

# Introduction to Gradient-Based Optimisation

## Part 7: Shape arametrisation

Dr. J.-D. Müller  
School of Engineering and Materials Science,  
Queen Mary, University of London  
j.mueller@qmul.ac.uk

UK Fluids Network SIG on Numerical Optimisation with Fluids  
Cambridge, 8-10 August 2018

© Jens-Dominik Müller, 2011-18, updated 9/8/18

Notes

---

---

---

---

---

---

---

---

1 / 1

## Organisation of the lectures

1. Introduction
2. Univariate optimisation
  - Bisection, Secant, Newton's method
3. Multivariate optimisation
  - Steepest descent, Newton, CG, TN, Quasi-Newton methods.
  - Line-search methods: Wolfe conditions, trust regions
4. Constrained Optimisation:
  - Projected gradient methods,
  - Penalty methods, exterior and interior point methods, SQP
5. Gradient computation
  - Manual derivation, Finite Differences
  - Algorithmic and automatic differentiation, fwd and bkwd.
6. Adjoint methods
  - Reversing time, Automatic Differentiation
  - Adjoint CFD codes

Notes

---

---

---

---

---

---

---

---

2 / 1

## Outline

Notes

---

---

---

---

---

---

---

---

3 / 1

## The economics of parametrisation: To save or to squander?

"He who does not economize will have to agonize."

Confucius (551 - 479 BC)



Notes

---

---

---

---

---

---

---

---



"I spent a lot of money on booze, birds and fast cars. The rest I just squandered."

George Best (1946 - 2005 AD)

4 / 1

## To save or to squander? The case for economy

- Designers prefer to work with "engineering parameters", such as thickness, camber, LE radius, ...
- This maps onto their understanding of basic principles, but is often limited to 1-D or 2-D analogies.
- Integrates straightforwardly with manual workflow chains, manual override, can be done in CAD.
- Geometric constraints can be built into the parametrisation.
- But how to ensure that the innovative solution can be expressed with this restricted set?
- You could define thickness and camber as spline curve distributions, add knots. That adds parameters, but does it add the mode you need?

Notes

---

---

---

---

---

---

---

---

5 / 1

## To save or to squander ? The case to be profligate

- Gradient-based optimisation using adjoint solvers do not [really] penalise for size of the design space.
- The richest design space the solver can see is the mesh.
- How much do you smooth?
- How do you express all the mesh deformation modes you need in CAD?
- How do you impose constraints?

Notes

---

---

---

---

---

---

---

---

6 / 1

## To save or to squander ? Mixed economy

1. Acceptance by the designer/into the existing workflow requires that optimisation needs to be able to work with engineering parameters.
  - **CAD in the loop.**
  - We need gradients for the CAD system.
  - **Design philosophy "Inflation": start with a small design space, enrich.**
2. Rich design spaces need to be available for exploratory "out of the box" design, new configurations.
  - **but needs to be in CAD.**
  - Need to develop efficient methods to impose constraints.
  - **Design philosophy "Bill Gates": have it all, and give away (to constraints) only if you are forced to.**

Both approaches are needed, both need adjoint solvers and differentiated CAD.

Notes

---

---

---

---

---

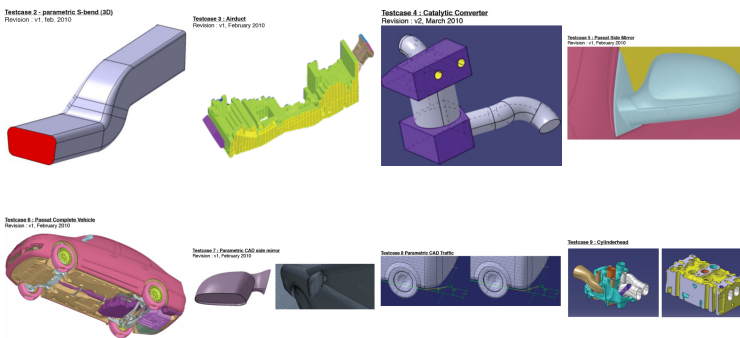
---

---

---

7 / 1

## Practical industrial shape optimisation cases



Notes

---

---

---

---

---

---

---

---

8 / 1

## Outline

Notes

---

---

---

---

---

---

---

---

9 / 1

## Shape parametrisation with gradient-based optimisation

To compute sensitivities of design variables  $\alpha$ , complete with the chain rule. In tangent mode:

$$\frac{\partial J}{\partial \alpha} = \underbrace{\frac{\partial J}{\partial R}}_{\text{Tangent}} \underbrace{\frac{\partial R}{\partial X_V}}_{\text{Metrics}} \underbrace{\frac{\partial X_V}{\partial X_S}}_{\text{Mesh}} \underbrace{\frac{\partial X_S}{\partial \alpha}}_{\text{Parametrisation}}$$

Can be evaluated analytic, or with finite differences, AD, complex ...

Since  $N_\alpha \ll N_S \ll N_V$ , efficiency demands reverse differentiation:

$$\frac{\partial J^T}{\partial \alpha} = \underbrace{\frac{\partial X_S^T}{\partial \alpha}}_{\text{Parametrisation}} \underbrace{\frac{\partial X_V^T}{\partial X_S}}_{\text{Mesh}} \underbrace{\frac{\partial R^T}{\partial X_V}}_{\text{Metrics}} \underbrace{\frac{\partial J^T}{\partial R}}_{\text{Adjoint}}$$

Can be evaluated analytic, or with AD. Not with F-D, complex ..

10 / 1

Notes

---

---

---

---

---

---

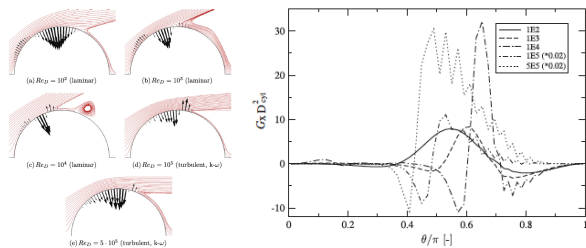
---

---

## Mesh-based parametrisation method

Node-based, or "parameter-free" approach

- Definition: use the coordinates of the design surface nodes as the design variable, i.e.,  $\frac{\partial X_S}{\partial \alpha} = \mathbf{I}$
- Optimisation procedure:
  - calculate node-based sensitivity using adjoint:  $\nabla J$
  - filter the (possibly noisy) raw sensitivity:  $\nabla J \rightarrow \nabla \tilde{J}$
  - use optimizer to drive surface node perturbation,  $\delta X_S$
  - perturb the volume mesh,  $\delta X$  (remesh or deform)



11 / 1

Stück, A.; Bung, T. Adjoint RANS with filtered shape derivatives for hydrodynamic optimization. 2011. Computers

Notes

---

---

---

---

---

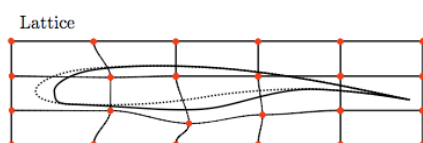
---

---

---

## Free form deformation, aka morphing

- Definition: use the vertices of the 2D/3D lattice to parameterize and deform the imbedded curve/surface
- Optimisation procedure:
  - calculate node-based shape sensitivity  $dJ/dX_S$
  - $dJ/d\alpha = dJ/dX_S \cdot dX_S/d\alpha$
  - use optimizer to drive surface and volume mesh perturbation,  $\delta X_S$  and  $\delta X$



Notes

---

---

---

---

---

---

---

---

12 / 1

## CAD-parameter based approach

- establish a CAD parametrisation that defines a suitable design space
- Obtain derivatives:
  - finite-differences (Robinson, Armstrong, QUB)
  - analytically for simple shapes (Haimes, MIT)
  - automatic differentiation (QMUL, UPB)
- Complete the chain rule

Notes

---

---

---

---

---

---

---

---

13 / 1

## Lattice-based

- Definition: use the control points of the analytic forms of the geometry as design variables
- Optimisation procedure:
  - calculate node-based shape sensitivity  $dJ/dX_s$
  - analytically calculate shape derivative matrix  $dX_s/d\alpha$
  - $dJ/d\alpha = dJ/dX_s \cdot dX_s/d\alpha$
  - use optimizer to perturb geometry  $\delta\alpha$  and surface mesh
  - Re-meshing or adapt the volume mesh

Notes

---

---

---

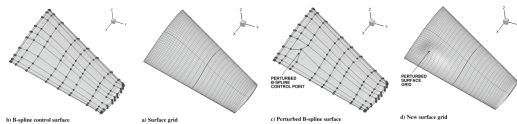
---

---

---

---

---



Hicken, J. E. ;Zingg, D. W., Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement, 2010, Aiaa Journal

14 / 1

## Outline

Notes

---

---

---

---

---

---

---

---

15 / 1

## Node-based parameterisation or no parameterisation

- Node-based parameterisation uses the coordinates of the surface nodes as the design variable, i.e.,  $\frac{dX_s}{d\alpha} = \mathbf{I}$
- Only need to calculate  $\frac{dJ}{dX_s} = \frac{dJ}{dX} \cdot \frac{dX}{dX_s}$
- It remains one of the most popular methods due to its rich design space and simplicity in implementation
- Issues need to be tackled when applied to practical design optimisation cases
  - **Smoothing** of both gradient and perturbation
  - **Regularisation** of surface and volume mesh
  - **Preconditioning** of the gradient for conv. speedup

Notes

---

---

---

---

---

---

---

---

16 / 1

## Node-based parametrisation, smoothing

- Fine mesh and large Re number tend to produce noisy shape sensitivity map  $G = dF/dX_s$  which needs to be smoothed (through a low-pass filter) to maintain the smoothness of the optimised shape
- Explicit smoothing, targets highest frequencies

$$\bar{G}(\xi) = \int_{\Gamma_D} K(r)G(\xi - r)dr, \quad \forall \xi \in \Gamma_D$$

with a conservative low-pass filter kernel  $K(r)$  s.t.

$$\int_{\Gamma_D} K(r)dr = 1$$

- Implicit smoothing

$$(\mathbf{I} - \varepsilon\Delta)\bar{G} = G$$

Notes

---

---

---

---

---

---

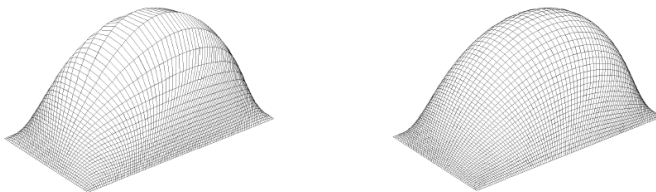
---

---

17 / 1

## Node-based parametrisation, regularisation

The perturb surface mesh with lower mesh quality needed to be optimised/regularized (smoothing the perturbation), example:



Notes

---

---

---

---

---

---

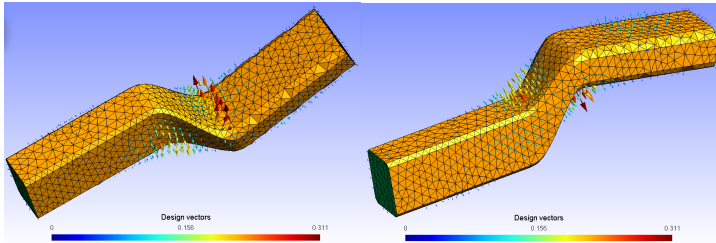
---

---

18 / 1

## Node-based parametrisation, applied to S-Bend

- $Re=100$ , laminar flow; cost function: total pressure loss
- Design var.: normal displacement of the surface nodes
- Optimiser: L-BFGS,  $\approx 1000$  design variables
- Mesh deformation: spring-analogy



Notes

---

---

---

---

---

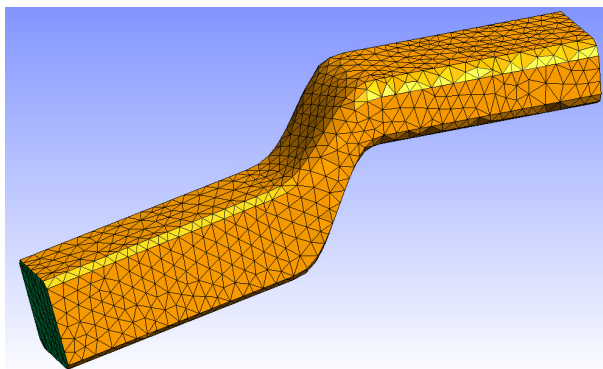
---

---

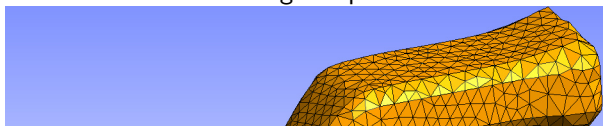
---

19 / 1

## S-Bend optimisation, node-based parametrisation



Design step 1



20 / 1

Notes

---

---

---

---

---

---

---

---

## Summary: node-based parametrisation

- Easy to implement: normal displacement of each surface node is a design variable
- Rich design space: maximum DoFs for a particular mesh
- Difficulty 1: smoothing for the perturbed mesh is needed to avoid oscillatory shapes, difficult to determine the appropriate level of smoothing
- Difficulty 2: Optimised shape exists as mesh, needs to transfer back to CAD file

Notes

---

---

---

---

---

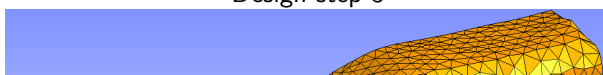
---

---

---

21 / 1

Design step 3



# Outline

Notes

---

---

---

---

---

---

---

22 / 1

## The economics of parametrisation: the case for saving

“He who does not economize will have to agonize.”

Confucius (551 - 479 BC)



Notes

---

---

---

---

---

---

---



“I spent a lot of money on booze, birds and fast cars. The rest I just squandered.”

George Best (1946 - 2005)

23 / 1

## Open Cascade Technology

Notes

---

---

---

---

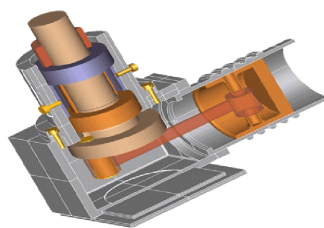
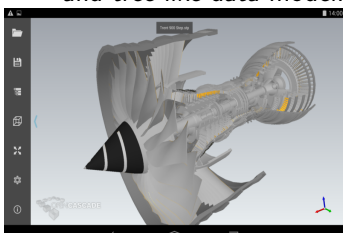
---

---

---

Open CASCADE<sup>1</sup> Technology (OCCT) is an open source C++ library, consisting of thousands of classes and providing solutions in the areas of:

- Surface and solid modelling: to model any type of object,
- 3D and 2D visualisation: to display and animate objects,
- Data exchange (import and export standard CAD formats) and tree-like data model.



<sup>1</sup><http://www.opencascade.com/>

24 / 1



## Calculating the CAD sensitivities (I)

Joint work with Univ Paderborn (Prof. A. Walther, M. Banovic)  
apply Automatic Differentiation (AD) tool ADOL-C to the CAD sources.

### Automatic Differentiation by OverLoading in C++

ADOL-C uses **operator overloading** concept to compute first and higher derivatives of vector functions that are written in C or C++.

#### Operator overloading (Scalar mode)

```
class myadouble{
  double value;
  double ADvalue;
  // multiplication ...
  inline myadouble operator * (const myadouble& a) const {
    myadouble tmp;
    tmp.value = value * a.value;
    tmp.ADvalue = ADvalue * a.value + value * a.ADvalue;
    return tmp;
  }
};
```

25 / 1

Notes

---

---

---

---

---

---

---

---

## Calculating the CAD sensitivities (I)

An exact way of calculating the shape sensitivities w.r.t. design parameters is to apply the Automatic Differentiation (AD) to the CAD sources. AD tool integrated to OCCT is **ADOL-C**.

### Automatic Differentiation by OverLoading in C++

ADOL-C uses **operator overloading** concept to compute first and higher derivatives of vector functions that are written in C or C++.

#### Operator overloading (Vector mode)

```
class myadouble{
  double value;
  double *ADvalue = new double[NUMBER_OF_DIRECTIONS];
  // multiplication ...
  inline myadouble operator * (const myadouble& a) const {
    myadouble tmp;
    tmp.value = value * a.value;
    for(size_t i = 0; i < NUMBER_OF_DIRECTIONS; ++i)
      tmp.ADvalue[i] = ADvalue[i] * a.value + value * a.ADvalue[i];
    return tmp;
  }
};
```

26 / 1

Notes

---

---

---

---

---

---

---

---

## Calculating the CAD sensitivities (II)

### ADOL-C vector mode

- Number of design parameters = NUMBER\_OF\_DIRECTIONS.
- Derivatives w.r.t. all design parameters are evaluated with just one code run.
- Computational cost and memory requirements are dependent on the NUMBER\_OF\_DIRECTIONS.

### Example of sensitivities calculated using AD vector mode



Notes

---

---

---

---

---

---

---

---

27 / 1

# Outline

Notes

---

---

---

---

---

---

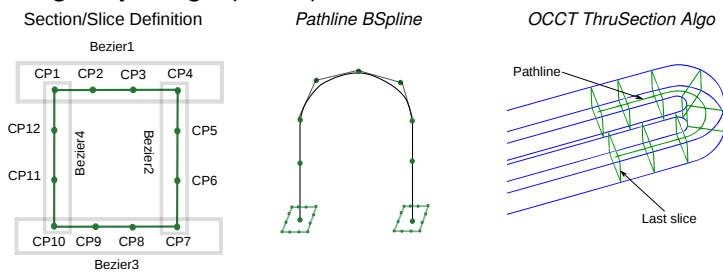
---

---

28 / 1

## Explicit parametrisation for turbo-machinery U-Bend

The generic slice consists of 4 Bézier curves; it is swept orthogonally along a planar pathline.



Notes

---

---

---

---

---

---

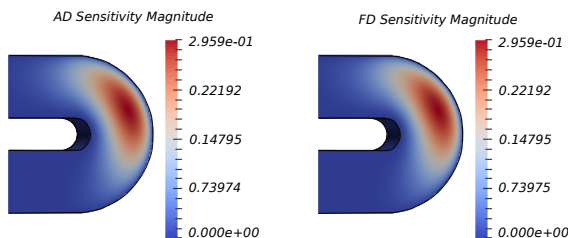
---

---

- Each control point of the section is defined by a law of evolution along a B-spline curve along the pathline.
- The 96 coordinates of its control points are the **design parameters**.

29 / 1

## Gradient comparison



Notes

---

---

---

---

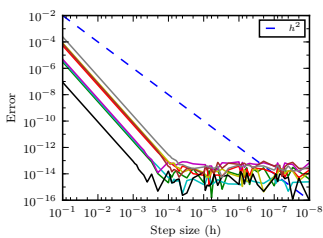
---

---

---

---

U-part shape sensitivities using AD (left), F-D (right).

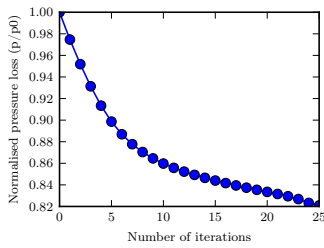


Taylor test (extrapolation of AD gradients vs. undivided difference) for eight U-bend surface point coordinates

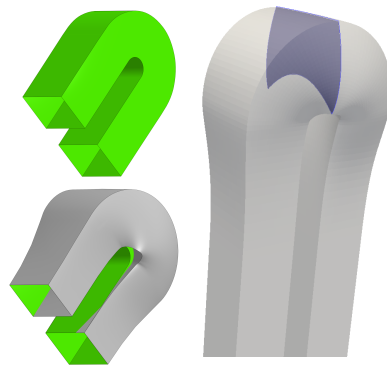
30 / 1

## U-Bend optimisation: optimised shape

Notes



Optimisation algorithm:  
BFGS.



Initial and final shape.

---

---

---

---

---

---

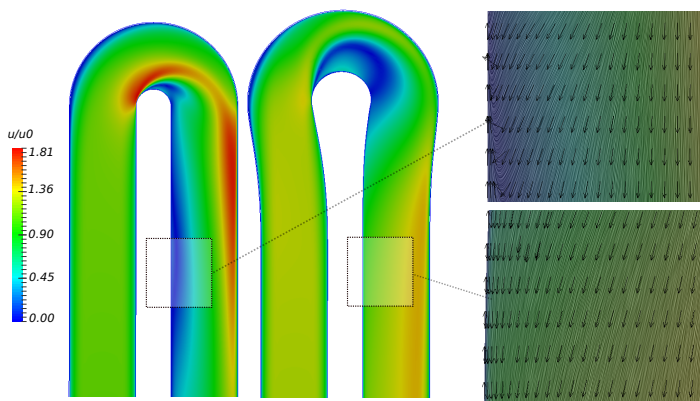
---

---

31 / 1

## U-Bend optimisation, flowfield

Notes



Left: Baseline and Optimised Mid-span Velocity Magnitude; Right: Flow Streamlines in the outlet leg

---

---

---

---

---

---

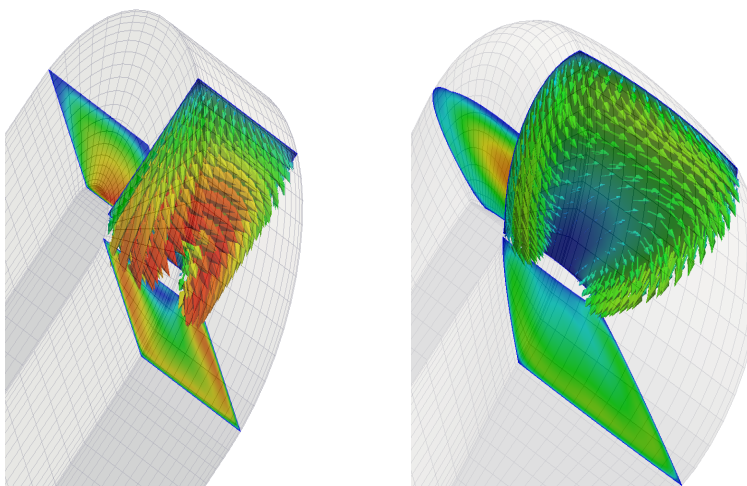
---

---

32 / 1

## U-Bend optimisation: velocity vectors at mid-turn

Notes



Baseline

Optimised

---

---

---

---

---

---

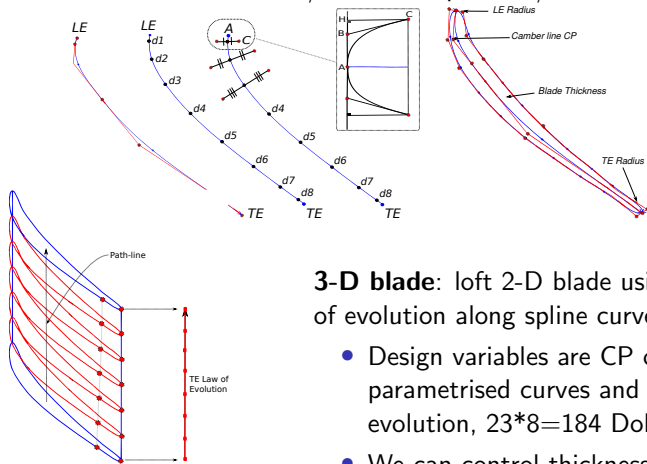
---

---

33 / 1

## Explicit Parametrisation for TUB Stator

2-D blade: camberline  $\rightarrow$  le/te radii  $\rightarrow$  pressure/suction curve



3-D blade: loft 2-D blade using laws of evolution along spline curves

- Design variables are CP of our parametrised curves and laws of evolution,  $23 \cdot 8 = 184$  DoF
- We can control thickness with bounding values

34 / 1

Notes

---

---

---

---

---

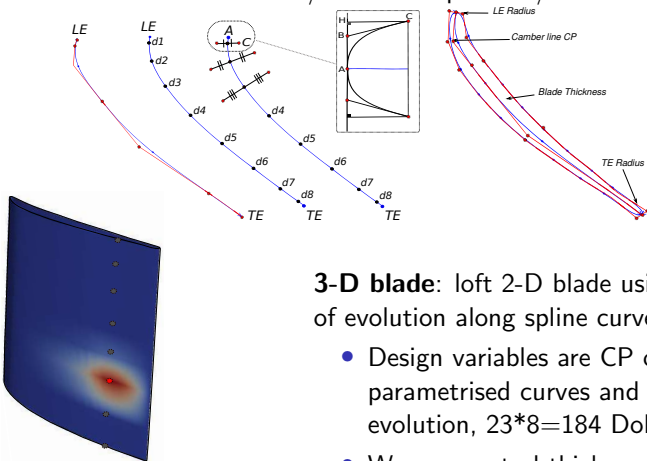
---

---

---

## Explicit Parametrisation for TUB Stator

2-D blade: camberline  $\rightarrow$  le/te radii  $\rightarrow$  pressure/suction curve



3-D blade: loft 2-D blade using laws of evolution along spline curves

- Design variables are CP of our parametrised curves and laws of evolution,  $23 \cdot 8 = 184$  DoF
- We can control thickness with bounding values

35 / 1

Notes

---

---

---

---

---

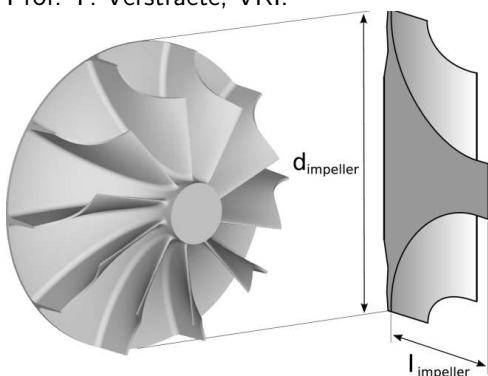
---

---

---

## Multi-disciplinary optimisation with CAD

Aero-structural optimisation of a radial turbine, joint work with Prof. T. Verstraete, VKI.



Optimise efficiency, subject to maximal stress constraint

Notes

---

---

---

---

---

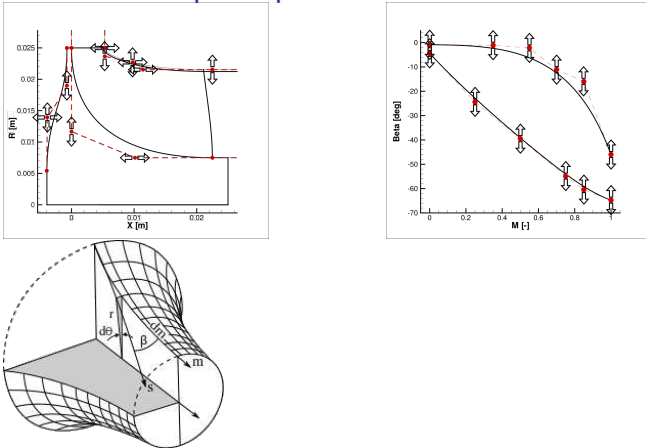
---

---

---

36 / 1

## Explicit parametric CAD model



Classic turbomachinery parametrisation, define hub and shroud surfaces, blade angle, blade thickness. Curves defined by B-splines. Implemented in VKI in-house CAD tool CADO, differentiated using Automatic Differentiation tool ADOL-C.

37 / 1

Notes

---

---

---

---

---

---

---

---

## Explicit CAD parametrisation

Notes

---

---

---

---

---

---

---

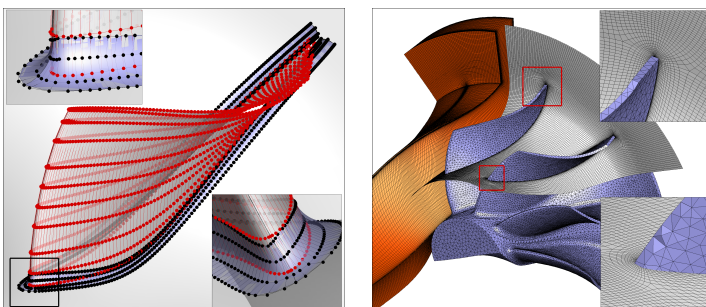
---

Shroud

Cut Back

38 / 1

## CAD Master Model as Datum Surface



Parametric model is 'shown' as NURBS surfaces to meshers.

Defines a unique reference surface for both disciplines.

Notes

---

---

---

---

---

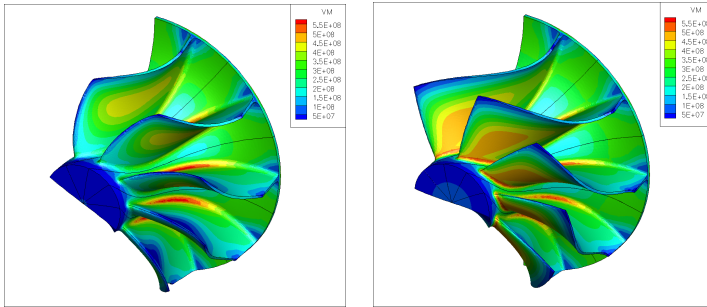
---

---

---

39 / 1

## Optimisation results



Baseline

Iteration 60

Notes

---

---

---

---

---

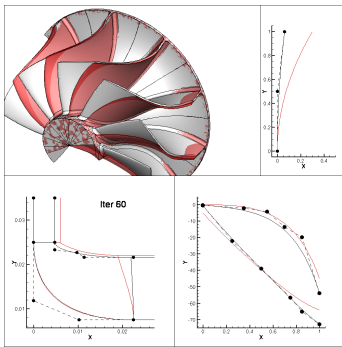
---

---

---

40 / 1

## Shape evolution, objective improvement



	Baseline	Iter 60
Total/static eff.	75.4%	81.4%
Total/total eff.	85.6%	87.9%
Max stress	461 MPa	446 MPa
Axial exit vel.	179 m/s	143 m/s
Exit swirl vel.	41.6 m/s	13.4 m/s
Outlet swirl	13.0 deg.	5.36 deg

- High-fidelity optimisation allows accurate trade-off between competing objectives of aero efficiency and structural load
- Significant improvement in efficiency!
- CAD surface is essential as datum to transfer shape changes between either discipline.

Notes

---

---

---

---

---

---

---

---

41 / 1

## Outline

Notes

---

---

---

---

---

---

---

---

42 / 1

## The economics of parametrisation: The case for squandering.

"He who does not economize will have to agonize."

Confucius (551 - 479 BC)



"I spent a lot of money on booze, birds and fast cars. The rest I just squandered."

George Best (1946 - 2005)

Notes

---

---

---

---

---

---

---

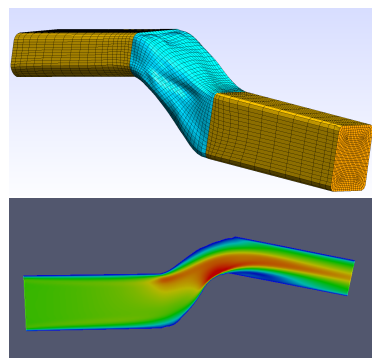
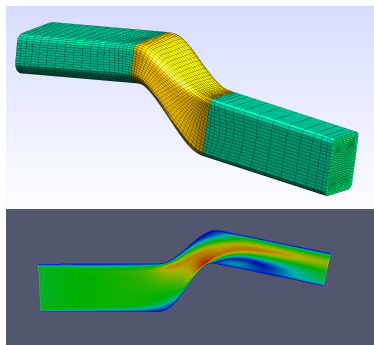
---

43 / 1

## Node-based optimisation: mesh and velocity magnitude

Initial S-bend

Node-based optimisation



Notes

---

---

---

---

---

---

---

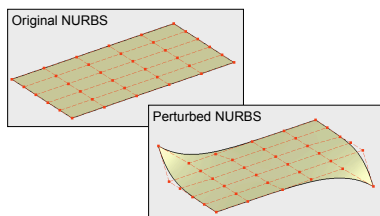
---

Unintuitive shape changes, how to capture this for CAD? How to impose geometrical constraints?

44 / 1

## CAD-based parametrisation using NURBS

To integrate num. optimisation into the design chain, the optimised shape must exist in a CAD format.



- NURBS-based approach: use the position of the control points and weights of the NURBS as design variables.
- The difficulty is to maintain geometric continuity across patch interfaces for complex CAD geometries.
- QMUL group developed the NSPCC methodology to ensure G0 (watertight), G1 (tangency) or G2 (curvature) continuity.

(NSPCC: NURBS-based Parametrisation with Complex Constraints)

Notes

---

---

---

---

---

---

---

---

45 / 1

## NSPCC: NURBS-based parametrisation with complex constraints

When using more than one or unconstrained patches, the parametrisation needs to ensure that patches have G0 (watertight), G1 (tangency) or G2 (curvature) continuity:

- Standard methods define geometric constraints on control point positions, e.g. first control point inside needs to be in the tangency plane for G1.
- This is not practical for general cases.

Alternatively:

- iterate on control point positions to satisfy constraints
- evaluate constraints at test-points distributed along patch boundaries/intersections.

Notes

---

---

---

---

---

---

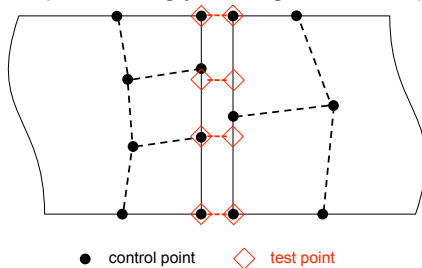
---

---

46 / 1

### Continuity constraint

- Continuity is required along joint edges between patches



- Constraint functions are evaluated at test points

$$G_0 = (X_s)_L - (X_s)_R \text{ linear}$$

$$G_1 = (\vec{\tau})_L \times (\vec{\tau})_R \text{ non-linear}$$

$$G_2 = (\vec{k})_L - (\vec{k})_R \text{ non-linear}$$

and are required to maintain their initial values at each design step.

Notes

---

---

---

---

---

---

---

---

47

### Continuity constraint II

Constraint functions are linearised w.r.t. control points

$$\delta G = \frac{\partial G}{\partial P} \delta P + \text{h.o.t.}$$

Let the perturbation  $\delta P$  lie in the null-space of  $\partial G / \partial P$ ,

$$\delta P = \ker C \cdot \delta \alpha$$

A basis  $\ker C$  for the null-space of  $C$  is computed using SVD.

$G_1$  and  $G_2$  will only be approximately zero after perturbation, then the range space is used to take a normal step to recover the non-linear constraints.

Notes

---

---

---

---

---

---

---

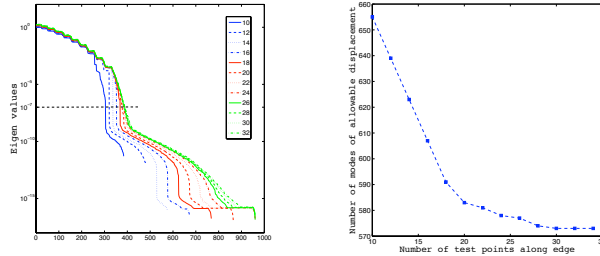
---

48 / 1



## Continuity constraint III

- The number of non-zero singular values determines the design space: the SVD computes an orthogonal basis, which significantly improves convergence rate
- The cut-off value for 'zero' singular allows to select the size of the design space.



Notes

---

---

---

---

---

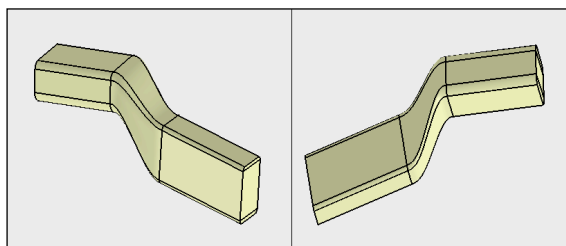
---

---

---

49 / 1

## Optimisation of a Volkswagen S-Bend climate duct



- 30 NURBS patches, only cranked centre section is allowed to move. 640 control points, 1920 DoF.
- Imposing G1 continuity constraints leaves a design space of 570 modes.
- Objective function: minimise pressure loss from inlet to outlet.

Notes

---

---

---

---

---

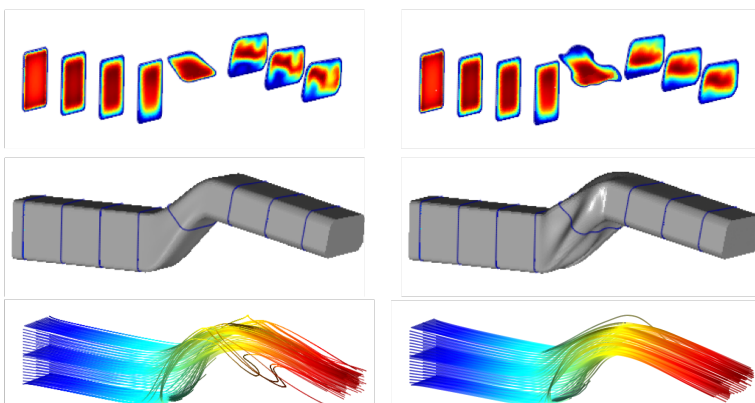
---

---

---

50 / 1

## S-Bend: Improvement in flow field



Notes

---

---

---

---

---

---

---

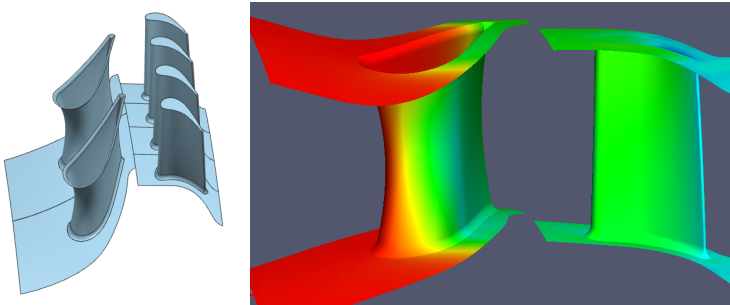
---

The optimisation produces strake-like features which very effectively suppress secondary flow motion.

51 / 1

## Turbomachinery optimisation with CAD-based parametrisation

A high-pressure turbine geometry provided by Rolls-Royce is optimised using CAD-based parametrisation.



Pressure contours on nozzle guide vane (NGV) and rotor

Notes

---

---

---

---

---

---

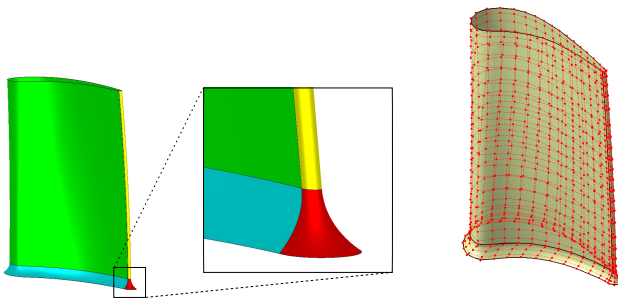
---

---

52 / 1

## CAD-parametrisation

The rotor is modelled using 4 NURBS patches joined with  $G_1$  continuity. The control points of the NURBS patches are used as design variables.



Notes

---

---

---

---

---

---

---

---

53 / 1

## Optimisation setup

Objective function:

$$J = \frac{\eta - \eta^0}{\eta^0} + a_1 \left( \frac{\phi - \phi^0}{\phi^0} \right)^2 + a_2 \left( \frac{\chi - \chi^0}{\chi^0} \right)^2$$

where

Efficiency:  $\eta = \frac{(\dot{m}h)_{0,1} - (\dot{m}h)_{0,3}}{(\dot{m}h)_{0,1} - (\dot{m}h)_{0s,3}}$

Inlet capacity:  $\phi = \frac{\dot{m}\sqrt{T_{0,1}}}{p_{0,1}}$

Reaction:  $\chi = \frac{h_2 - h_3}{h_{0,1} - h_{0,3}}$

subject to:

Flow constraint:  $|\delta\phi|/\phi_0 \leq 0.1, \quad |\delta\chi|/\chi_0 \leq 1.0\%$

Geometric constraint:  $G_1$  – continuity

Notes

---

---

---

---

---

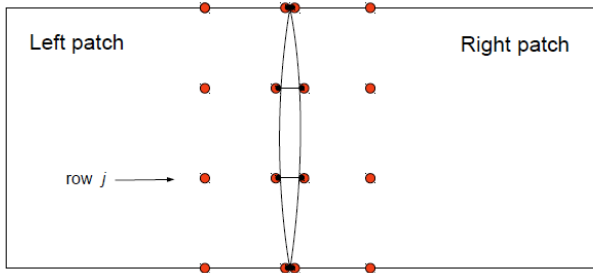
---

---

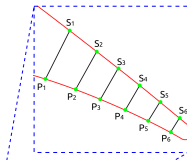
---

54 / 1

## Impose thickness and radius constraints



- Control points ( $X^{cp}$ )
- Test points ( $\xi_j$ )



55 / 1

Notes

---

---

---

---

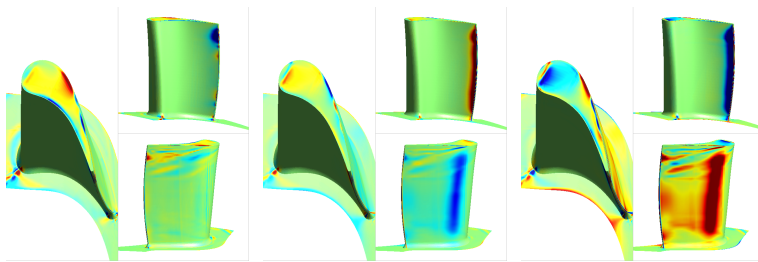
---

---

---

---

## Adjoint surface sensitivities



efficiency

capacity

reaction ratio

56 / 1

Notes

---

---

---

---

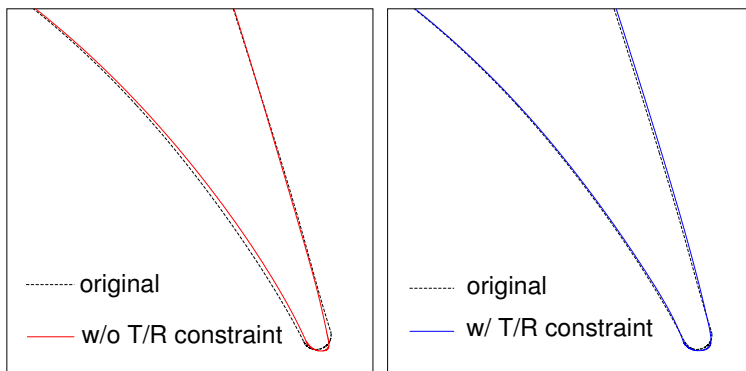
---

---

---

---

## Impose thickness and radius constraints



without T/R constraint

with T/R constraint

57 / 1

Notes

---

---

---

---

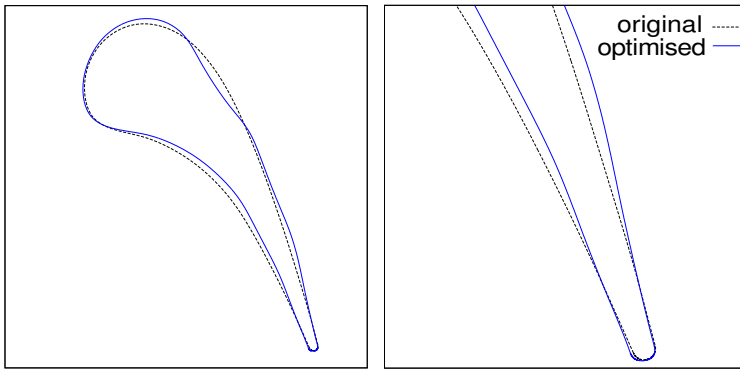
---

---

---

---

## Constrained optimisation: rotor tip change



The tip leakage vortex induces significant changes in the blade profile at the tip.

Notes

---

---

---

---

---

---

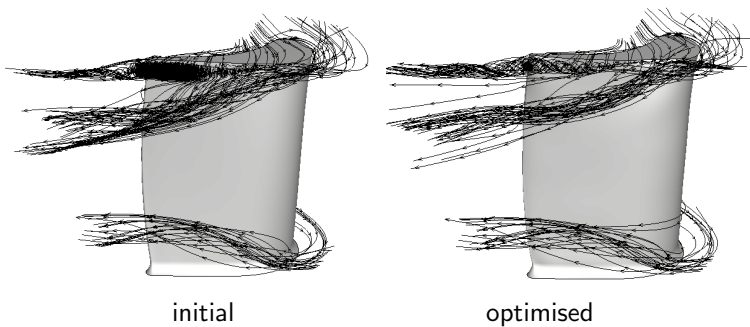
---

---

58 / 1

## Constrained optimisation

The strong profile change in the tip region diminishes the tip leakage vortex.



Notes

---

---

---

---

---

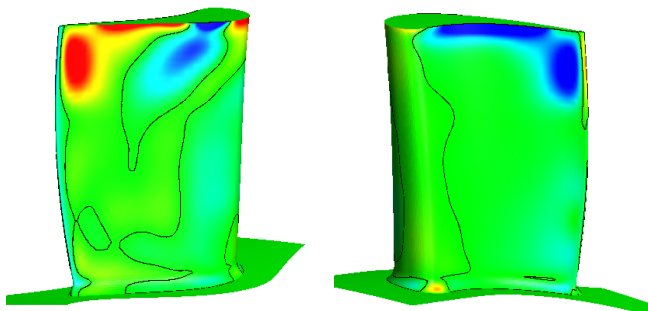
---

---

---

59 / 1

## Result: shape change



Normal deformation plotted on the original shape (left: suction side, right: pressure side). **blue**: inward **red**: outward. Contour line marks zero displacement.

Notes

---

---

---

---

---

---

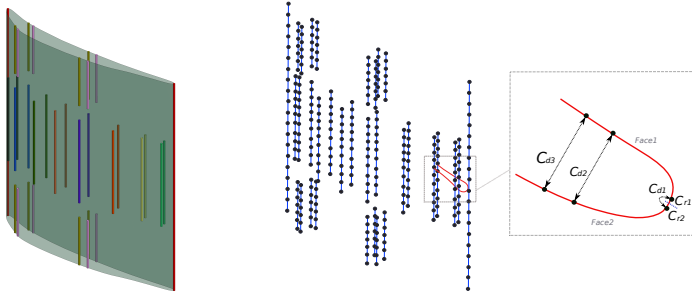
---

---

60 / 1

## Implicit Parametrisation in OCCT using NURBS patches

Geometric constraints evaluated at test points



Control Points Updates and Constraints Matrix (SVD)

$$C = \frac{\partial C_x}{\partial P} = \begin{pmatrix} \frac{\partial C_{d1}}{\partial P_1} & \frac{\partial C_{d2}}{\partial P_1} & \dots & \frac{\partial C_{cr1}}{\partial P_1} & \frac{\partial C_{cr2}}{\partial P_1} & \dots & \frac{\partial C_{cr}}{\partial P_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_{d1}}{\partial P_N} & \frac{\partial C_{d2}}{\partial P_N} & \dots & \frac{\partial C_{cr1}}{\partial P_N} & \frac{\partial C_{cr2}}{\partial P_N} & \dots & \frac{\partial C_{cr}}{\partial P_N} \end{pmatrix} \quad (1)$$

✓ Constraints  $P^{n+1} = P^n + t \cdot \ker(C) \left[ (\nabla J) \ker(C) \right]^T$  (6)

61 / 1

Notes

---

---

---

---

---

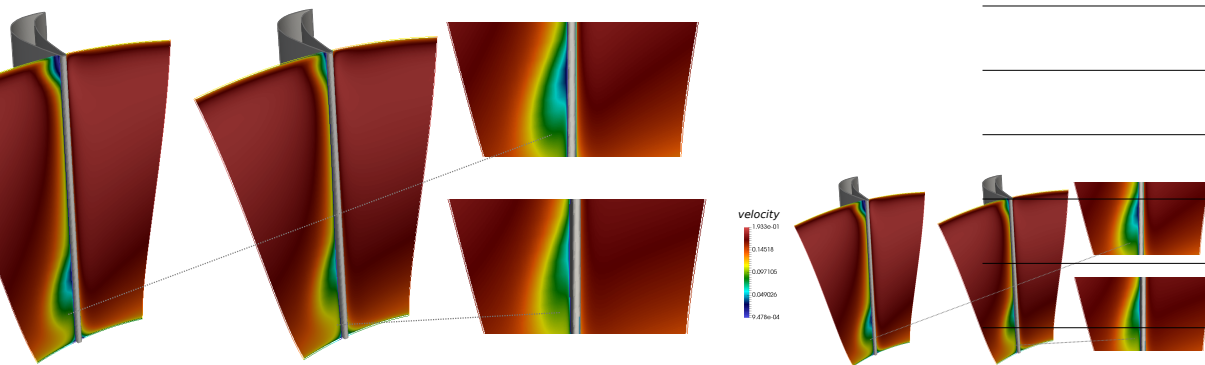
---

---

---

## TUB Stator Optimisation results using NSPCC parametrisation

Optimised CFD flow, block-structured mesh, 0.4M hexahedra.



Optimisation History



62 / 1

Notes

---

---

---

---

---

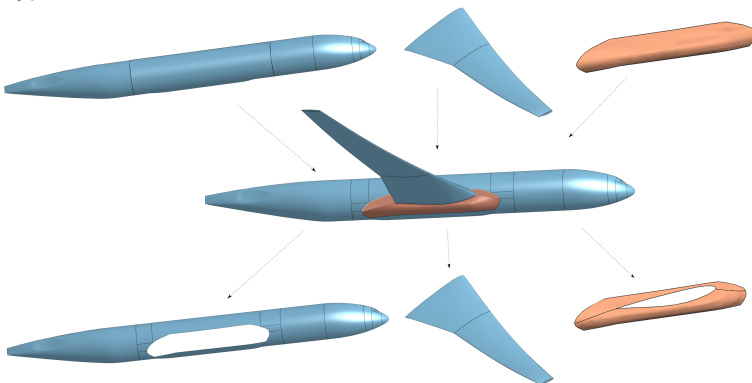
---

---

---

## CAD Boolean Operations and Trimmed Faces

Typical CAD Workflow: CRM



- Independent parts are trimmed after Boolean Operations (Fuse, Common): Surface-Surface intersections
- We want to optimise geometries with trimmed patches
- We want to find places where trims should be performed

63 / 1

Notes

---

---

---

---

---

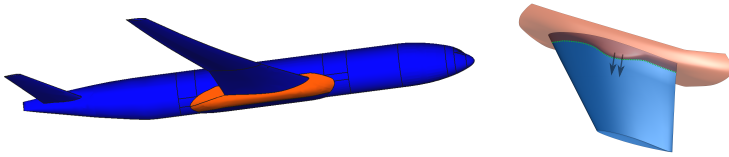
---

---

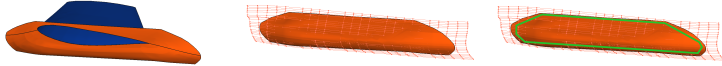
---

## Testcase: Wing-fairing intersection

Design area (orange)



NURBS Parametrisation



- Fairing design: 484 Control Points
- 169 Control Points allowed to move = 507 design variables
- Continuity with Fuselage

Notes

---

---

---

---

---

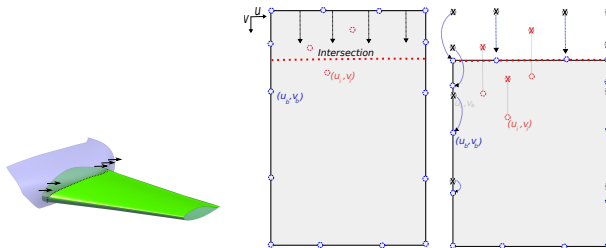
---

---

---

64 / 1

## Mesh Morphing



0. Mesh point inversions: Find  $(\text{Face}_{CAD}, u, v)$

CAD-Mesh Movement Algorithm:

1. Change Control Points  $P$  of the fairing patch
2. Recalculate new intersection curve
3. Sample curve to define displacement  $(\delta u_b, \delta v_b)$ : 1-D
4. Updated mesh points  $(u_i, v_i)$  in parametric space: 2-D IDW
5. Project to Cartesian space  $(x, y, z) = S(u, v)$
6. Propagate surface movement into volume: 3-D IDW

Notes

---

---

---

---

---

---

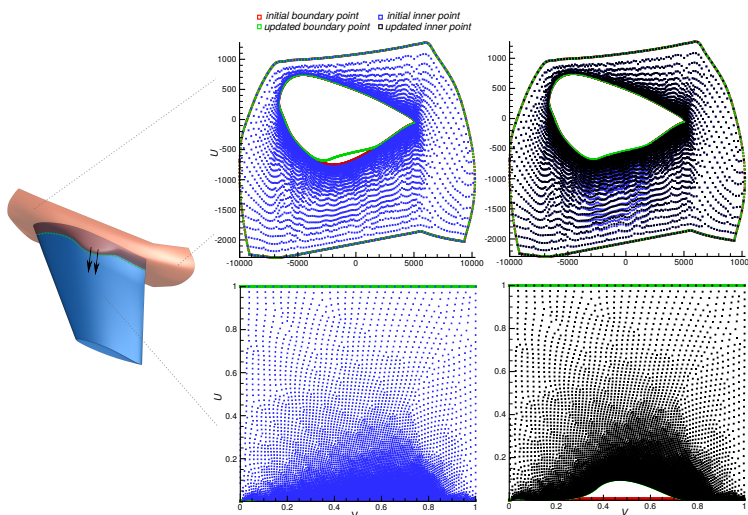
---

---

65 / 1

## Parametric 2-D Mesh Movement

2-D IDW on Fuselage and the wing



Notes

---

---

---

---

---

---

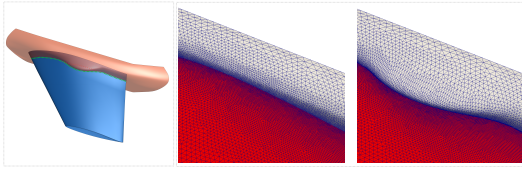
---

---

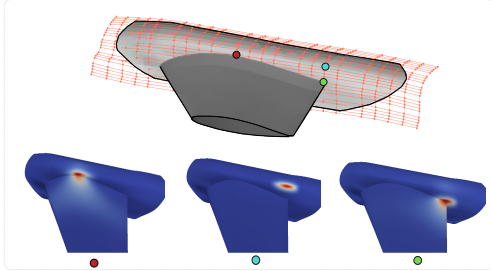
66 / 1

## Computing CAD sensitivities for trimmed patches

3-D Mesh Movement



**Algorithm Intersections Sensitivity:** Move CPs, AD OCCT intersection, mesh ( $1D \rightarrow 2D \rightarrow 3D$ )



67 / 1

Notes

---

---

---

---

---

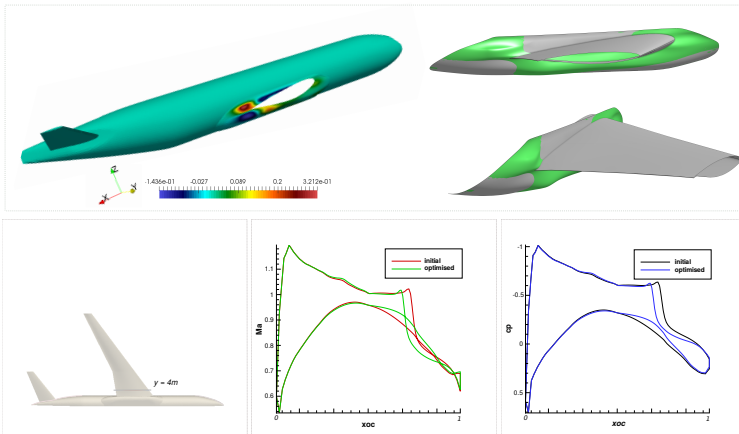
---

---

---

## NASA CRM Results

Non-trivial updated intersection



Result:  $\frac{C_L}{C_D}$  : +1.8%,  $C_L$  : -4%,  $C_D$  : -6%.

68 / 1

Notes

---

---

---

---

---

---

---

---

## Outline

Notes

---

---

---

---

---

---

---

---

69 / 1

## Volume mesh deformation technique, overview

- Maintaining the mesh topology is preferred: no jump in discretisation error, more efficient, remeshing may not be feasible.
- Mesh deformation propagates a surface change into the volume, its differentiation allows to compute  $\frac{dX}{dX_S}$

Desired features of a good mesh deformation algorithm

- low computational cost (negligible compared to flow/adjoint computation)
- maintain the original mesh characteristics (topology, local refinement, etc.)
- robustness and stability: can cope with large deformation while maintaining mesh quality.

Notes

---

---

---

---

---

---

---

---

70 / 1

## Inverse Distance Weighting

Compute the displacement of an internal (volume) mesh point by the sum of inverse distances of all boundary points:

$$\Delta x_k = \frac{1}{|j \in S|} \sum_{j \in S} \frac{\Delta x_j}{|x_k - x_j|}$$

- Very fast, global support.
- Can be augmented by Quaternions to maintain orthogonality with the surface.
- Does not guarantee positive volumes.

Notes

---

---

---

---

---

---

---

---

71 / 1

## Spring Analogy, Laplacian smoothing

- Each node is seen as a massless ball connected to neighboring nodes with a spring with stiffness  $k_{ij}$  inversely proportional to the edge length

$$k_{ij}(\Delta x_i - \Delta x_j) = 0$$

spring stiffness  $k_{ij} \propto \frac{1}{|x_i - x_j|}$

- Not very robust in dealing with large deformation (the spring-mass model has no mechanism to maintain positive volume)
- Need to solve a linear equation iteratively with cost comparative with the flow/adjoint solver

Notes

---

---

---

---

---

---

---

---

72 / 1



## Linear Elasticity

- The linear elasticity problem with Dirichlet boundary condition is solved on the existing mesh to find the interior mesh node displacement  $\Delta X$

Governing equation :  $\nabla \cdot \sigma + f = 0$  on  $\Omega$

Cahchy stress tensor :  $\sigma = \lambda \text{tr}(\epsilon(\Delta X))\mathbf{I} + 2\mu\epsilon(\Delta X)$

Strain tensor :  $\epsilon(\Delta X) = \frac{1}{2}(\nabla(\Delta X) + (\nabla(\Delta X))^T)$

Boundary condition :  $\Delta X = g$  on  $\partial\Omega$

- It's a PDE-based method, thus has physical mechanism to maintain positive volume for deformed elements
- The stiffness tensor can be localized to preserve baseline mesh features, e.g., stiffer in the boundary layer

Ref: Dwight, Richard P. "Robust mesh deformation using the linear elasticity equations." Computational Fluid Dynamics 2006. Springer Berlin Heidelberg, 2009. 401-406.

73 / 1

Notes

---

---

---

---

---

---

---

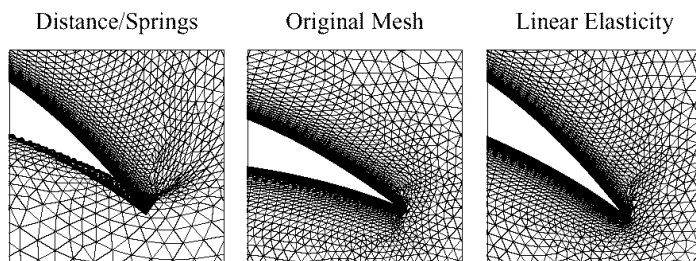
---

---

---

## Spring-analogy vs. linear elasticity

- Both are suitable for unstructured meshes
- Both can be modified to preserve boundary layer mesh
- Linear elasticity is more robust in dealing with large deformation
- One comparison example:



Ref: <http://fun3d.larc.nasa.gov/example-56.html>

74 / 1

Notes

---

---

---

---

---

---

---

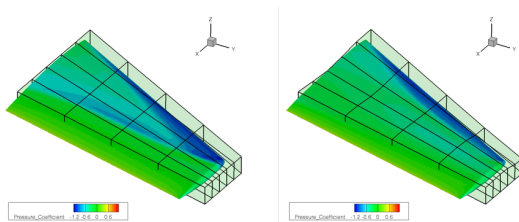
---

---

---

## Free Form Deformation (FFD)

- Perturb the control points for the enclosing lattice to morph the mesh:



- Can cope with large deformation, but difficult to localize the surface deformation
- Non-iterative method, computationally inexpensive

Palacios, Francisco, et al. "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design." 51st AIAA Aerospace Sciences Meeting. 2013.

75 / 1

Notes

---

---

---

---

---

---

---

---

---

---

## Radial Basis Functions

- The displacement of the surface and interior mesh points  $\Delta X$  at location  $r$  is governed by

$$\Delta X(r) = \sum_{i=1}^N \omega_i \phi(|r - r_i|)$$

where the weight coefficient  $\omega_i$  is determined by the prescribed displacement of the  $N$  perturbed surface nodes, i.e.,

$$\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_N \end{pmatrix} = \begin{pmatrix} \phi_{1,1} & \phi_{2,1} & \cdots & \phi_{N,1} \\ \phi_{1,2} & \phi_{2,2} & \cdots & \phi_{N,2} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{1,N} & \phi_{2,N} & \cdots & \phi_{N,N} \end{pmatrix} \cdot \begin{pmatrix} \Delta \omega_1 \\ \Delta \omega_2 \\ \vdots \\ \Delta \omega_N \end{pmatrix}$$

where  $\phi_{i,j} = \phi(|r_i - r_j|)$

de Boer, A., van der Schoot, M. S., Bijl, H., Mesh deformation based on radial basis function interpolation Year, 2007, Computers & Structures

76 / 1

Notes

---

---

---

---

---

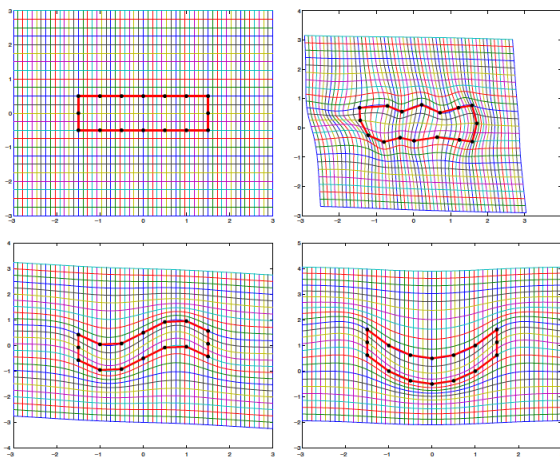
---

---

---

## Volume mesh deformation technique, cont'd

Example:



Jakobsson, S., Amoignon, O. Mesh deformation using radial basis functions for gradient-based aerodynamic shape

77 / 1

Notes

---

---

---

---

---

---

---

---

## Volume mesh deformation technique, summary

Mesh deform method	features & applicability
Inverse Distance Weighting	very fast, small $\delta x_s$ , unstructured grid
Spring-analogy	Fast, small $\delta x_s$ , unstructured grid
Linear-elasticity	Slow, large $\delta x_s$ , unstructured grid
Free form deformation	Fast, large $\delta x_s$ , unstructured grid difficult to localize surf deformation
Radial basis function	Slow, large $\delta x_s$ , unstructured grid localized surf. deformation

Notes

---

---

---

---

---

---

---

---

78 / 1

## Outline

Notes

---

---

---

---

---

---

---

---

79 / 1

## CAD based on BRep: NSPCC

Notes

NSPCC uses control points of the BRep to modify shape

- Approach is robust,
- Effective automatic parametrisation
- Rich design space, equivalent to node-based approach
- Output is CAD: STEP file

Next steps for NSPCC

- Add API (defined in IODA project) so can be used as library
- Link to graphical interfaces (either OCCT modeller, or Kitware CMB)
- Link to remeshing for large deformation: replace topology optimisation.

---

---

---

---

---

---

---

---

80 / 1

## Differentiating a CAD system

Notes

Automatic Differentiation (ADOL-C) applied to OpenCascade:

- Fully differentiated (except for one module)
- Good parametrisations require effort to set up
- Parametrisation can be built to include constraints
- Acceptable cost of forward differentiated OCCT in vector mode.
- Backward (adjoint) mode developed at the moment by IODA partner Paderborn Uni.

---

---

---

---

---

---

---

---

81 / 1

## Acknowledgements

Parts of this work have been conducted within the **About Flow**, **IODA** and **Maddog** projects at Queen Mary University of London

<http://{aboutflow,ioda,maddog}.sems.qmul.ac.uk>

We have received funding from the European Union's Seventh Framework Programme and Horizon 2020 under Grant Agreement Nos. 317006, 642959, 660759.



82 / 1

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---