Global parameterization of trimmed NURBS based CAD geometries

Mesh deformation based on CAD model deformation

Martin Siggel, Tobias Stollenwerk

German Aerospace Center (DLR)

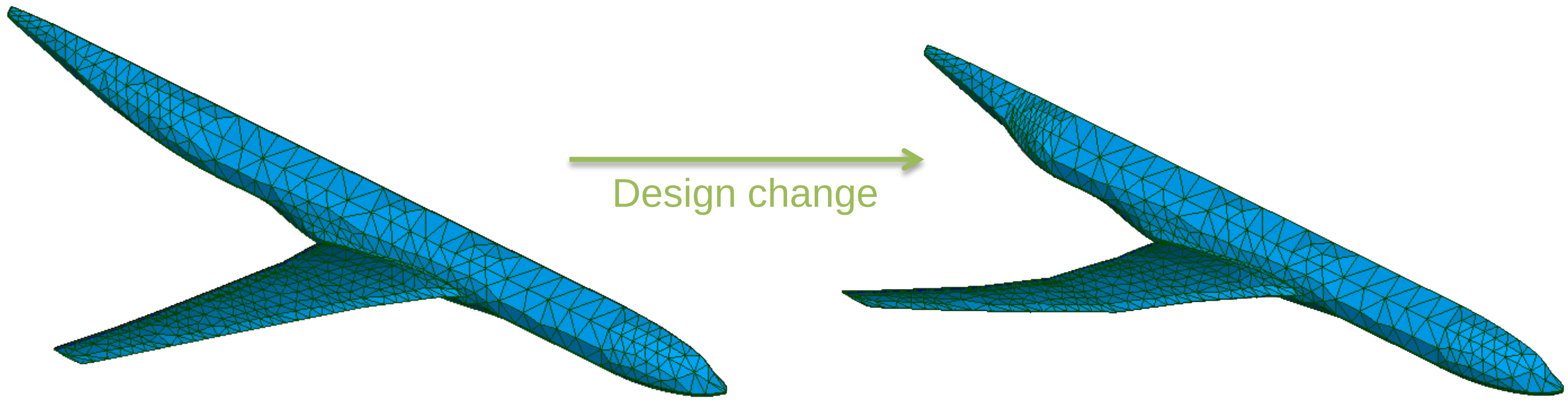
WCCM XII, Seoul

25-Jul-2016

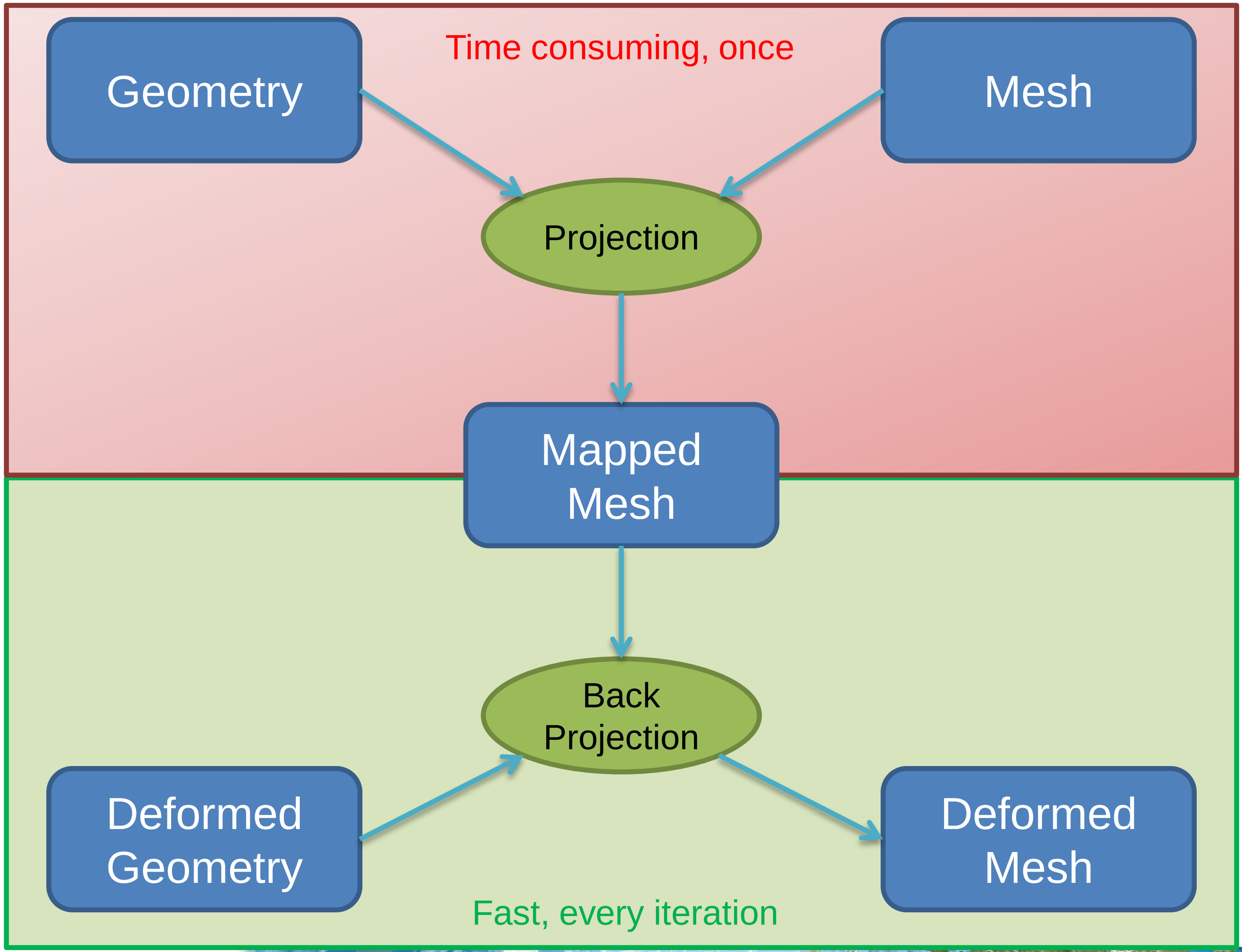
DLR

Motivation
Geometry based mesh deformation

- Fully automatic mesh creation is hard!
- Idea: reuse manually created mesh even for different designs
- Use for gradient based optimization to determine mesh gradient w.r.t design parameters

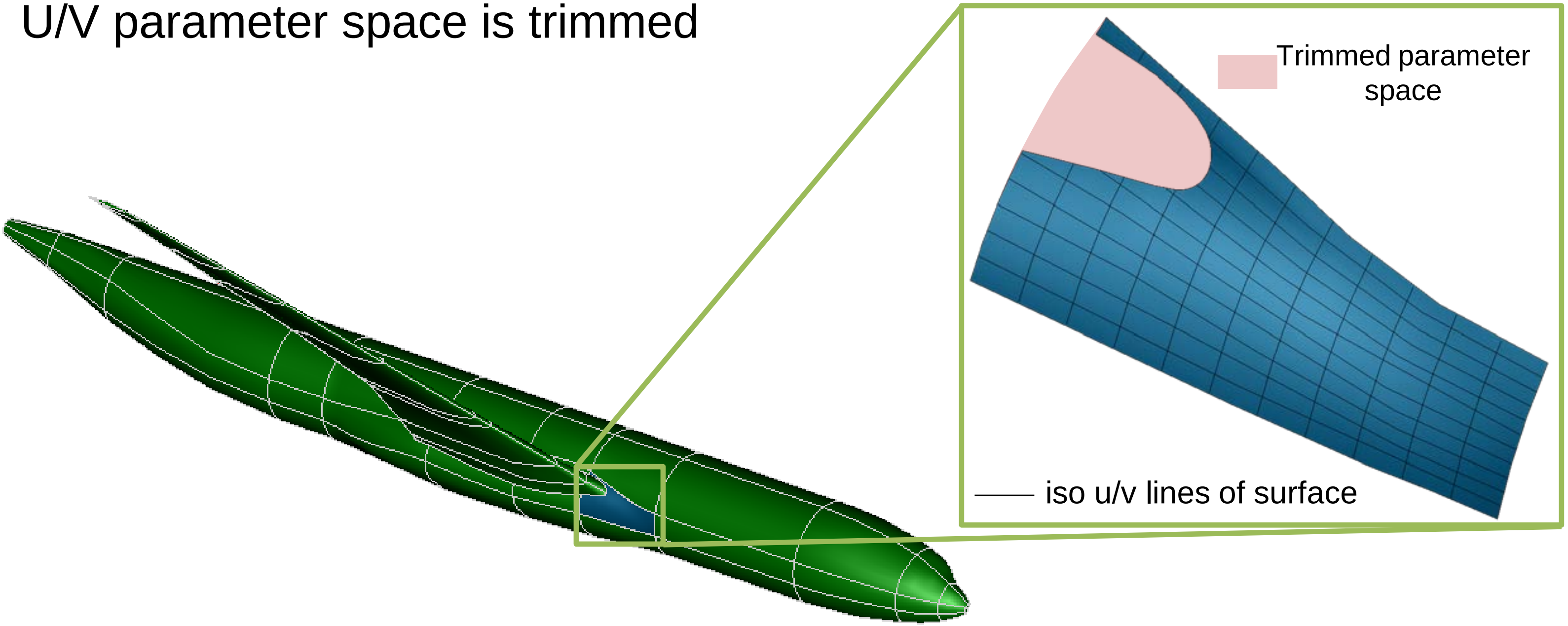


Geometry based mesh deformation
General idea



Trimmed NURBS based geometries

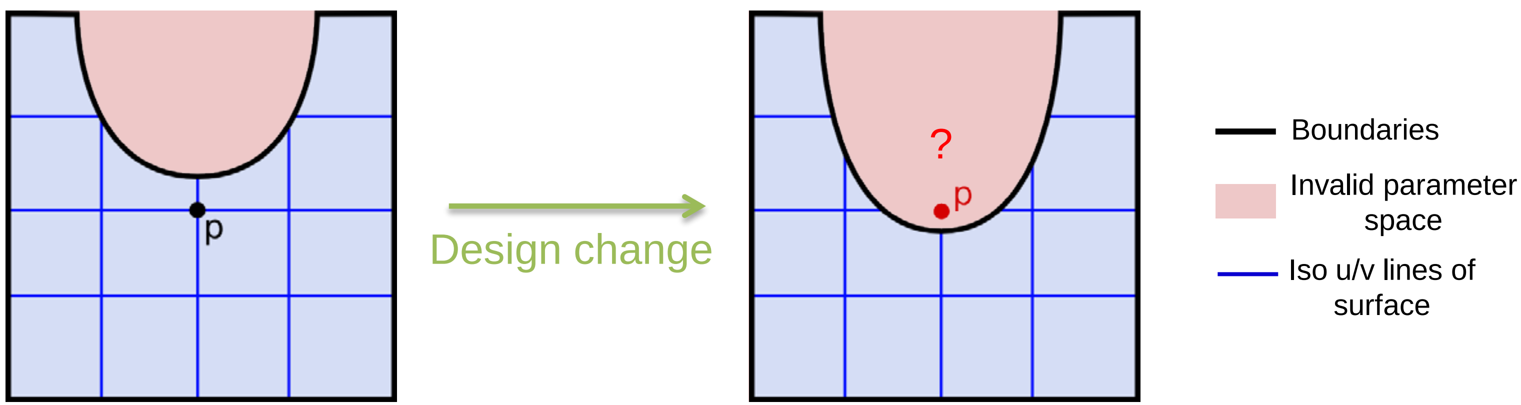
- Consists of many panels
- Each panel is a trimmed NURBS
- U/V parameter space is trimmed



The trimmed NURBS parameterization problem

- Trimmed NURBS:
  - Parameter space of surface trimmed by boundary curves

$p_{uv} = (0.5, 0.5)$



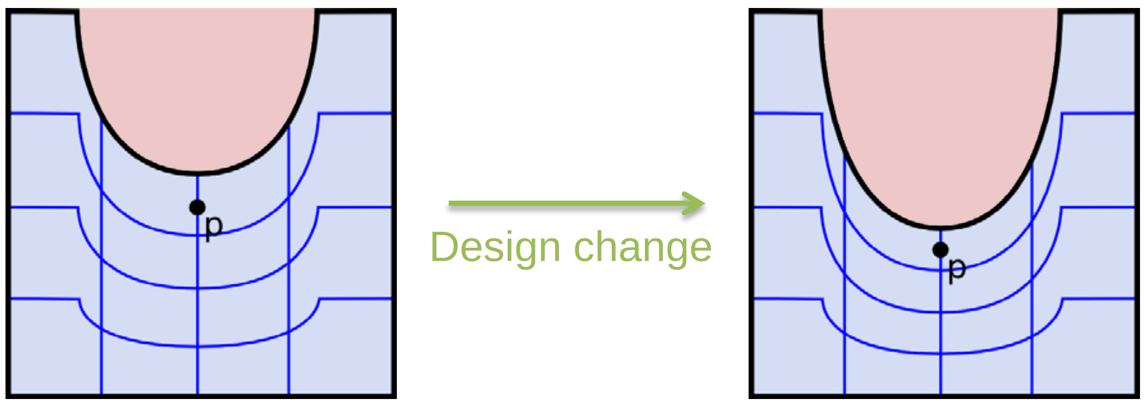
  - Mesh points get lost / are outside panel after design change

➢u/v surface coordinates no good choice for mapping

Solution: reparameterization of the parameter space

- Create 2D reparameterization surface that respects boundaries

$p_{u'v'} = (0.5, 0.125)$

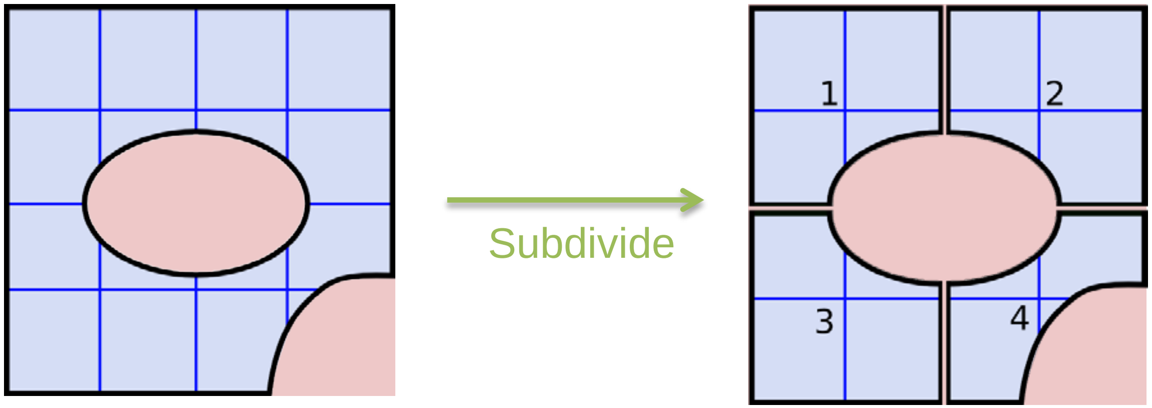


— iso u'/v' lines of reparameterization surface

- No loss of mesh points anymore
- Next problem: what to do with holes or complicated boundaries?

The trimmed NURBS parameterization problem

- Panels with complicated boundaries can not be reparameterized with one surface!



- Solution: subdivision of the panel into multiple sub-surfaces

➢ Reparameterize each sub-surface

Projection / Back projection

- **Projection:**

$$\vec{P} \in \mathbb{R}^3 \rightarrow i, j, u', v'$$

- **Back projection**:

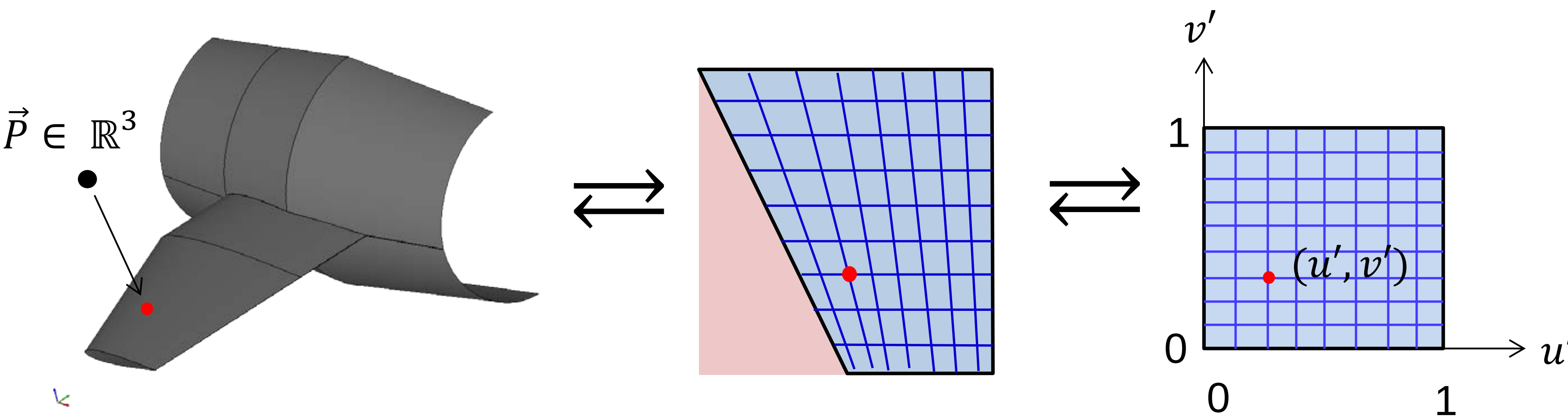
$$i, j, u', v' \rightarrow \vec{P} \in \mathbb{R}^3$$

with:

i : panel index

j : sub-surface index

$u', v' \in [0,1]$



3D reparametrized 2D surface 2D parameter space

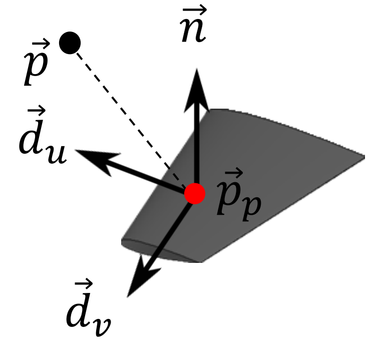
Improvement: Exact back projection

- Motivation:
  - Geometry changes are very small for finite differences
  - Mesh points may not exactly lie on the original geometry

- Requirement: Invariant geometries should result in <u>exactly</u> the same mesh

- Solution:
  1. Project point $\vec{p}$ onto surface
  2. Create local coordinate system (CS) from $\vec{d}_u,\ \vec{d}_v, \vec{n} = \overrightarrow{d_u} \times \overrightarrow{d_v}$
  3. Store also the deviation of projected point $\overrightarrow{p_p}$ in the local CS

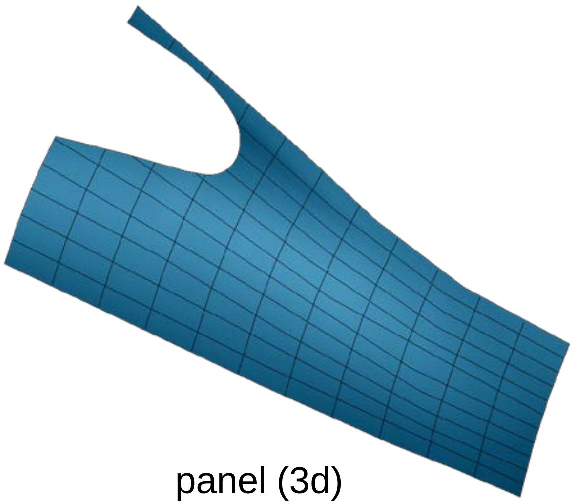
$$\vec{p} - \overrightarrow{p_p} = dn \cdot \vec{n} + du \cdot \overrightarrow{d_u} + dv \cdot \overrightarrow{d_v}$$



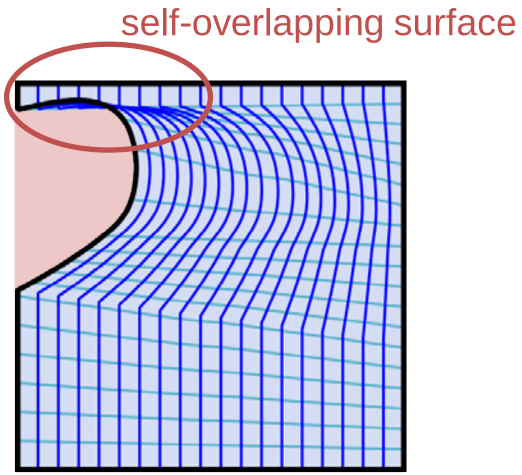
- (Back)Projection: $\vec{P} \in \mathbb{R}^3 \leftrightarrow i, j, u', v', dn, du, dv$

Uniqueness of projection

- Reparameterization surfaces can self-overlap



panel (3d)



reparameterization surface (2d)

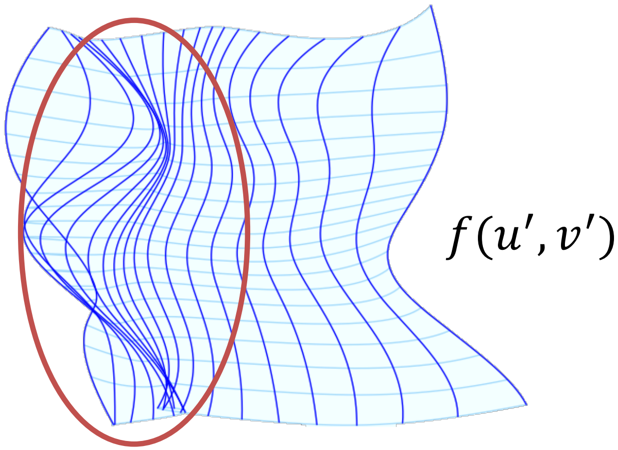
- Issue: Projection is not unique → multiple solutions !

- Resulting mesh might be invalid

Uniqueness of projection
Invertibility of reparameterization surfaces

- Reparameterization surface $f(u',v') \rightarrow (u,v)$ must be invertible
  - there must exists inverse function f', with

$$f'(u,v) \rightarrow (u',v')\,,\ u',v' \in [0,1],\ u,v \text{ inside trimmed parameter space}$$



- If invertible, projection is unique
- How to check existence of $f'(u,v)$?

Uniqueness of projection
Invertibility criterion

- Jacobi determinant (i.e. Z-component of normal vector) must not be negative, i.e.

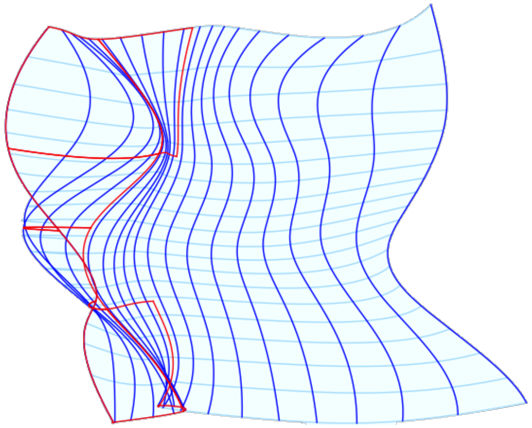
$$\det J(u',v') \stackrel{\text{def}}{=} \left[\frac{\partial}{\partial u'} f(u',v') \times \frac{\partial}{\partial v'} f(u',v')\right]_z \geq 0, \forall\, u',v' \in [0,1]$$

- Jacobi determinant of Bezier patch is 1d Bezier surface:
    - control points T_{pq} can be computed according to *, eq. (11)
    - Convex hull property: $\det J(u',v') \geq 0$, if $T_{pq} > 0$

➢ Split reparameterization surface f into Bezier patches

➢ Inspect control points for positivity

- But: false positives possible!



Non-invertible Bezier patches

*Lin, Hongwei et al. 2007: Generating strictly non-self-overlapping structured quadrilateral grids

Implementation details

- Written in C++ as a library

- Library offers functions to
  - Perform projection and back-projection of points
  - Import CAD models (IGES, Step, BREP)
  - Check invertibility of each sub-surface
  - Distribute the geometry to each node of a cluster
  - Compare the topology of two CAD models

- Uses the OpenCASCADE Technology* CAD library

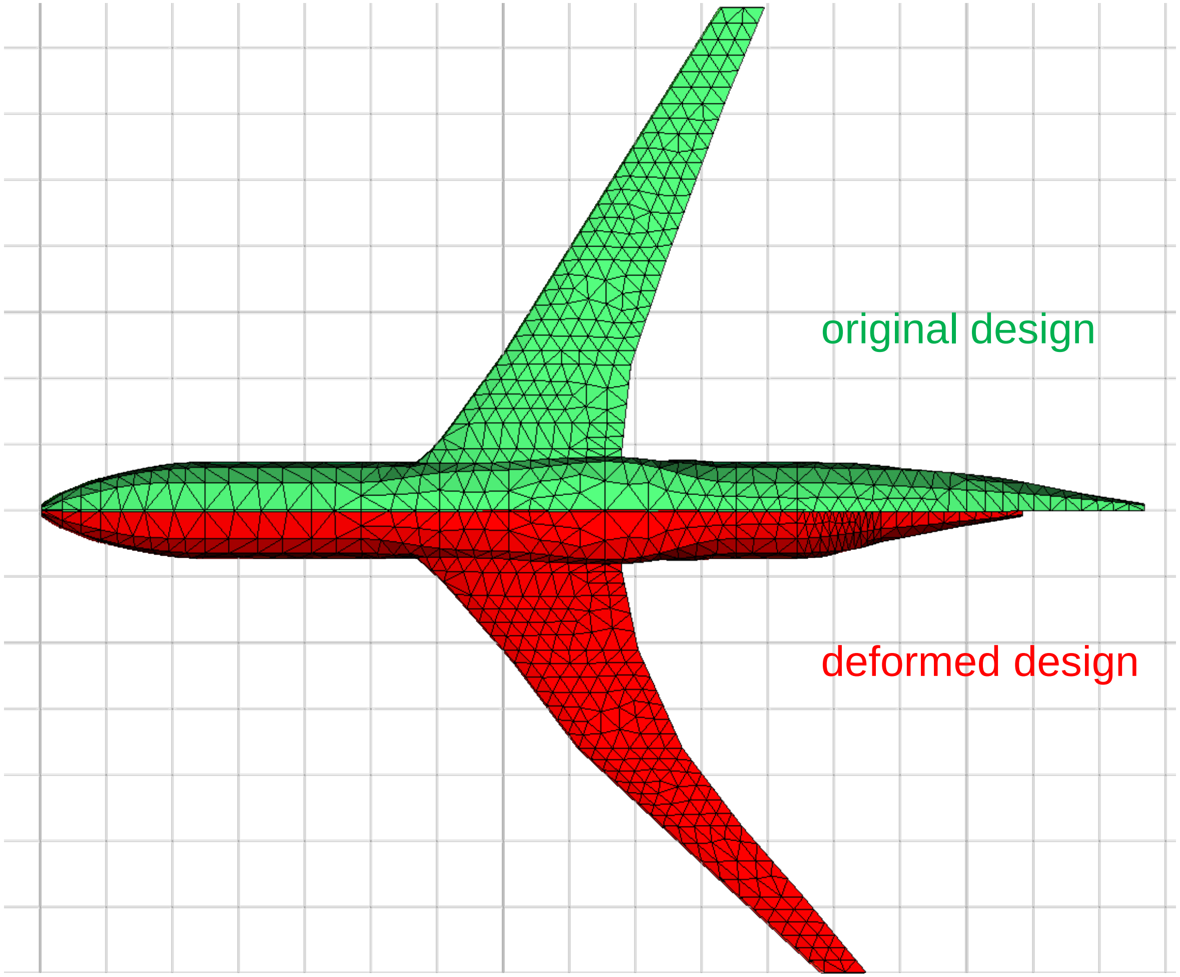
*http://www.opencascade.com

Results

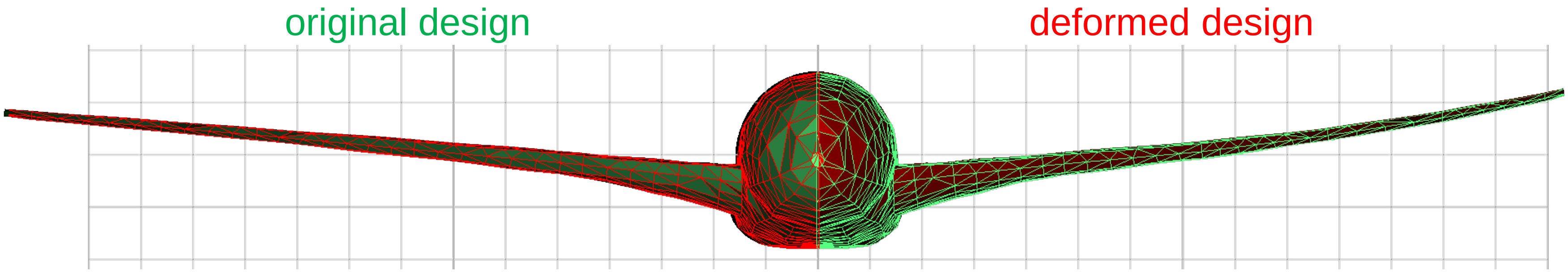
- Algorithm currently used at Airbus D&S for FEM based structural analysis
- Currently used in DLR for CFD simulations with structured meshes in the DLR internal project VicToria*:
  - Adjoint CFD solver TAU requires mesh gradients
    (i.e. how does the mesh change with some design parameter change)
  - This method suitable to compute the gradients
- Method not suitable for large design changes:
  - Topology of panels and boundaries must not change!

*http://www.dlr.de/as/en/desktopdefault.aspx/tabid-11460/20078_read-47033/

Results



Results



Summary + Outlook

Questions

martin.siggel@dlr.de
tobias.stollenwerk@dlr.de