CAD-based shape optimisation using a discrete adjoint solver

Wolfram Jahn, Shenren Xu, Dominic Jones, Jens-Dominik Müller

Dept. of Engineering, Queen Mary, University of London

March 28th 2012
Outline

Motivation

Implementation

Results

Summary and future work
Outline

Motivation

Implementation

Results

Summary and future work
Outline

Motivation

Implementation

Results

Summary and future work
Outline

Motivation

Implementation

Results

Summary and future work
Outline

Motivation

Implementation

Results

Summary and future work
Motivation

- Parameterisation is the key to successful optimisation
  - Node-based optimisation interface not well with CAD
  - Full CAD-based approach needs either (1) differentiating the CAD system or (2) solving transformation matrix using finite difference

Figure: Node-based (left) v.s. full CAD-based (right) optimisation.
Motivation

Parameterisation is the key to successful optimisation

- Node-based optimisation interface not well with CAD
- Full CAD-based approach needs either (1) differentiating the CAD system or (2) solving transformation matrix using finite difference

Figure: Node-based (left) v.s. full CAD-based (right) optimisation.

A more flexible CAD-based optimisation is needed
Motivation

▶ Use the control points of the boundary representation (BRep) as design parameters

Figure: Surface with control points (C.P.) viewed in CAD.

▶ Advantages: good interface to CAD, no smoothing/fd

▶ Challenge: imposition of various constraints
Motivation

- Use the control points of the boundary representation (BRep) as design parameters

Figure: Surface with control points (C.P.) viewed in CAD.

- Advantages: good interface to CAD, no smoothing/fd
- Challenge: imposition of various constraints
Motivation

▶ Use the control points of the boundary representation (BRep) as design parameters

Figure: Surface with control points (C.P.) viewed in CAD.

▶ Advantages: good interface to CAD, no smoothing/fd
▶ Challenge: imposition of various constraints
Outline

Motivation

Implementation

Results

Summary and future work
Boundary representation, definition of NURBS

- Surfaces in CAD systems are NURBS (Non-Uniform Rational B-Spline), defined as follows

\[
X_s(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{ij}(u, v) P_{i,j}
\]

\[
R_{ij}(u, v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n} \sum_{l=0}^{m} N_{k,p}(u)N_{l,q}(v)w_{k,l}}
\]

- Position, tangent vectors and curvatures can be computed inexpensively for imposing (nonlinear) continuity constraints
Boundary representation, definition of NURBS

- Surfaces in CAD systems are NURBS (Non-Uniform Rational B-Spline), defined as follows

$$X_s(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{ij}(u, v) P_{i,j}$$

$$R_{ij}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^{n} \sum_{l=0}^{m} N_{k,p}(u) N_{l,q}(v) w_{k,l}}$$

- Position, tangent vectors and curvatures can be computed inexpensively for imposing (nonlinear) continuity constraints
Test points for imposing constraints

- Continuity is enforced at test points located along the joint edge.

![Diagram showing test points distributed along the joint edge.]

- Constraint functions evaluated at test points:

\[
C_{G0} = (X_s)_{left} - (X_s)_{right} = 0
\]
\[
C_{G1} = (\vec{\tau})_{left} \times (\vec{\tau})_{right} = 0
\]
\[
C_{G2} = (k)_{left} - (k)_{right} = 0
\]
Imposing continuity constraints using nullspace

- Control points are only allowed to move within the nullspace of the linearized constraint equations

\[
\frac{\partial C_{Gi}}{\partial P_i} dP_i = 0 \ (i = 0, 1, 2)
\]

\[\Rightarrow \delta \tilde{P} = \text{Ker} \left( \frac{\partial C_{G0}}{\partial P}, \frac{\partial C_{G1}}{\partial P}, \frac{\partial C_{G2}}{\partial P} \right) \delta \tilde{\alpha}\]

- Nonlinear constraints \(G_1\) and \(G_2\) can be dealt with
- Nullspace is independent of mesh size, thus remains inexpensive to compute for large cases
Imposing continuity constraints using nullspace

- Control points are only allowed to move within the nullspace of the linearized constraint equations

\[ \frac{\partial C_{G_i}}{\partial P_i} dP_i = 0 \quad (i = 0, 1, 2) \]

\[ \implies \delta \vec{P} = \text{Ker} \left( \frac{\partial C_{G_0}}{\partial P}, \frac{\partial C_{G_1}}{\partial P}, \frac{\partial C_{G_2}}{\partial P} \right) \delta \vec{\alpha} \]

- Nonlinear constraints \( G_1 \) and \( G_2 \) can be dealt with

- Nullspace is independent of mesh size, thus remains inexpensive to compute for large cases
Imposing continuity constraints using nullspace

- Control points are only allowed to move within the nullspace of the linearized constraint equations

\[
\frac{\partial C_G^i}{\partial P_i} dP_i = 0 \quad (i = 0, 1, 2)
\]

\[
\implies \delta \vec{P} = \text{Ker} \left( \frac{\partial C_{G0}}{\partial P}, \frac{\partial C_{G1}}{\partial P}, \frac{\partial C_{G2}}{\partial P} \right) \delta \vec{\alpha}
\]

- Nonlinear constrains $G1$ and $G2$ can be dealt with
- Nullspace is independent of mesh size, thus remains inexpensive to compute for large cases
Outline

Motivation

Implementation

Results

Summary and future work
S-bend shape optimisation, case parameters

- Geometry parameterisation: 30 NURBS patches, with 8 patches for "S" with 288 C.P.
- 203 design variables, 313,925-cell mesh, $Re_H = 300$
- Solver: in-house incompressible discrete adjoint solver

Figure: NURBS patches (left) and all-hex mesh (right).

(Ref: D. Jones, J.-D. Müller and S. Bayyuk, CFD Development with Automatic Differentiation, AIAA-2012-573)
S-bend shape optimisation, primal validation

- Small separation bubble about to reach pressure outlet
- Flow speed contour plots at outlet indicate complex secondary flow

Figure: Flow speed contour plots in medien plane and outlet boundary plane, GPDE v.s. FLUENT.
S-bend shape optimisation, results

Iteration 0
\(\Delta P = 0.015273 \text{ Pa}\)
Improvement 0.00%
S-bend shape optimisation, results

Iteration 1
\[ \Delta P = 0.015182 \text{ Pa} \]
Improvement 0.60%
S-bend shape optimisation, results

Iteration 2
ΔP = 0.014839 Pa
Improvement 2.84%
S-bend shape optimisation, results

Iteration 3
ΔP = 0.013074 Pa
Improvement 14.4%
S-bend shape optimisation, results

Iteration 4
ΔP = 0.012610 Pa
Improvement 17.4%
S-bend shape optimisation, results

Iteration 5
\( \Delta P = 0.012214 \text{ Pa} \)
Improvement 20.0\%
S-bend shape optimisation, results

- Unexpected shape at 6th iteration

![Updated shape at 6th iteration.](image)

- Possible cause for failure
  - Interference from non-vanishing large sensitivity at S-bend throat ⇒ allow the throat to deform
S-bend shape optimisation, results

- Unexpected shape at 6th iteration

Figure: Updated shape at 6th iteration.

- Possible cause for failure
  - Interference from non-vanishing large sensitivity at S-bend throat ⇒ allow the throat to deform
Outline

Motivation

Implementation

Results

Summary and future work
Summary

A novel CAD-based optimisation method is developed, using the control points of the boundary representation (BRep) for parameterisation

- Good interface with CAD
- Continuity constraints easy to impose
- Extendable to more complex geometry
- Cost is negligible compared to flow solver
Future work

- **Short-term (Apr. 2012)**
  - Implementation of G2 constraint
  - Extend to more complex geometries

![Examples of more complex geometric entities.](image)

- **Long-term (May 2011- May 2013)**
  - Apply to turbomachinery components, such as compressor/turbine blade shape optimisation
Thank you! Questions?