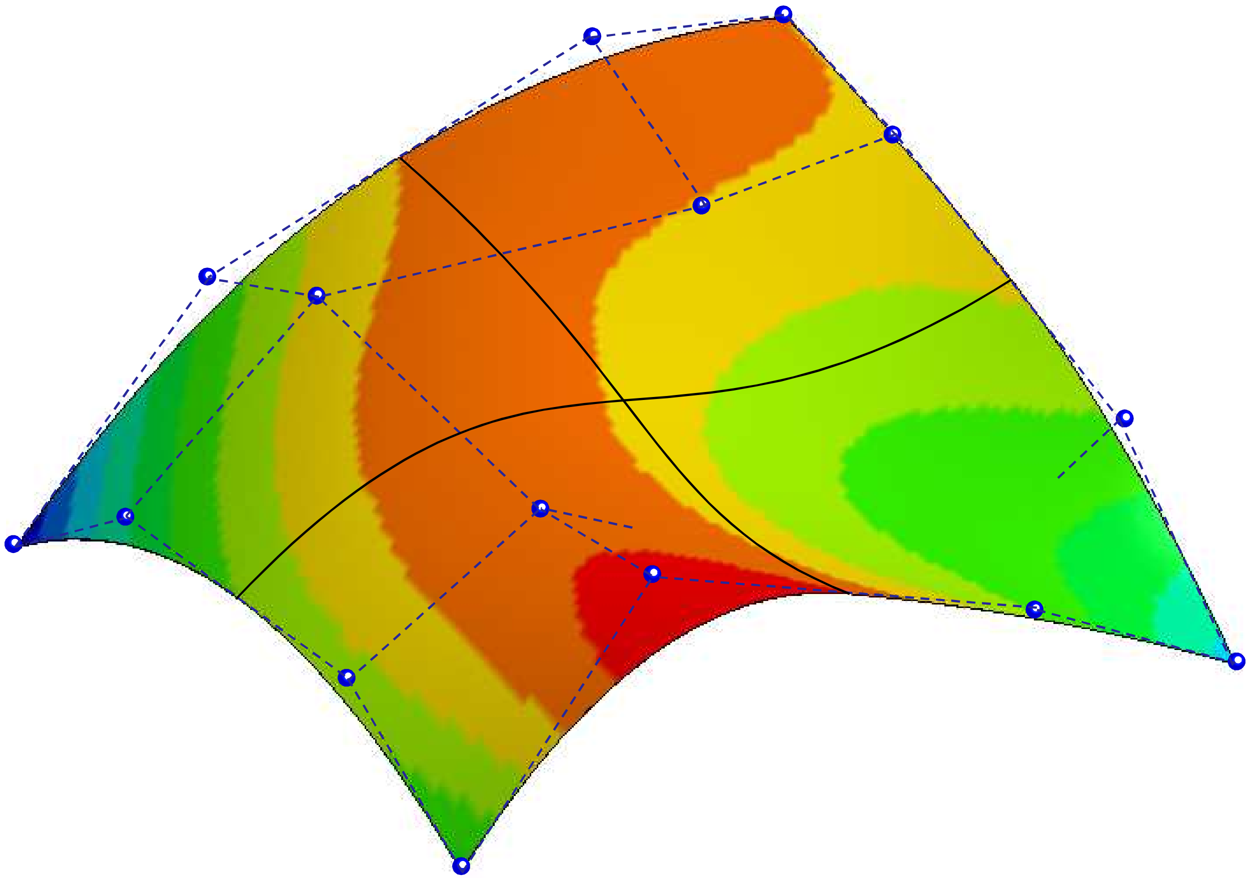
Current Status of Isogeometric Analysis in LS-DYNA



Stefan Hartmann

Developer Forum, September 24th, 2013, Filderstadt, Germany

Development in cooperation with:
D.J. Benson: Professor of Applied Mechanics, University of California, San Diego, USA

Outline

- Isogeometric Analysis
  - motivation / definition
- NURBS-based finite elements
  - NURBS / *ELEMENT_SHELL_NURBS_PATCH / elements - connectivity
- Shell formulations
  - continuity / rotation free shells / blended shells
- Contact with NURBS
  - smoothness / example
- Current status
  - MPP / examples / limits / LSPP / …
- Todo-List
  - mass-scaling / merging / trimmed NURBS / NURBS-solids / LSPP / …

Isogeometric Analysis – motivation & definition

- reduce effort of geometry conversion from CAD into a suitable mesh for FEA (?)
- reduce discretization errors → adaptivity
- increase continuity of geometry representation → contact / rot. free shells / …

- ISOPARAMETRIC (FE-Analysis)
  use same approximation for geometry and deformation

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

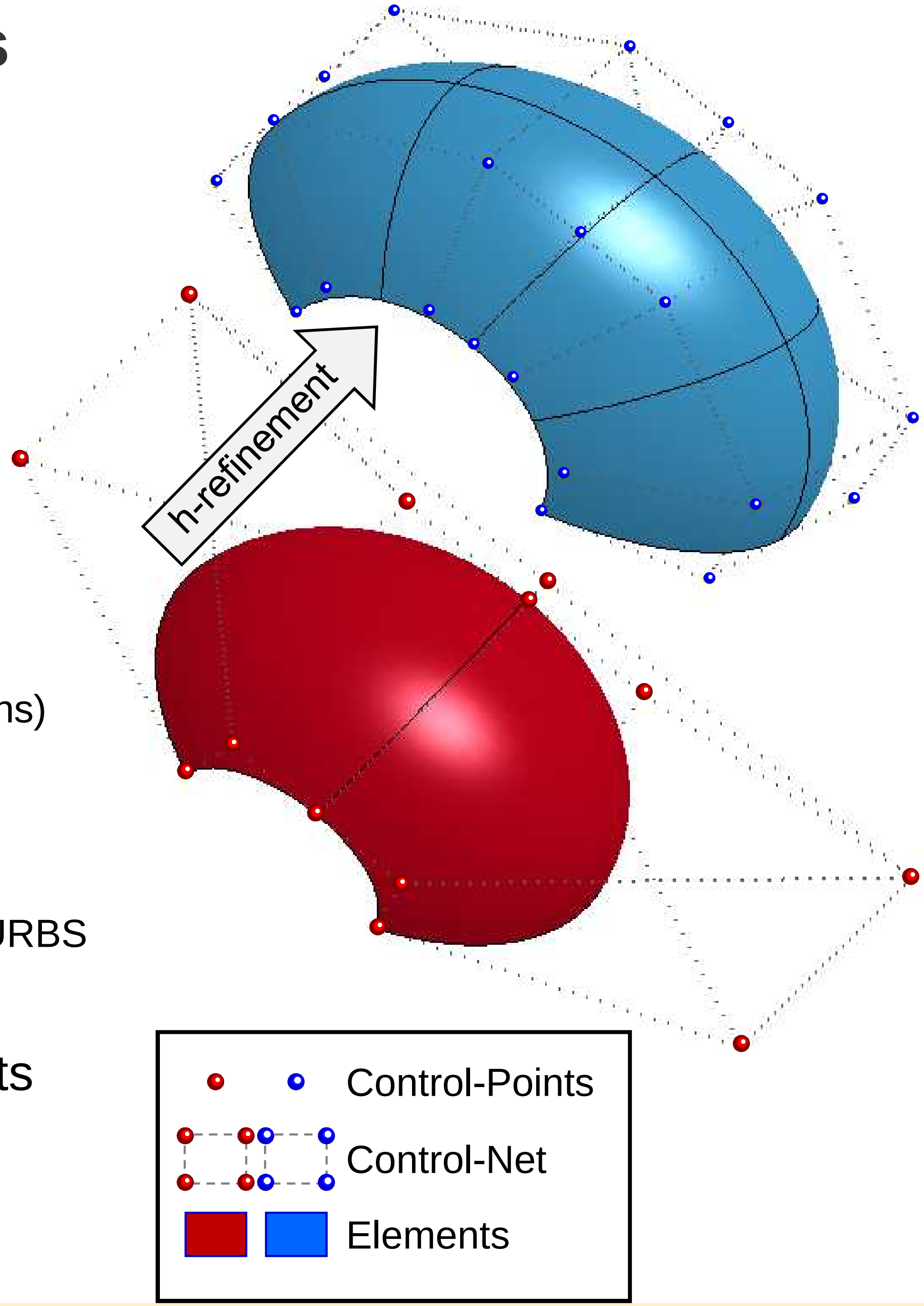
NURBS-based finite elements

- NURBS = Non Uniform Rational Basis Spline

- B-spline basis functions
  - recursive
  - positive everywhere
  - dependent on knot-vector and 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)

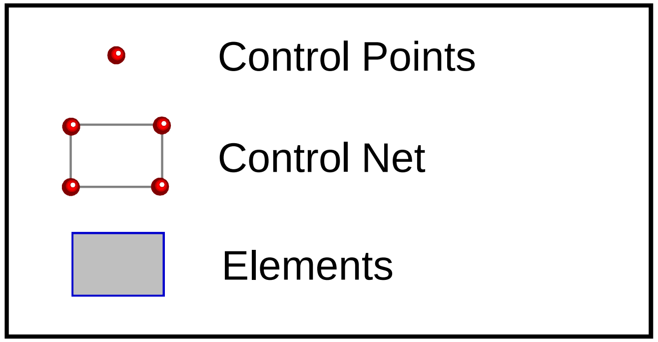
- NURBS
  - B-spline basis functions + control net with weights
  - all mentioned properties for B-splines apply for NURBS

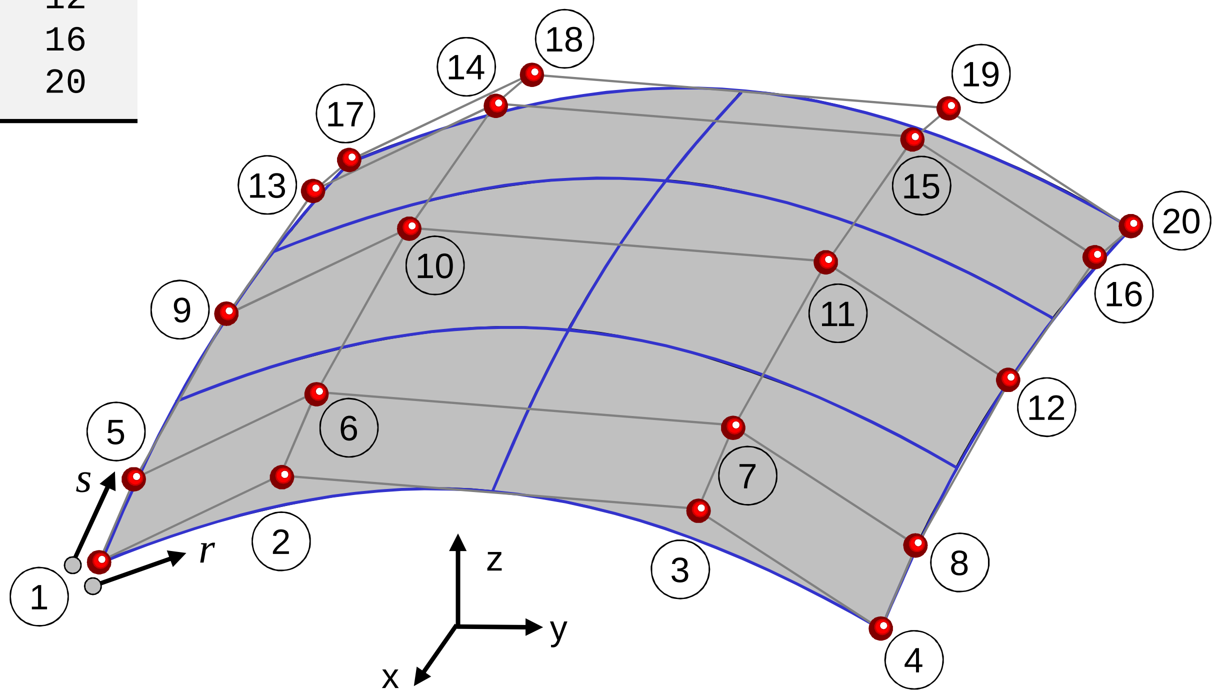
- A typical NURBS-Patch and the elements
  - elements are defined through the knot-vectors
  - shape functions for each control-point (→DOFs)



NURBS-based finite elements

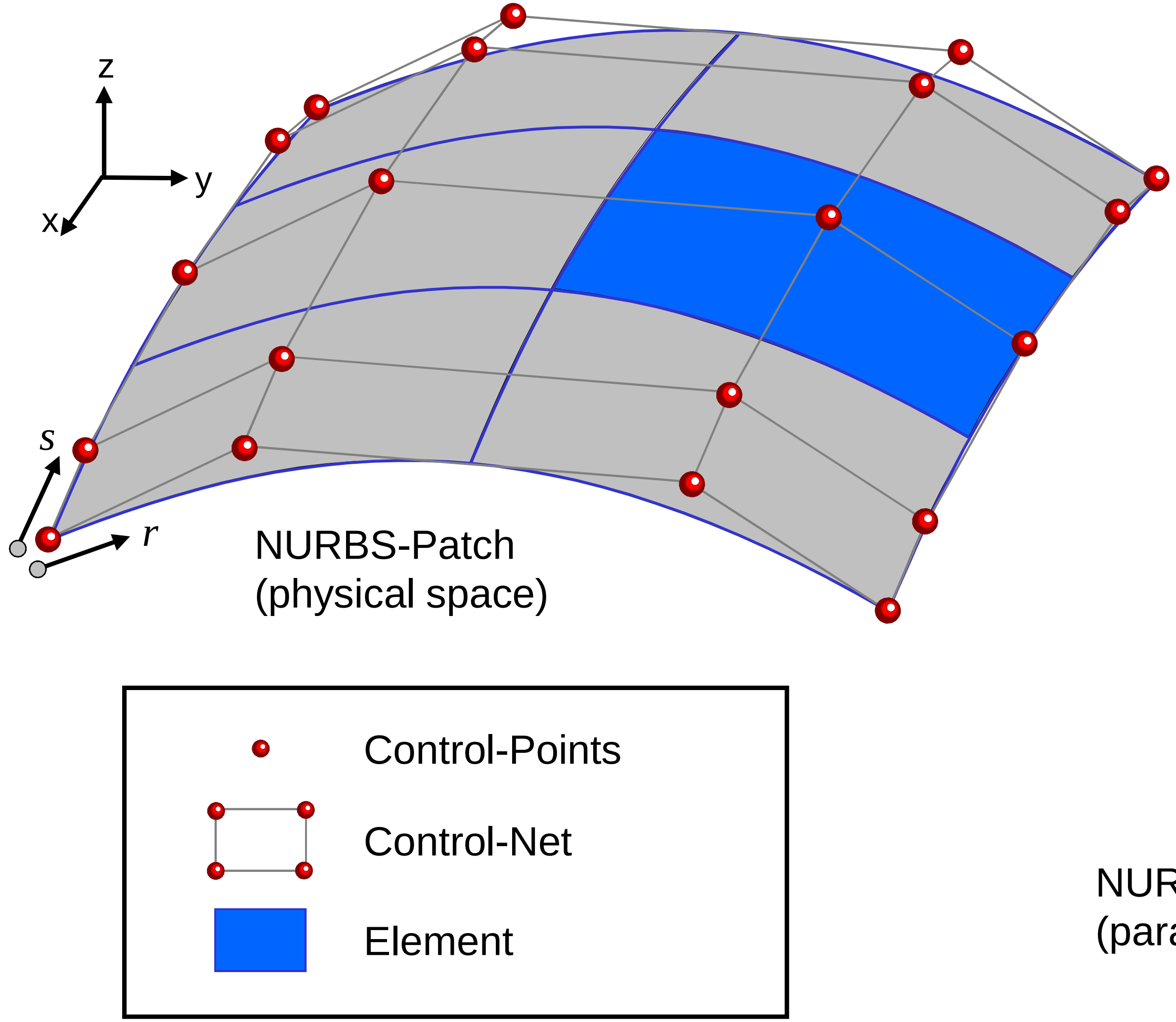
```
*ELEMENT_SHELL_NURBS_PATCH
$---+--EID----+--PID----+--NPR----+---PR----+--NPS----+---PS----+----7----+----8
        11        12         4         2         5         2
$---+--WFL----+-FORM----+--INT----+-NISR----+-NISS----+IMASS----+----7----+----8
         0         0         1         2         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
```



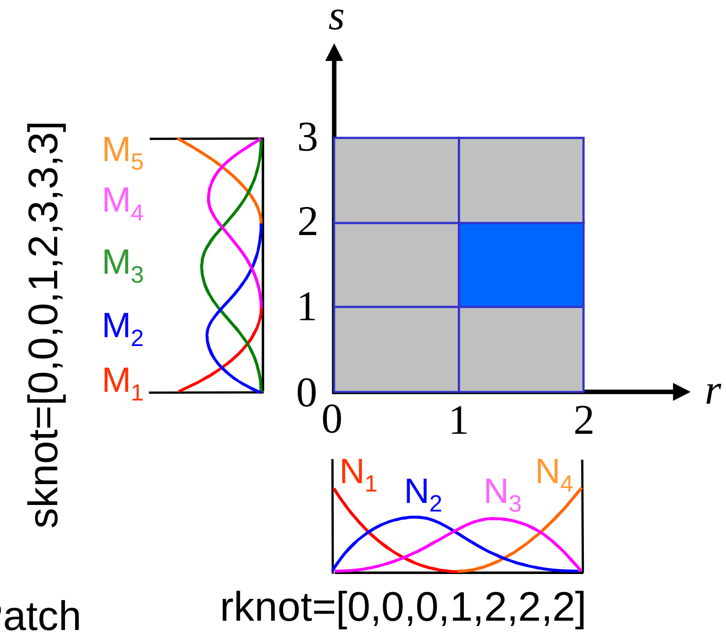


NURBS-based finite elements

- 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

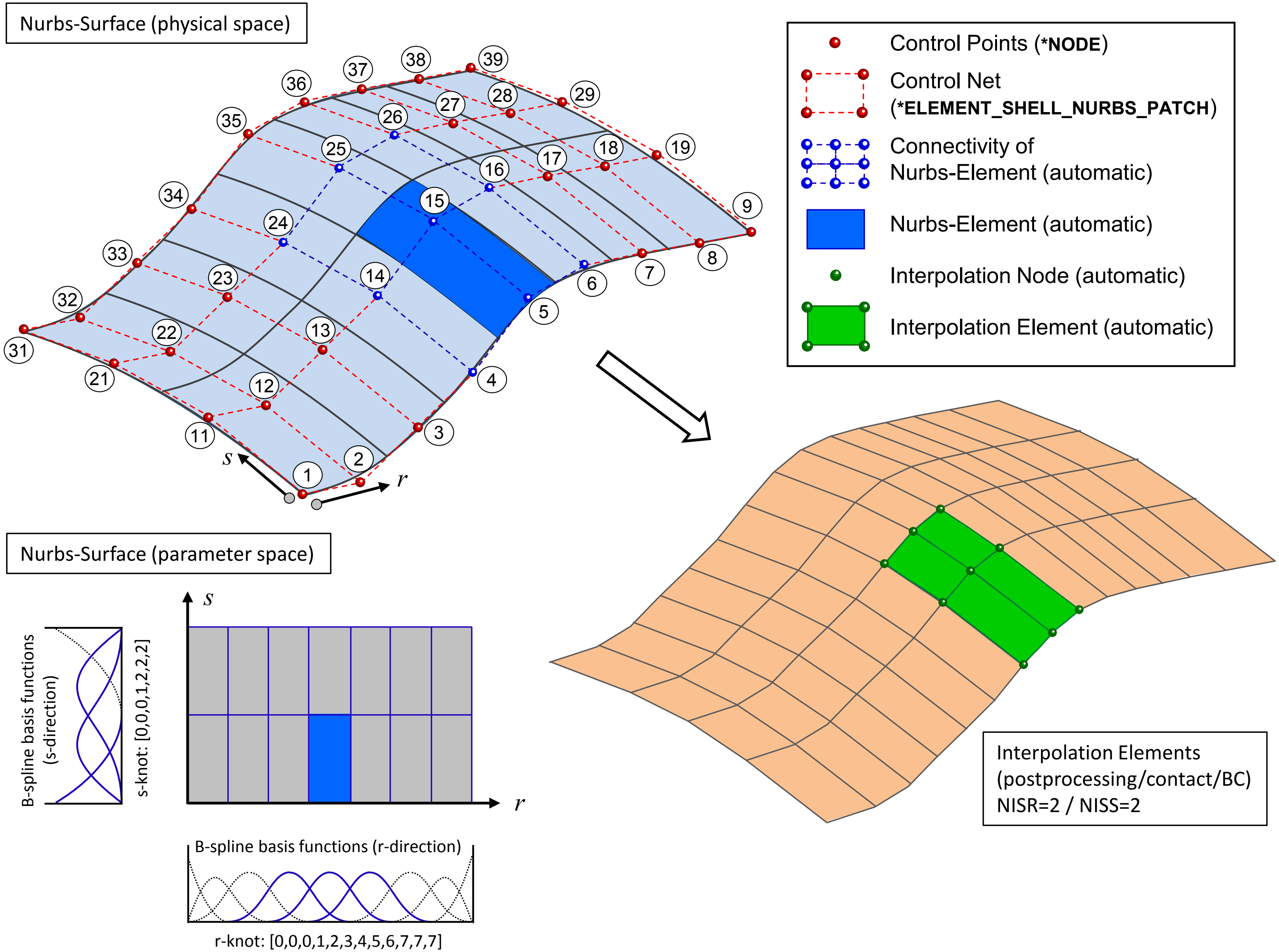


polynomial order:
- quadratic in r-direction (pr=2)
- quadratic in s-direction (ps=2)



NURBS-Patch
(parameter space)

NURBS-based finite elements



from:

LS-DYNA®

KEYWORD USER'S MANUAL

VOLUME I

Livermore Software Technology Corporation (**LSTC**)

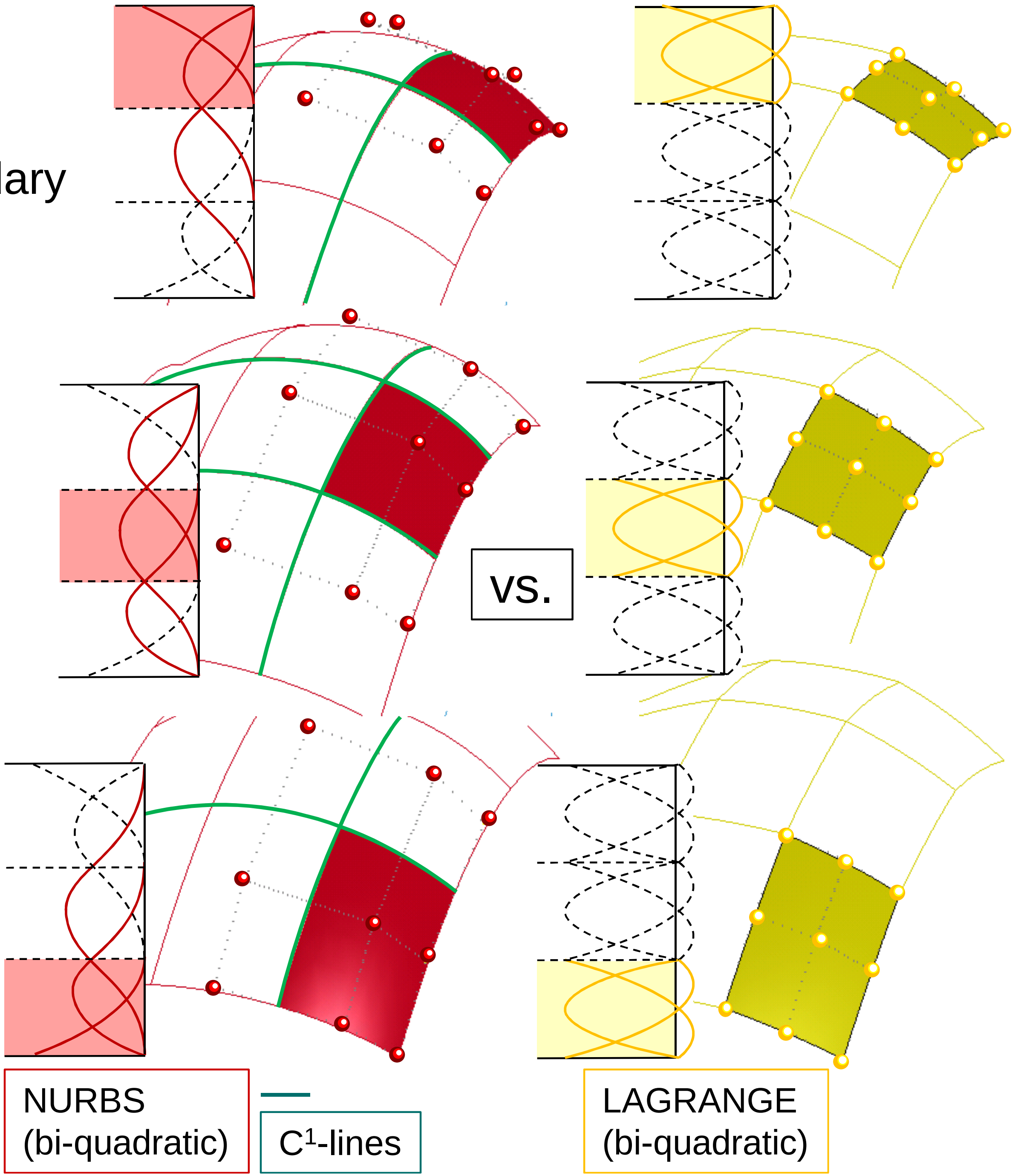
Shell formulations

- continuity at element boundary
  - Lagrange: C^0
  - NURBS: C^{p-1}
    support of shape functions
    (control points) do overlap

- FORM=0:
  - shear deformable shell theory
    with rotational DOFs
    (6DOFs/control point)

- FORM=1: (requires C^1)
  - rotation free shell theory
    without rotational DOFs
    (3DOFs/control point)

- FORM=4/-4:
  - hybrid shell formulation
    combination of FORM=0 and
    FORM=1 (especially necessary
    at patch boundaries where
    continuity drops to C^0)



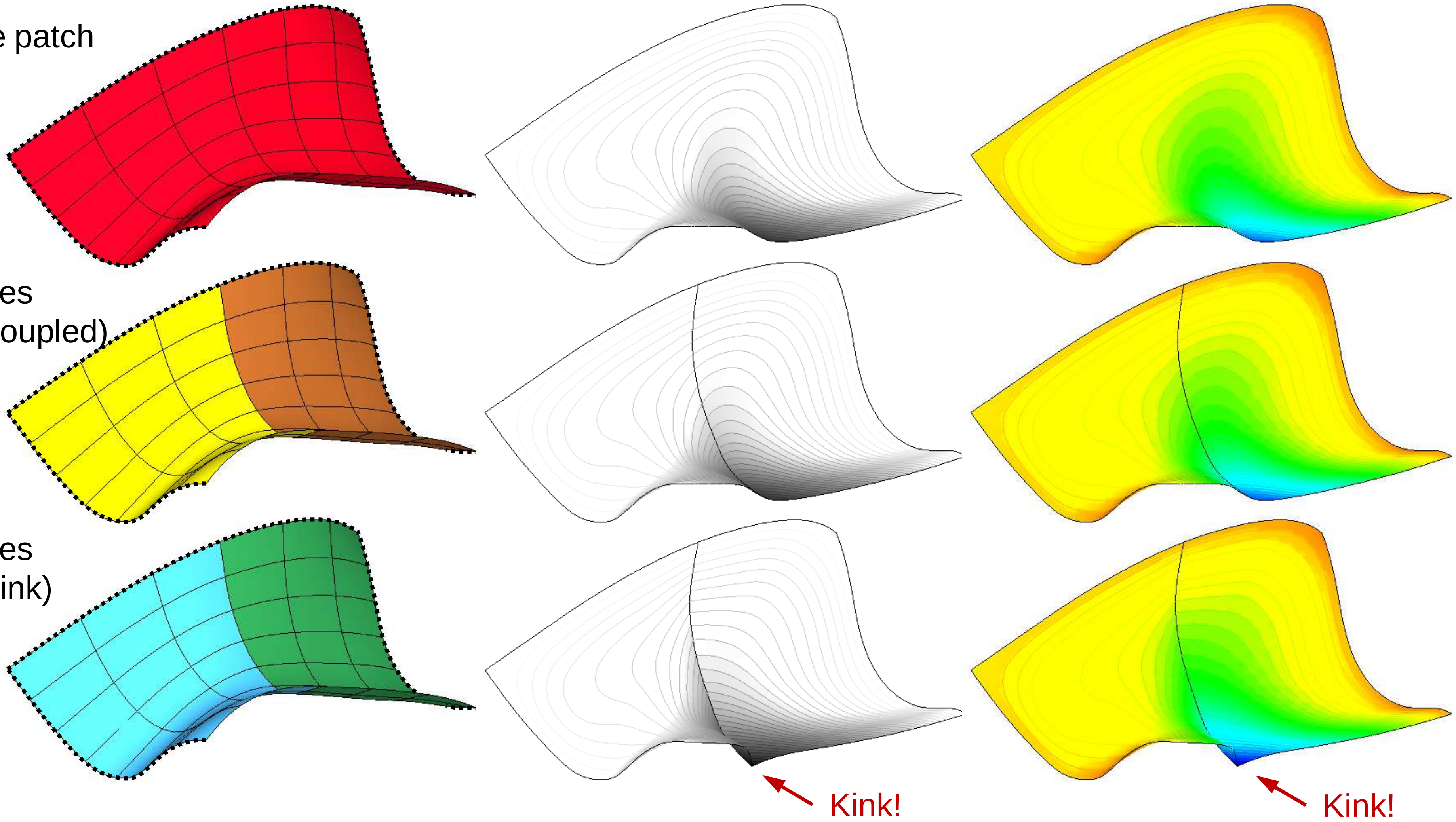
Shell formulations – multi-patch coupling (rotation free formulation)

- G^1 continuity along patch boundaries
  - at patch boundaries: C^0-lines (no transmission of bending moments with rot-free formulation)
  - add rotational DOFs at patch boundary (hybrid/blended shell: FORM=4/-4)

One single patch
(rot-free)

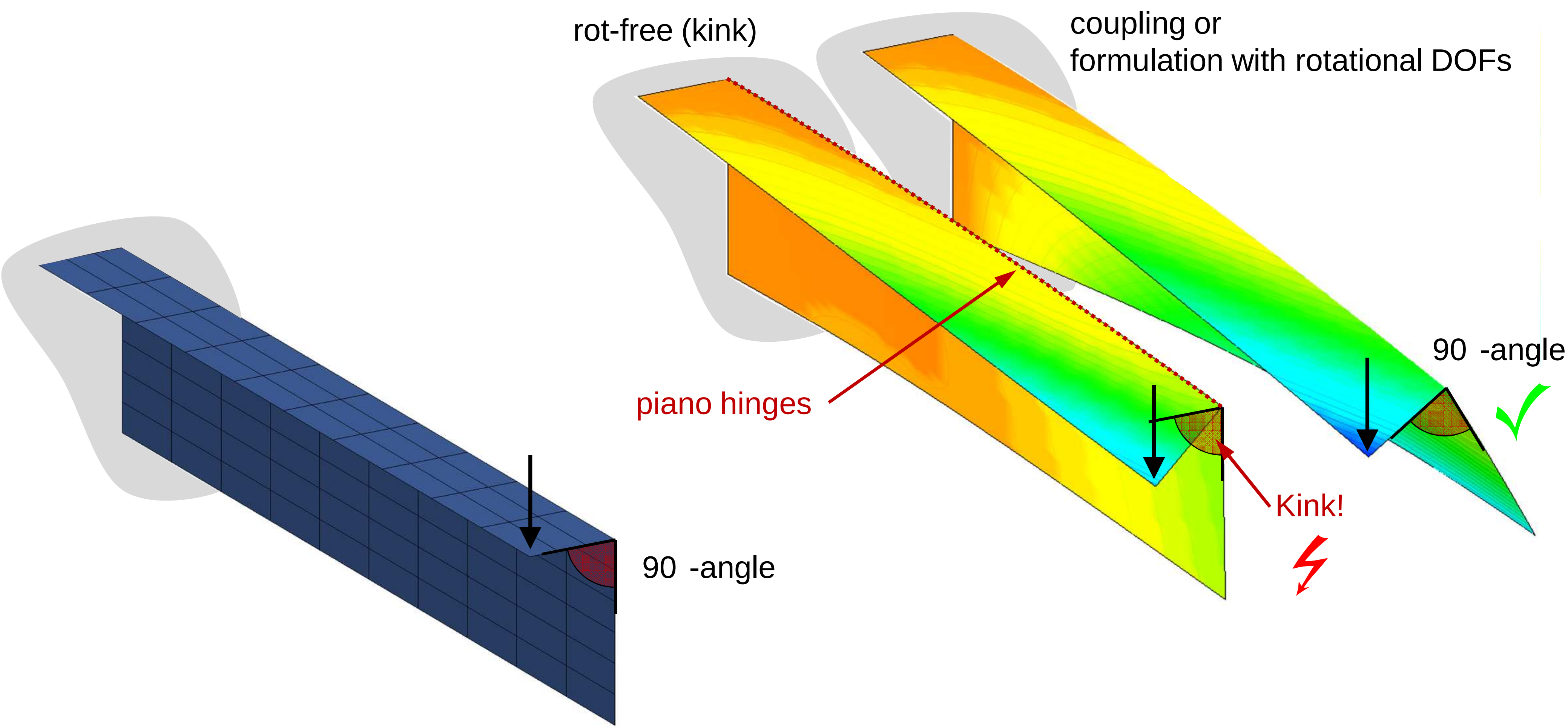
Two patches
(rot-free, coupled)

Two patches
(rot-free, kink)



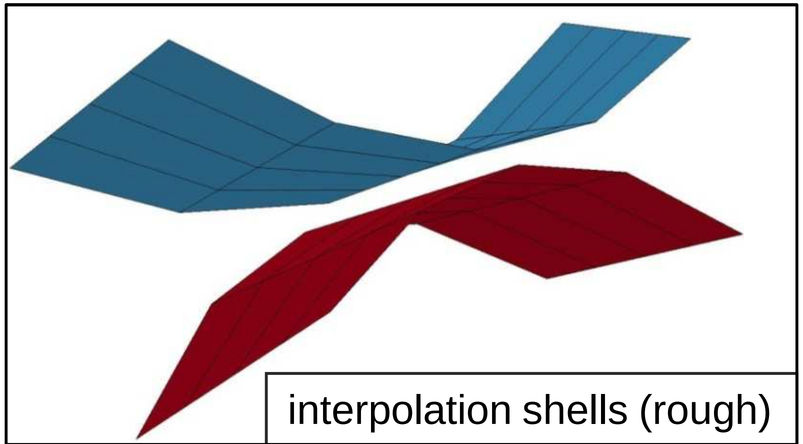
Shell formulations – multi-patch coupling (rotation free formulation)

- keep angle at discontinuities (internal C^0-lines)
  - no transmission of bending moments with rot-free formulation → kinks
  - add rotational DOFS at interior control points (FORM=4)

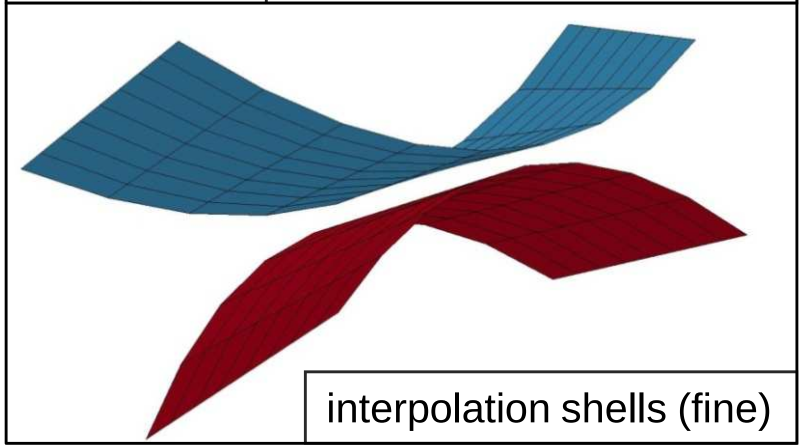


Contact with NURBS

- all (penalty based) contacts available
  - use interpolation shell elements (lin. Quads)
  - loss of smooth (contact) surface representation
    → may lead to problems in tangential sliding

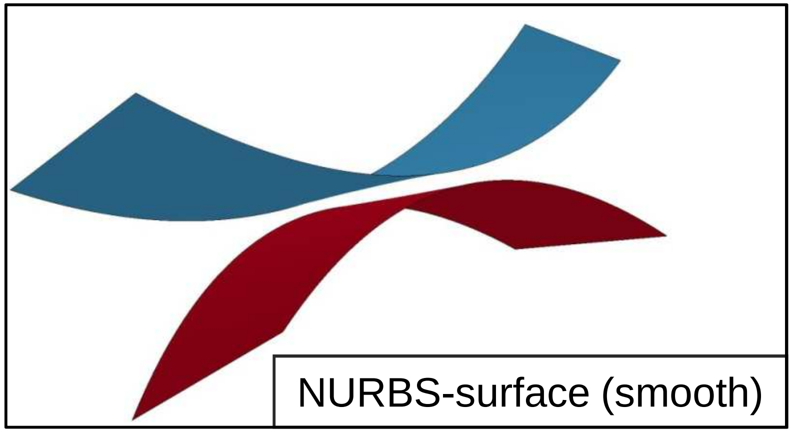


interpolation shells (rough)



interpolation shells (fine)

- NURBS contact for (**SMP only**)
  *CONTACT_AUTOMATIC_1WAY_S2S
  - use NURBS representation on master side
  - smooth contact behavior (i.e. tangential sliding)



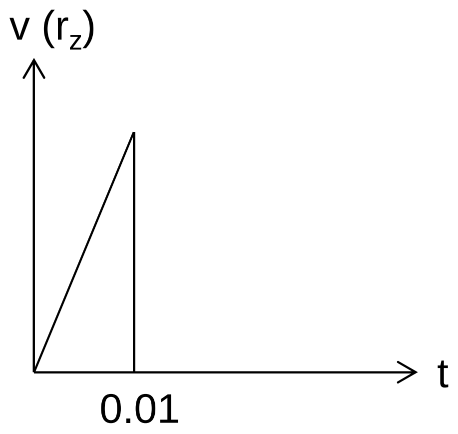
NURBS-surface (smooth)

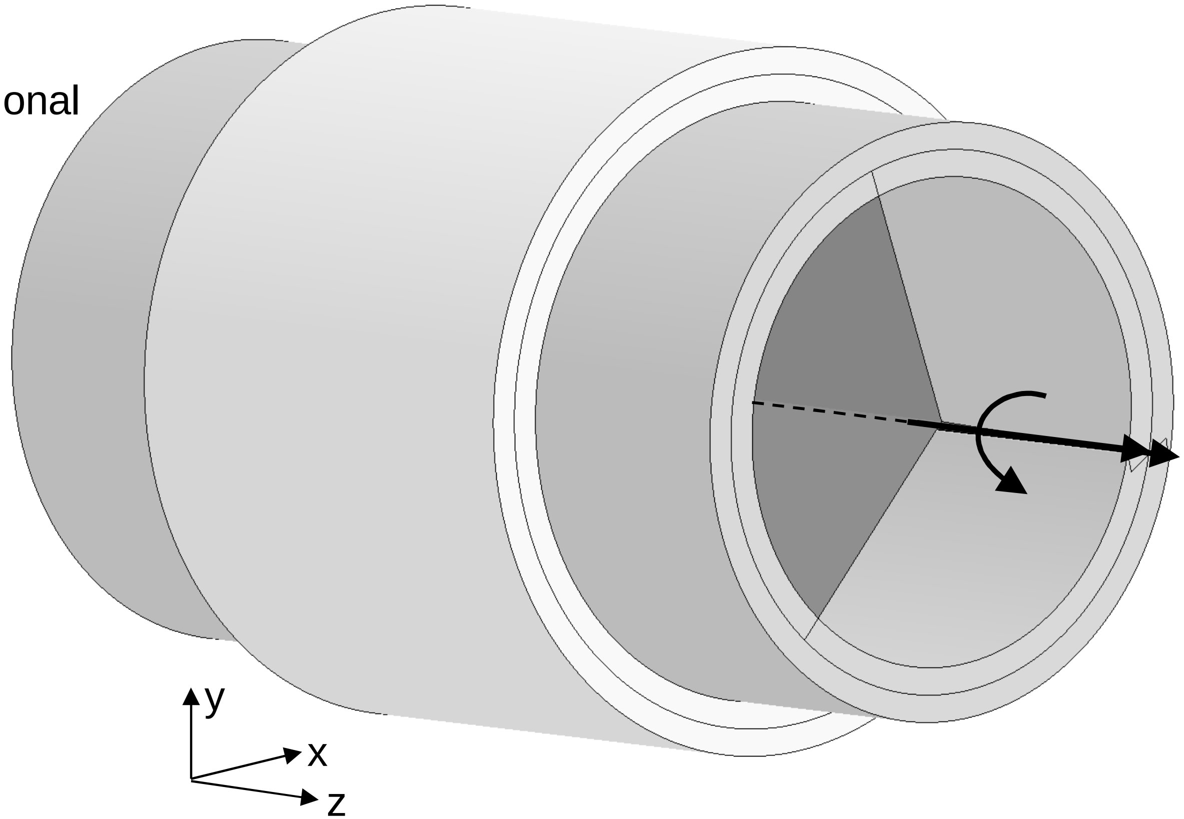
```
*CONTROL_CONTACT
$Optional 6        -2        -3        -4        -5        -6        -7        -8
$---SHLEDG----PSTIFF----ITHCNT----TDCNOF-----ftall---------6----shltrw----igactc
                                                                                1
```

Contact with NURBS – sliding example

- comparison *Contact_Automatic_One_Way_Surface_To_Surface
  - IGACTC=0: master surface defined by interpolation elements (faceted)
  - IGACTC=1: master surface defined by NURBS surface (smooth)
  - _SMOOTH: smooth curve-fitted surface to represent master surface

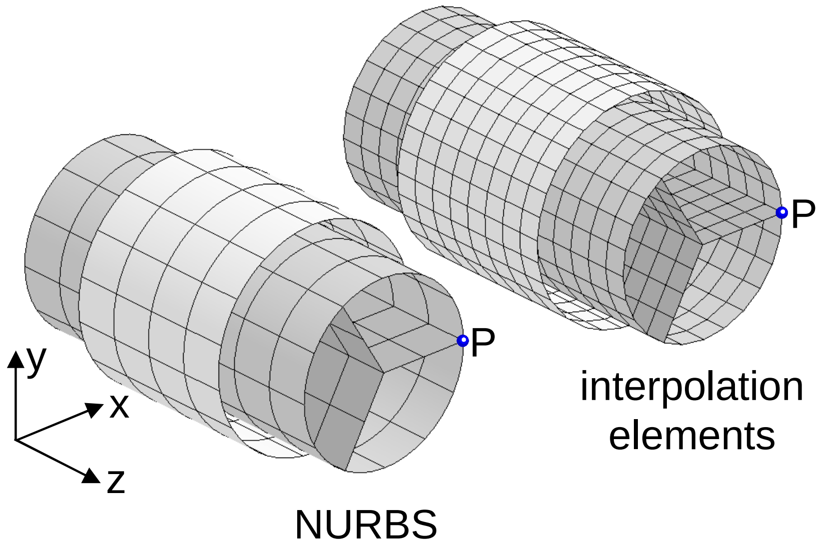
- Example: Tube-In-Tube (both elastic bodies – properties of steel)
  - frictionless sliding
  - outer tube is fixed
  - inner tube is free to move
  - inner tube with initial rotational velocity (death: t=0.01)
  - termination time: t=0.1



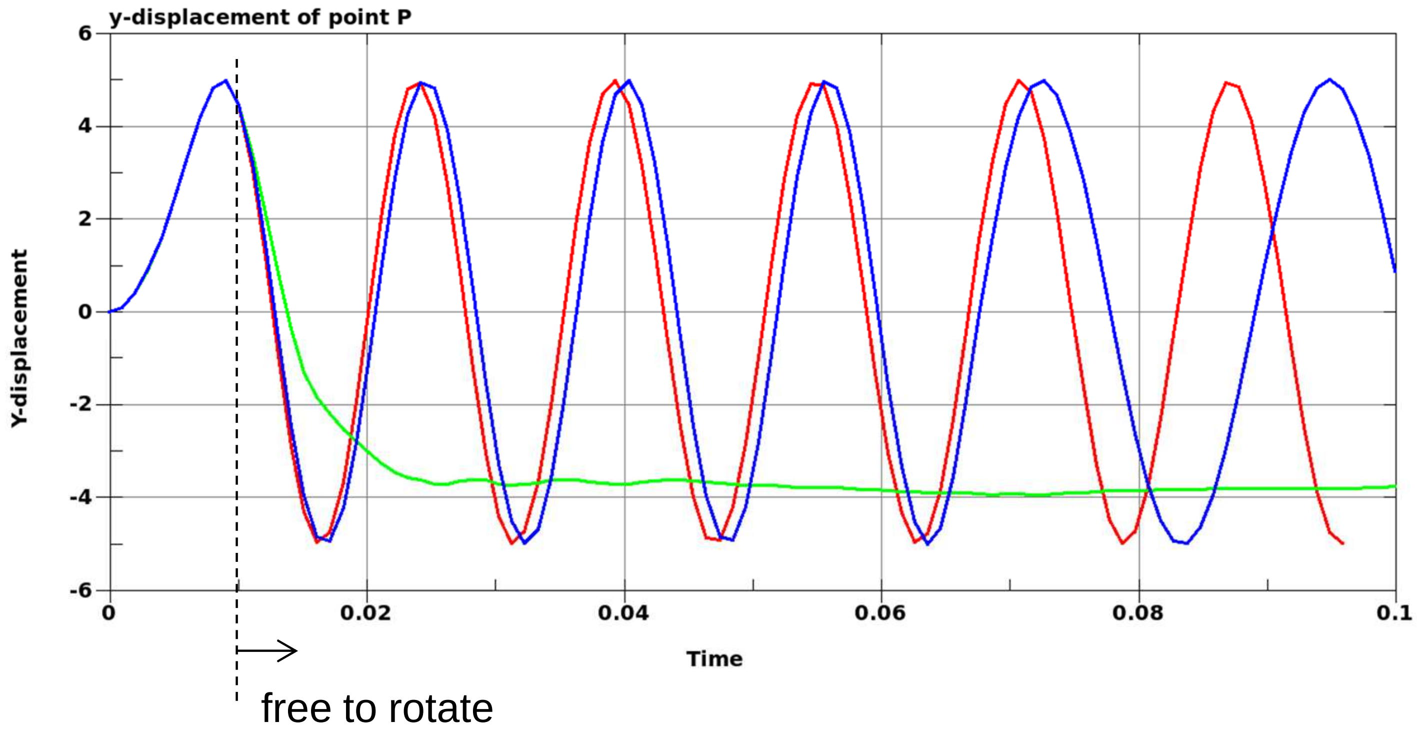


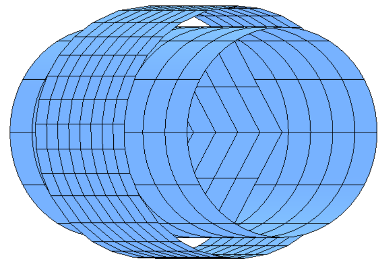
Contact with NURBS – sliding example

- discretization
  - inner tube: 8x12 bi-quadratic NURBS element
  - outer tube: 4x16 bi-quadratic NURBS element
  - 2x2 bi-linear interpolation shells per NURBS element



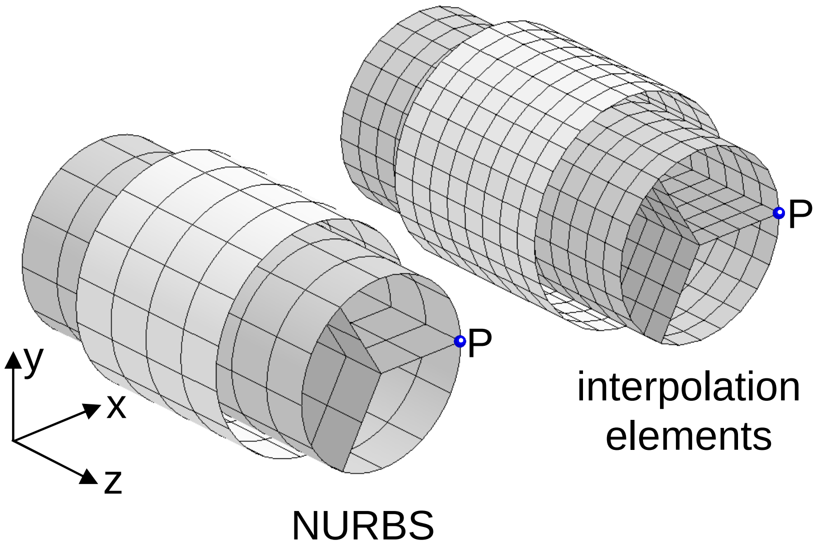
- Version A
  - inner tube is master
  - outer tube is slave



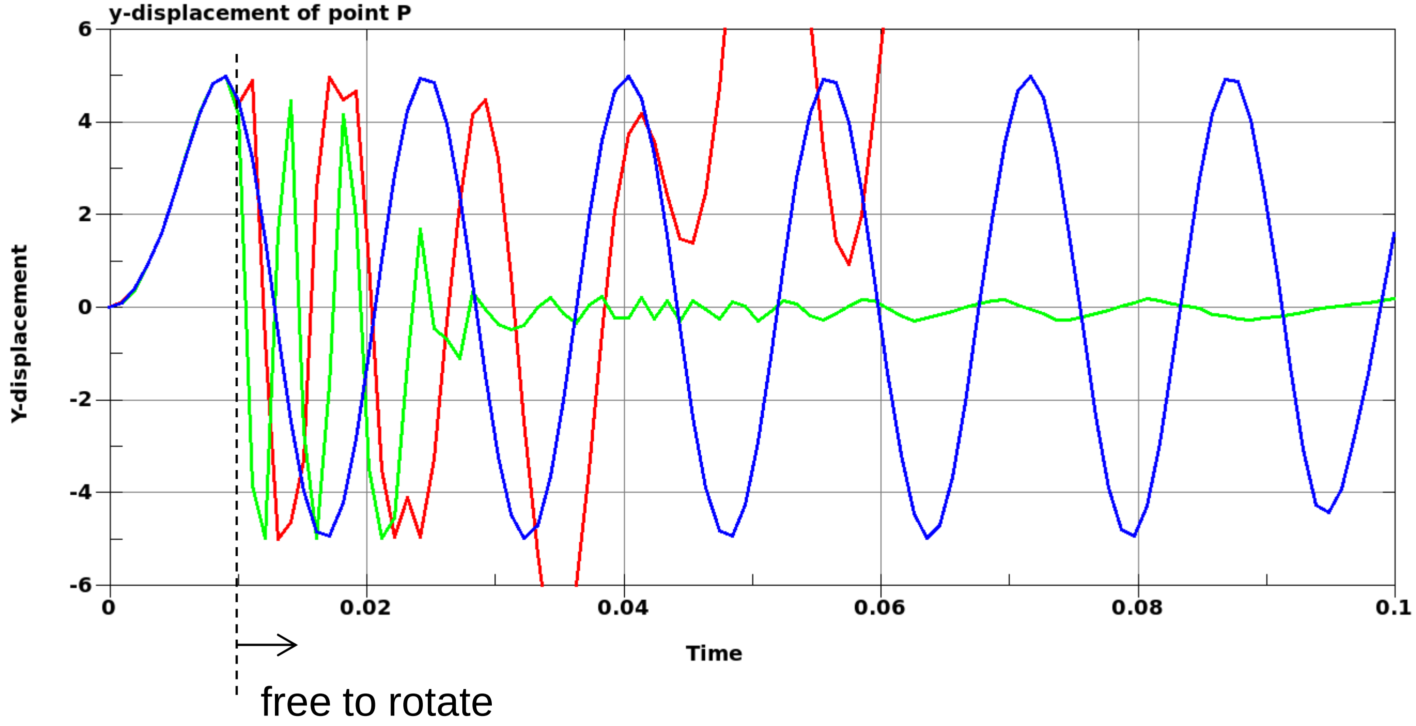


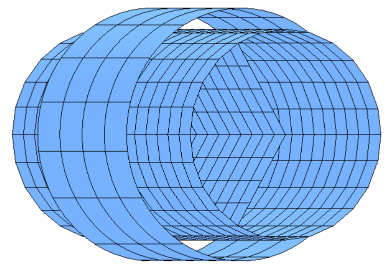
Contact with NURBS – sliding example

- discretization
  - inner tube: 8x12 bi-quadratic NURBS element
  - outer tube: 4x16 bi-quadratic NURBS element
  - 2x2 bi-linear interpolation shells per NURBS element



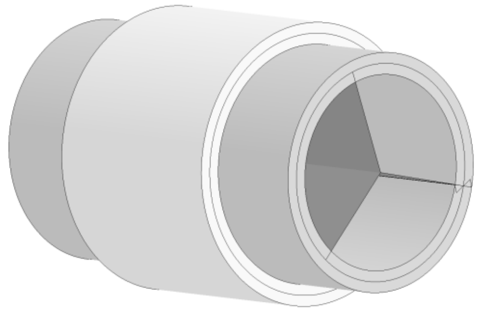
- Version B
  - inner tube is slave
  - outer tube is master





Contact with NURBS – sliding example

- computational time



	Version A		Version B		
	CPU time	#of cycles	CPU time	#of cycles	
IGACTC=0	3 min 32 sec	346535	3 min 18 sec	346484	SMP (single) ncpu=1
IGACTC=1	9 min 27 sec	346537	7 min 55 sec	346536	SMP (single) ncpu=1
_SMOOTH	8 min 43 sec	346484	6 min 10 sec (loss of contact at around t=0.03)	349561	MPP (single) -np 1

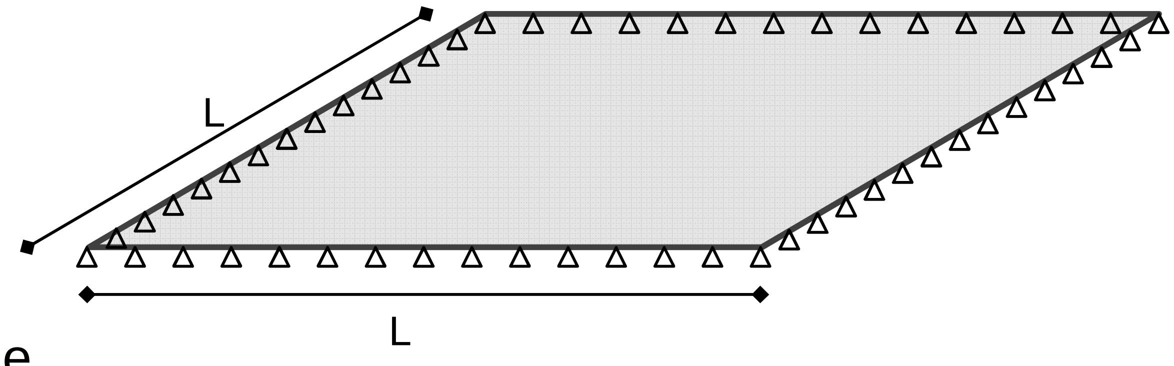
- NURBS contact (IGACTC=1)
  - insensitive to Slave-Master-Pair definition
  - more expensive than standard contact (IGACTC=0) but comparable with _SMOOTH option
  - NOTE: comparison is not representative (SMP vs. MPP), _SMOOTH option and Version B looses contact around t=0.03 (30% of computation time)

Current status - summary

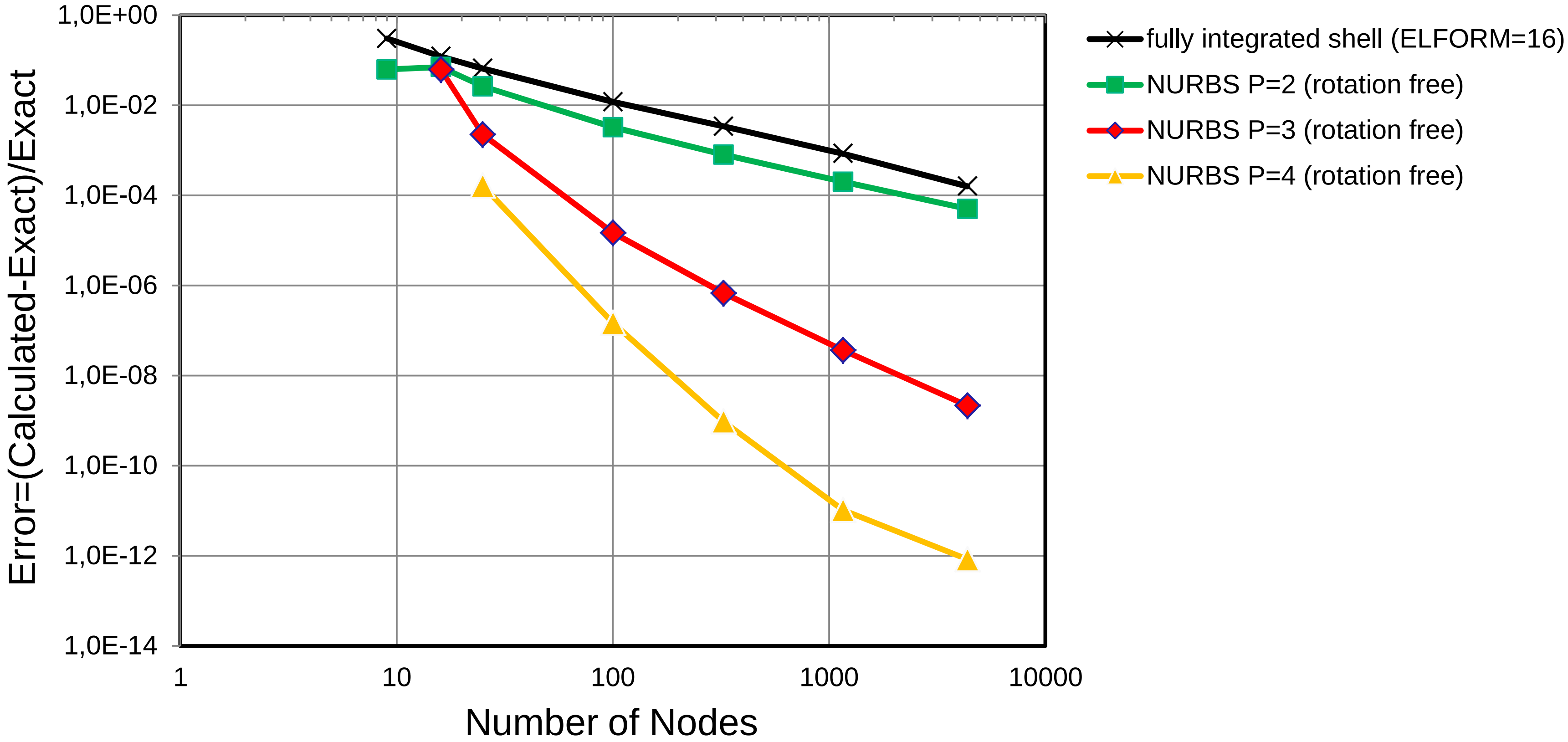
- Keyword: *ELEMENT_SHELL_NURBS_PATCH
  - definition of NURBS-surfaces
  - shell formulations with/without rotational DOFs and hybrid shell
  - NURBS elements run in MPP with good scaling
  - NURBS contact (*CONTACT_AUTOMATIC_ONE_WAY_SURFACE_TO_SURFACE, IGACTC=1, SMP only)
- Pre- and Postprocessing
  - work in progress for LS-PrePost … current status (lspp4.1beta)
    → visualization of 2D-NURBS-Patches
    → import IGES-format and construct *ELEMENT_SHELL_NURBS_PATCH
    → modification of 2D-NURBS geometry
    → … much more to come!
- Postprocessing and boundary conditions (i.e. contact) mainly with
  - interpolation nodes (automatically created)
  - interpolation elements (automatically created)
- Analysis capabilities
  - 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)

Example – Vibration of a square plate

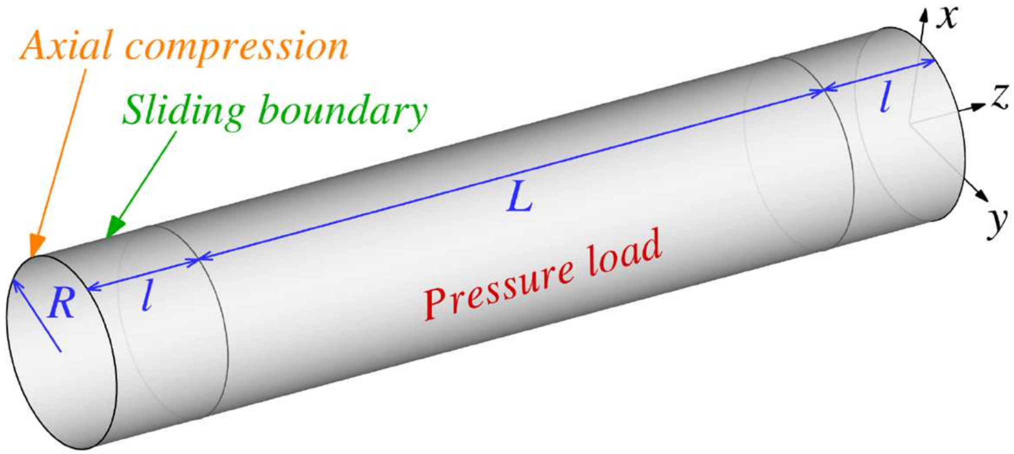
- eigenvalue analysis
  - simply supported square plate
  - linear elastic material
  - consistent mass matrix
  - rotation free shell formulation



- accuracy of first eigenvalue



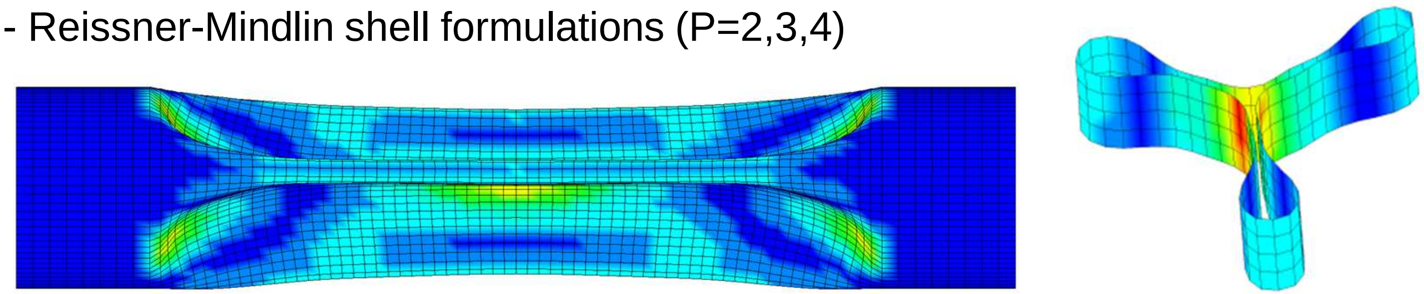
Example – dynamic buckling



- nonlinear dynamic buckling
  - isotropic elastic-plastic material
  - linear plastic hardening
  - NURBS (P=2,3,4)
  - Reissner-Mindlin formulation

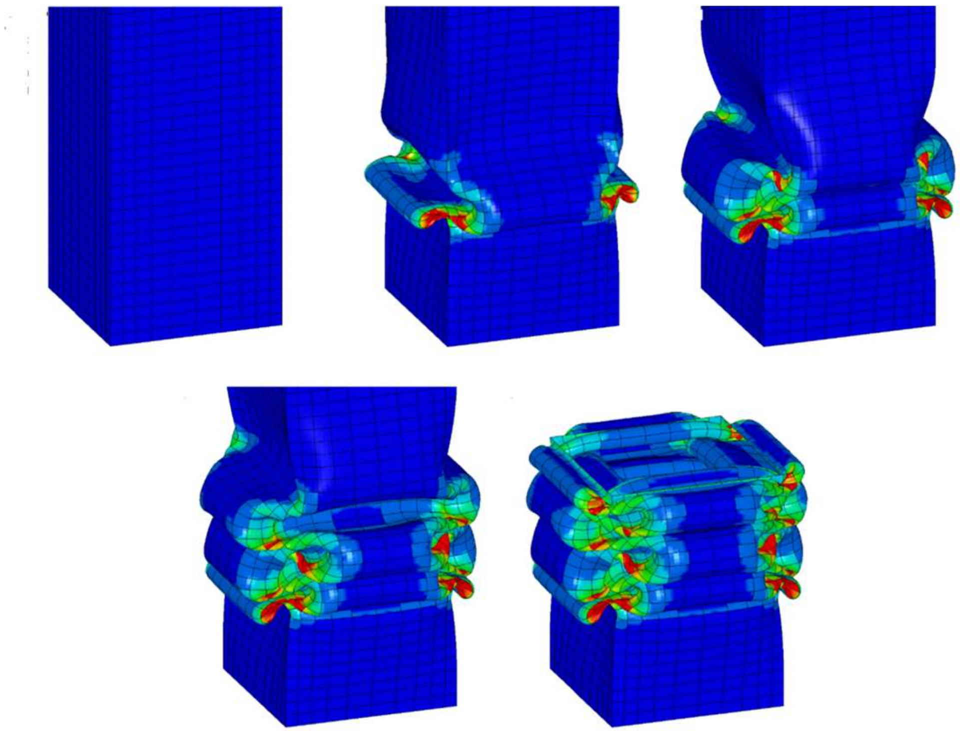
Experimental results from Kyriakides S. Kyriakides, Personal communication, 2008

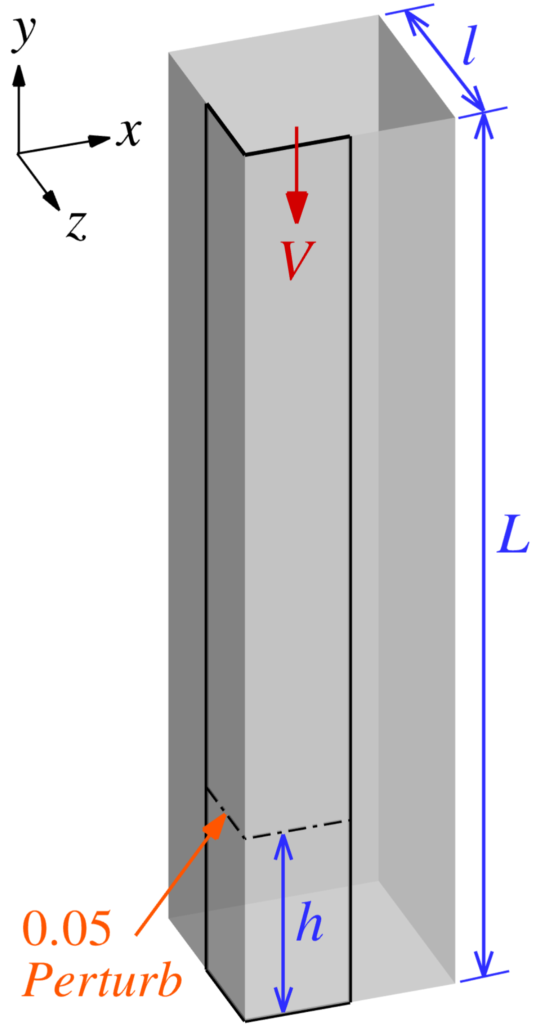
- good agreement of
  - buckling load & buckling pattern
  - Reissner-Mindlin shell formulations (P=2,3,4)



Example - buckling

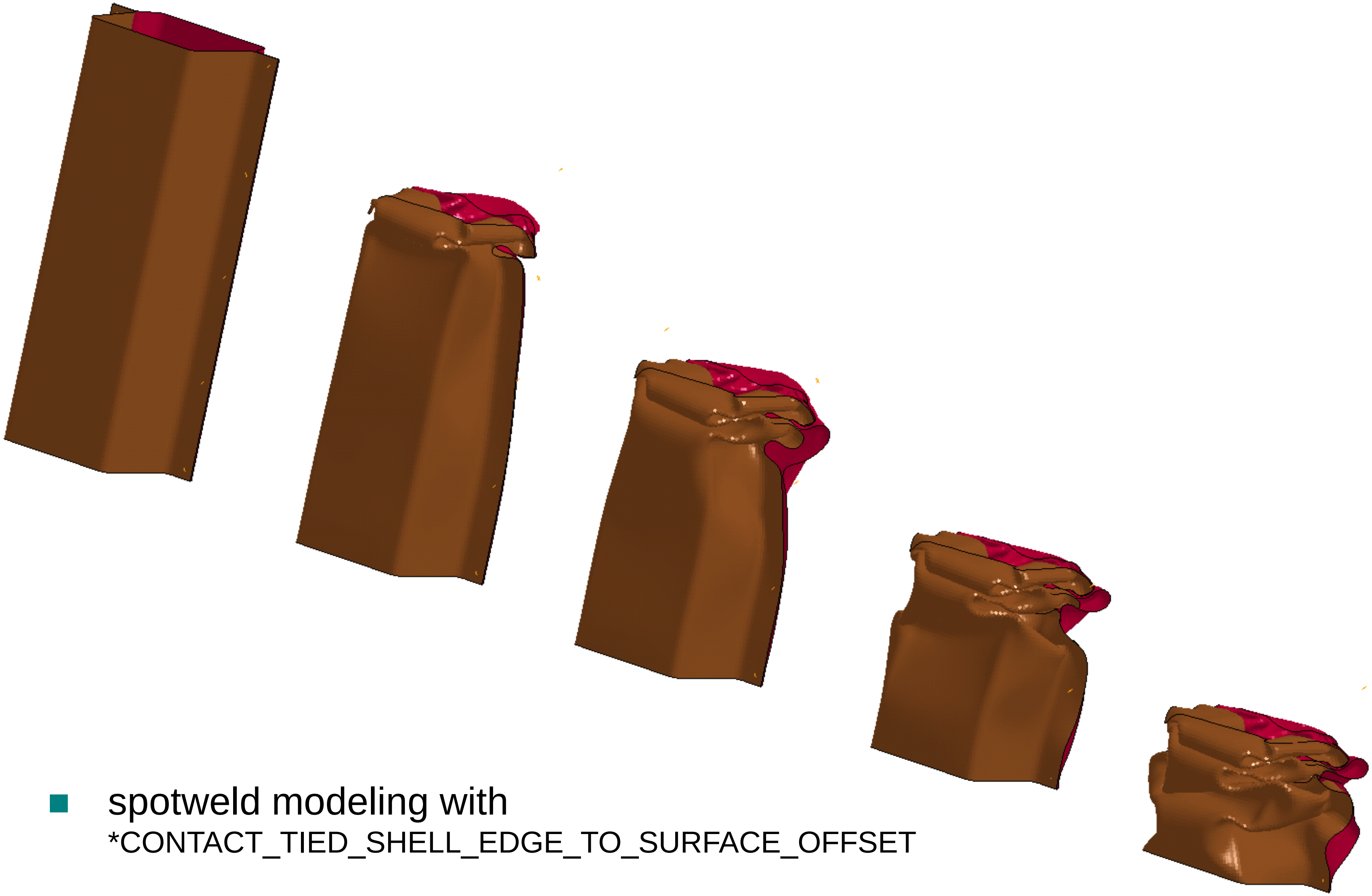
- standard benchmark for automobile crashworthiness
- quarter symmetry to reduce cost
- perturbation to initiate buckling mode
- J_2 plasticity with linear isotropic hardening
- mesh:
  - 640 quartic (P=4) elements
  - 1156 control points
  - 3 integration points through thickness





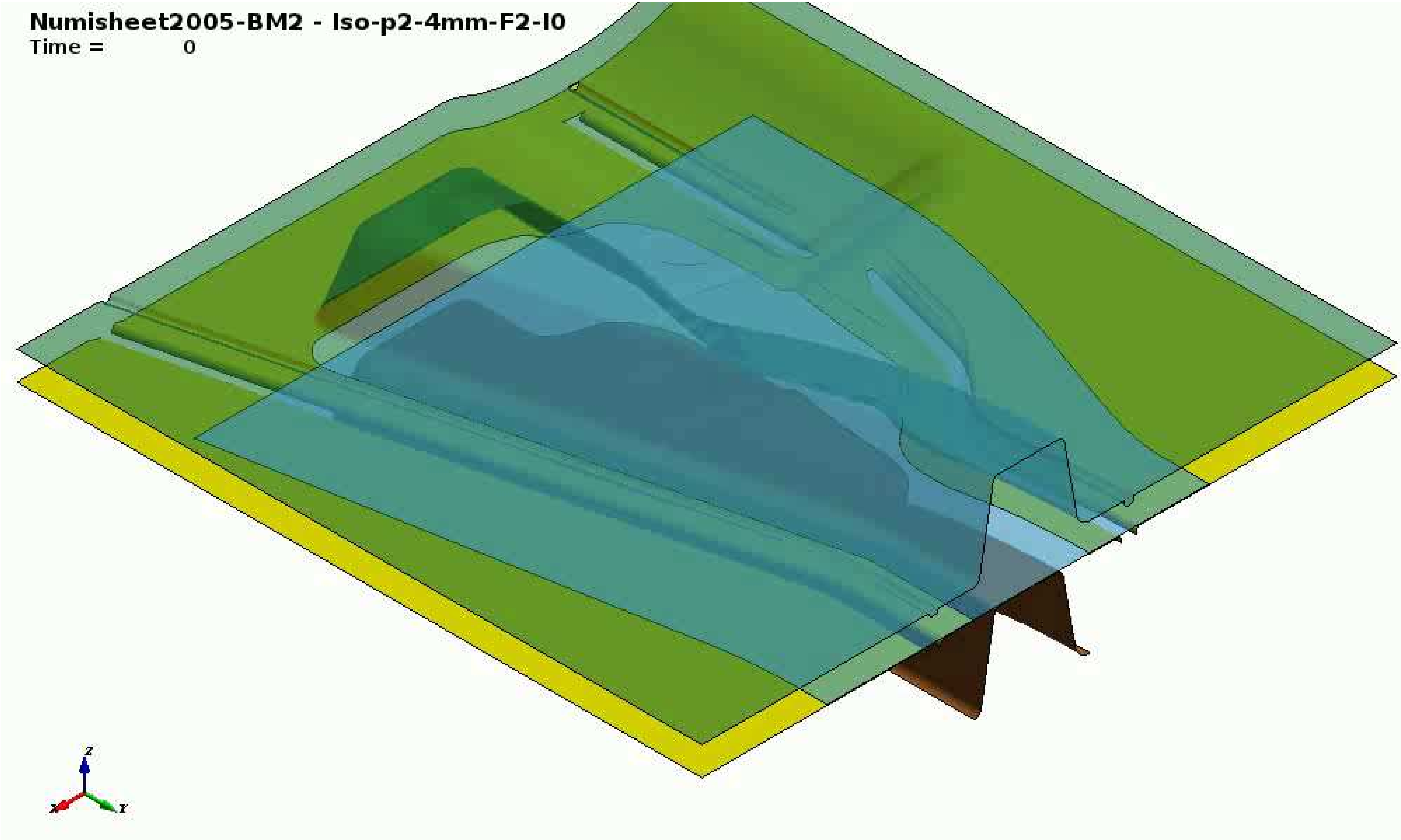
D.J. Benson

Example – column crush

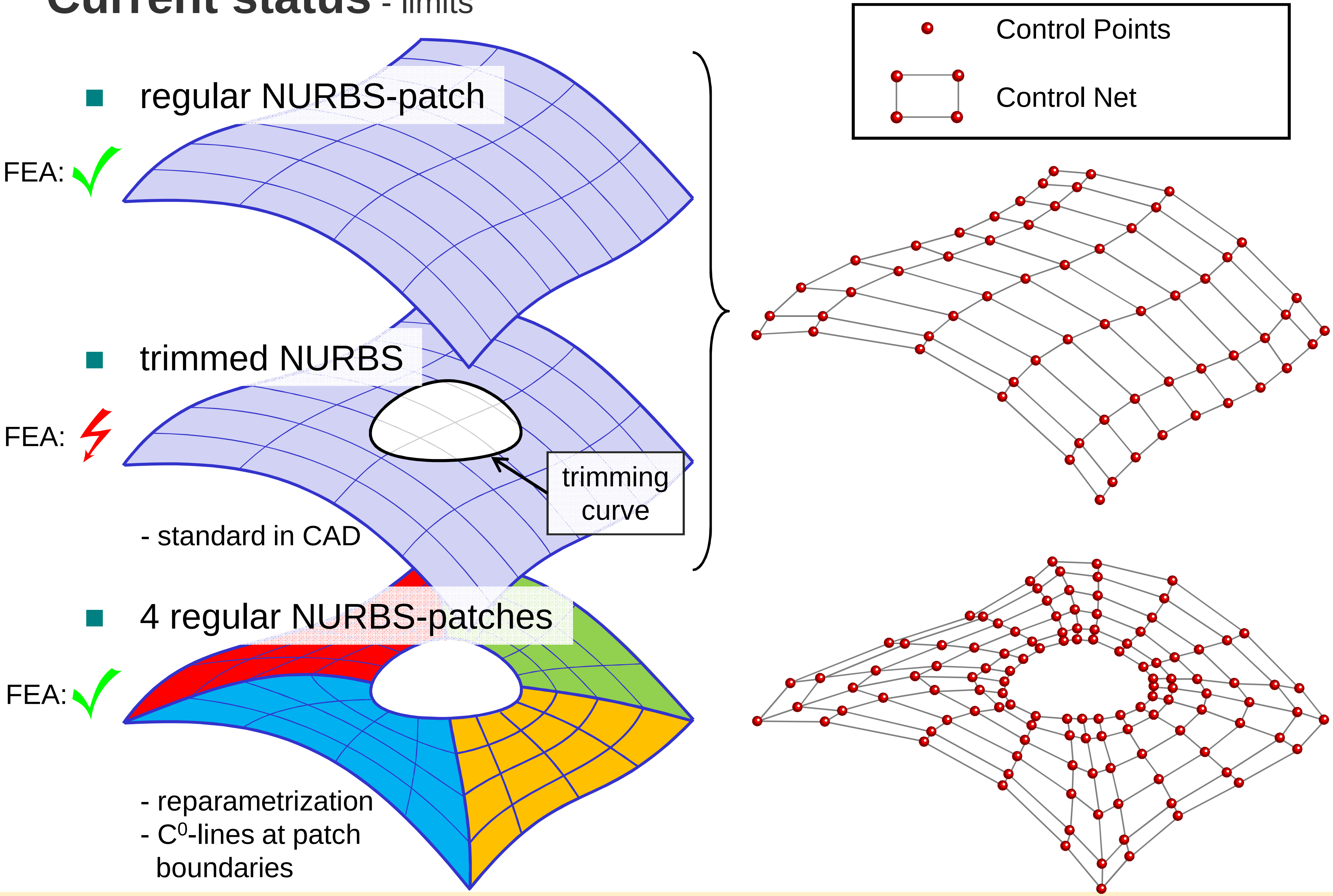


- spotweld modeling with
  *CONTACT_TIED_SHELL_EDGE_TO_SURFACE_OFFSET

Example - forming



Current status - limits



Summary

- NURBS-based elements run stable and scale good in MPP
- higher order accurate isogeometric analysis may be cost competitive
- code optimization necessary to make it faster
- NURBS contact works well
- model creation with standard NURBS is currently the bottleneck!

ToDo-List

- perform a lot more studies in different fields → experience
- motivate customers (and researchers) to "play" with these elements
- further implementation
  - post-processing directly with NURBS (partially available already)
  - mass scaling
  - NURBS contact implementation for MPP
  - merging of non-matching NURBS patches
  - trimmed NURBS (research currently done at UCSD)
  - make pre- and post-processing more user-friendly and faster
  - introduce 3D NURBS elements
  - … much more

Thank you!