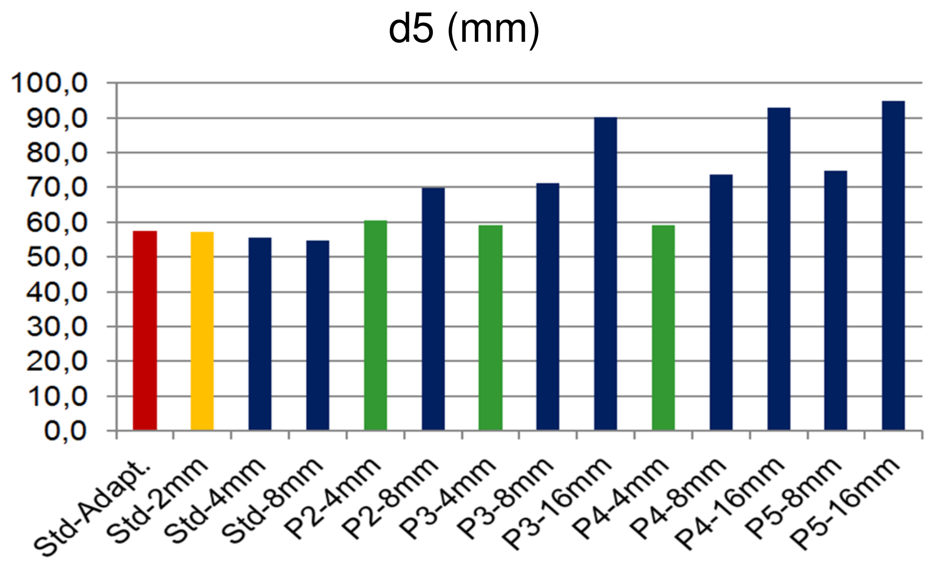


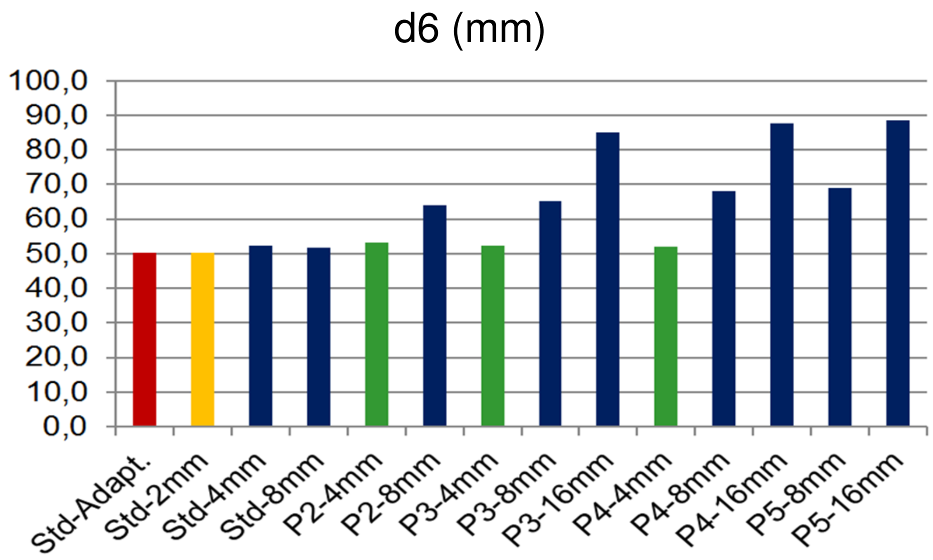
- larger mesh size → less draw-in (behavior is too stiff)

Underbody Cross Member – Draw-in

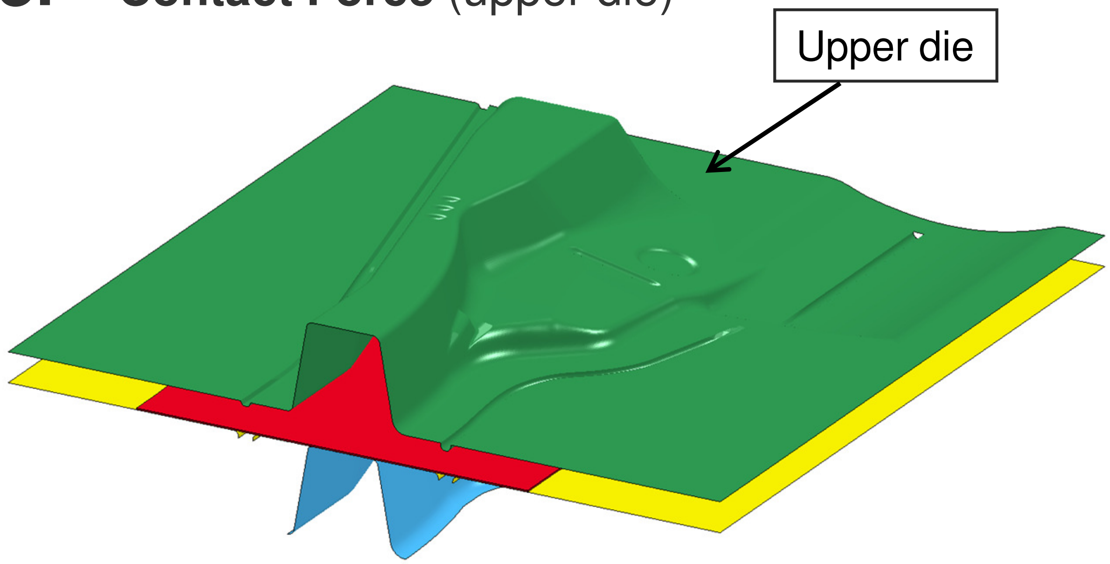


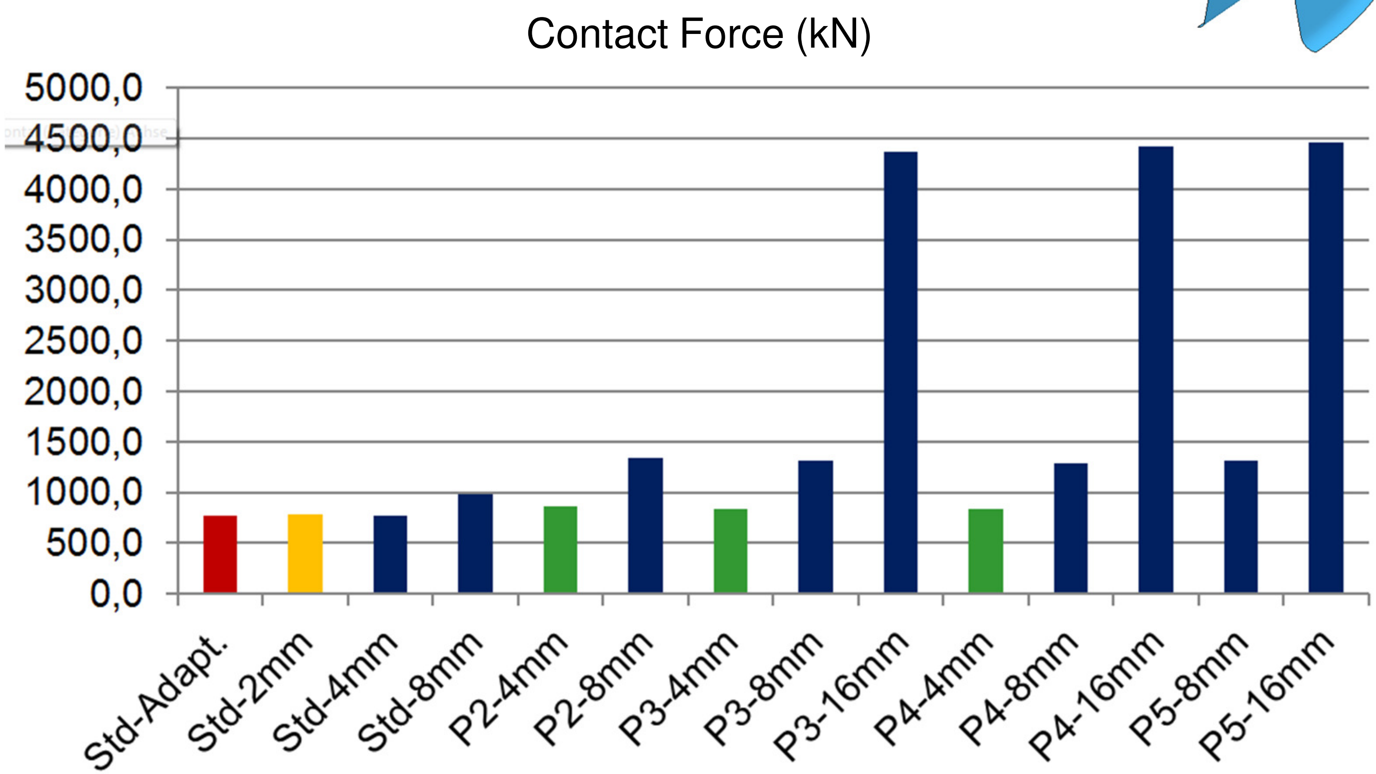




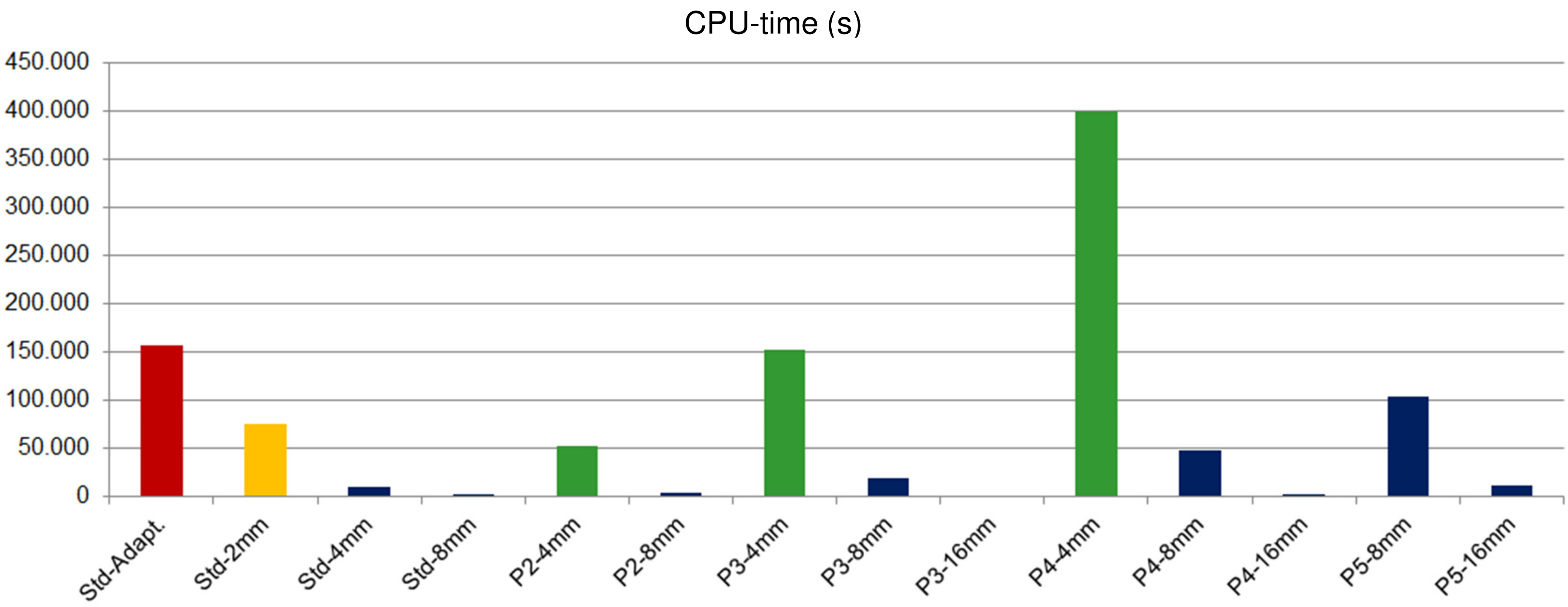


Underbody Cross Member – Contact Force (upper die)





Underbody Cross Member – CPU-time

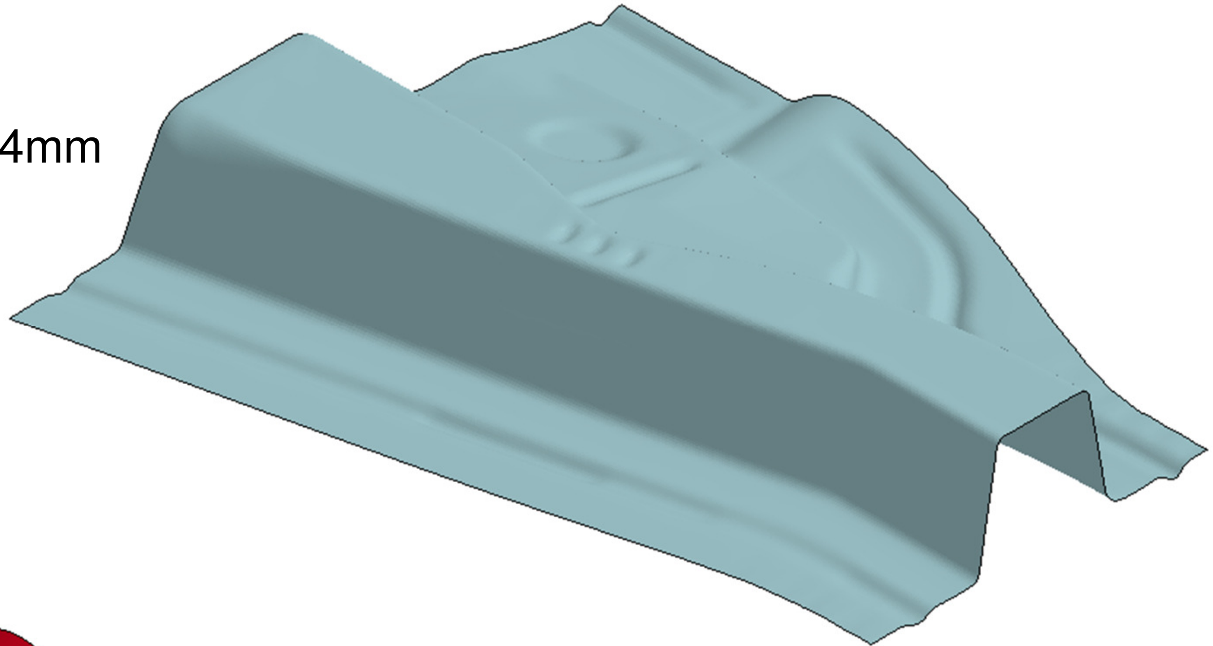


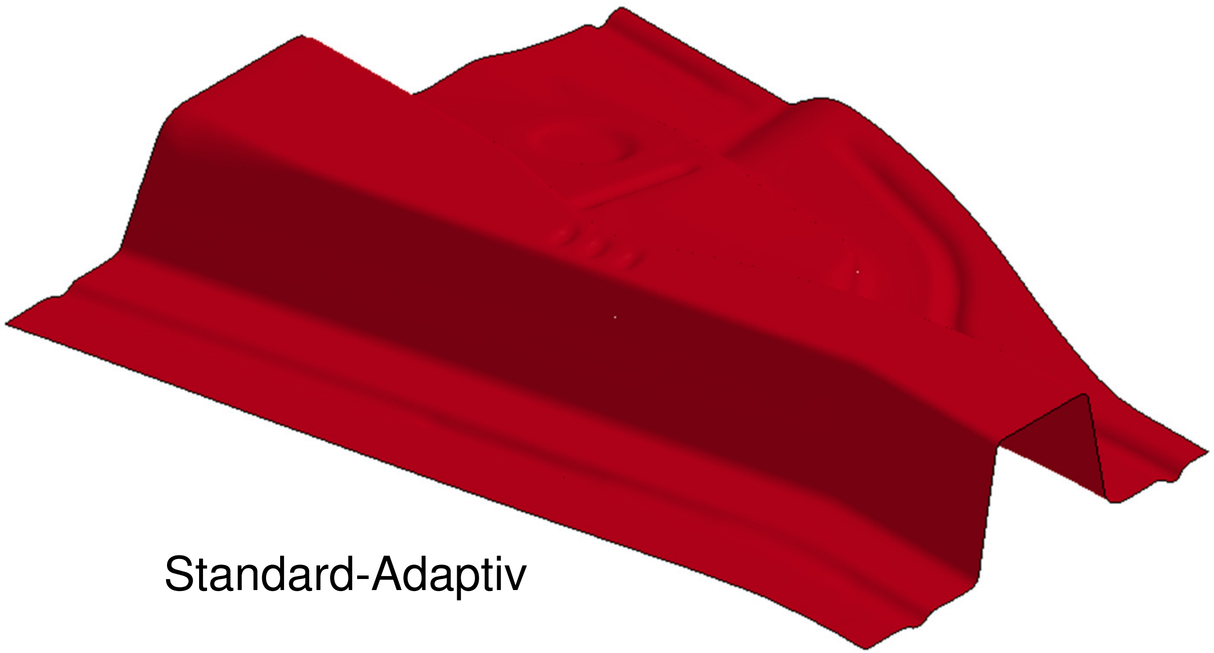
43:46 21:06 2:55 0:43 14:31 1:06 42:12 5:19 0:25 110:52 13:26 0:55 28:45 3:08

CPU-time (h:min)

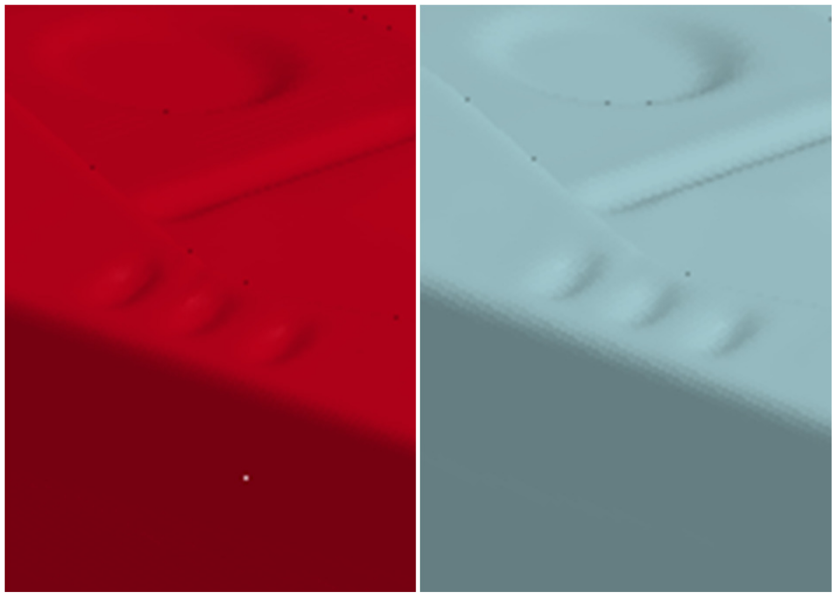
Underbody Cross Member – Final deformation

NURBS-P2-4mm





Standard-Adaptiv



Underbody Cross Member – Summary

Radius of Drawbead about 4mm

- Detailed discretization of “Drawbead” needs a fine discretization (<=4mm), no matter what type of elements
- Rotation free elements with reduced integration show best behavior
- CPU-time for comparable discretizations (i.e: p1_2mm ←→ p2_4mm) are promising (no CODE optimization yet!)→ cost competitive
- CPU-time increase for NURBS with same discretizations for next order of polynomial (i.e.: p2_4mm→p3_4mm): Factor 2.5-2.8
- Higher order does not help anything in this example
  (spacing of control points define mesh size)

Summary

- NURBS-based elements run stable
- higher order accurate isogeometric analysis can be cost competitive
  - but missing a couple of "special" issues for industrial sheet metal forming applications
- code optimization necessary to make it faster
- in this example: geometry dictates the mesh size (independent of polynomial order!)

Outlook

- perform a lot more studies in different fields → experience
- motivate customers (and **researchers**) to "play" with these elements
- further implementation
  - make NURBS elements work with MPP
  - (selective) mass scaling
  - thickness update of shells
  - use NURBS for contact (instead of interpolation elements)
  - make pre- and post-processing more user-friendly
  - introduce 3D NURBS elements
  - … much more

Thank you!