Parameterization of trimmed NURBS geometries for mesh deformation

Martin Siggel, Tobias Stollenwerk
German Aerospace Center (DLR)

Workshop
Cologne University + DLR
04-Oct-2016
Motivation
Gradient based design optimization

Optimization

- Design variables $\vec{p}$
- Geometry Generation $g(\vec{p})$
- Mesh Generation $m(g)$
- CFD Simulation
- Optimization Objective $f(\vec{p})$
Motivation
Geometry based mesh deformation

- Problem: fully automatic mesh creation is hard:
  - Structured meshes require fine tuning
  - Number of mesh points varies → Finite differences not possible

- Solution: deform manually created mesh on design changes

- Can be used for gradient based optimization to determine mesh gradient w.r.t design parameters
Geometry based mesh deformation

General idea

- **Geometry**
- **Mesh**
- **Projection**
- **Mapped Mesh**
- **Back Projection**
- **Deformed Geometry**
- **Deformed Mesh**

*Time consuming, once*

*Fast, every iteration*
Trimmed NURBS based geometries

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

Trimmed parameter space

iso u/v lines of surface
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

\[ p_{u'v'} = (0.5, 0.125) \]

- 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
- \( u', v' \in [0,1] \)

\[ \vec{P} \in \mathbb{R}^3 \]

3D reparametrized 2D surface

\[ \text{reparametrized 2D surface} \]

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

- **But:** Invariant geometries should result in **exactly** the same mesh!

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

\[
\mathbf{p} - \mathbf{p}_p = dn \cdot \mathbf{n} + du \cdot \mathbf{d}_u + dv \cdot \mathbf{d}_v
\]

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

- Reparameterization surfaces can self-overlap

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

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

  ➢ Split reparameterization surface \( f \) into Bezier patches
  
  ➢ Inspect control points of Bezier patches for positivity

  • But: False positives possible!

*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 (Parallelization)
  • 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!

Results

original design

deformed design
Results

original design

deformed design
Summary + Outlook

- Presented a method to globally parameterize points on a trimmed NURBS based CAD geometry
  - Re-parameterization of surfaces
  - Surface splitting
  - Check for uniqueness of projection

- The method is already successfully used for large scale aircraft optimization

- In future:
  - Improve automatic splitting into sub-surfaces for different scenarios (hole(s), multiple boundaries …)
Questions

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