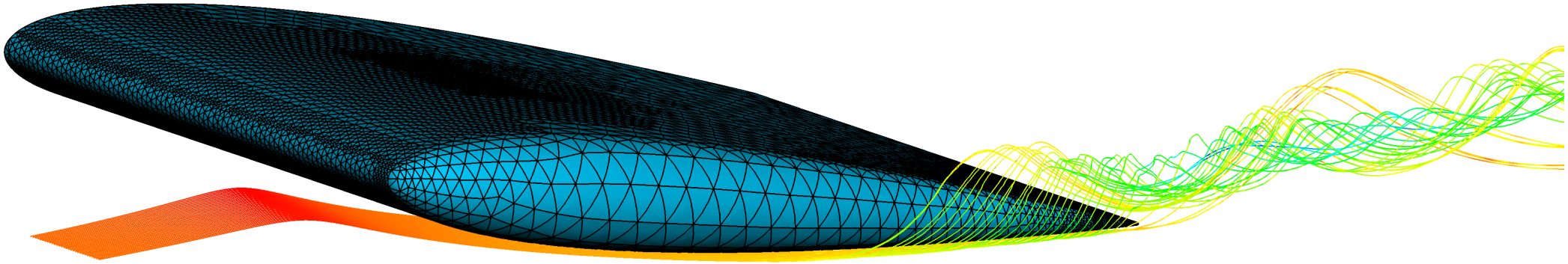
NekMesh: an unstructured high-order mesh generator for Nektar++

David Moxey, Michael Turner, Joaquim Peiró
Department of Aeronautics, Imperial College London

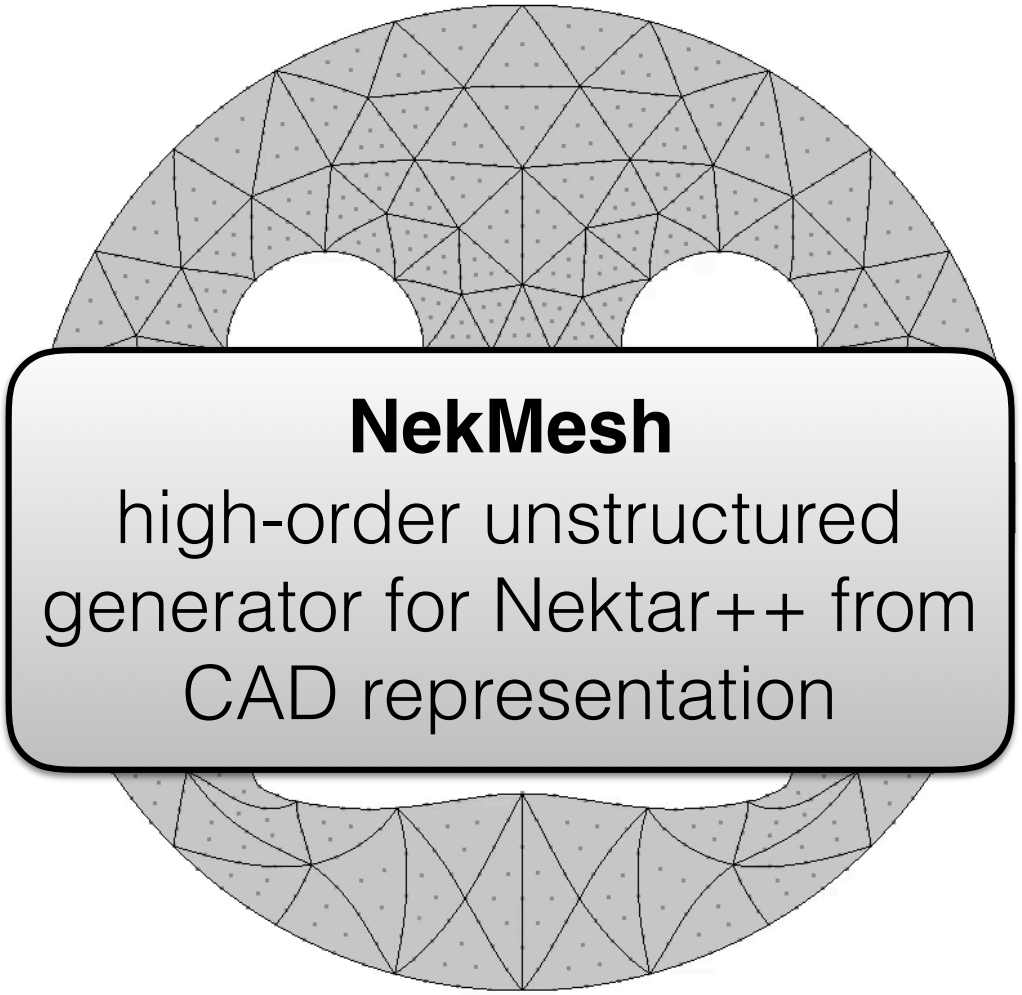
Nektar++ Workshop, London, UK
8th June 2016



Overview

- Motivation
- NekMesh
- Results and ongoing work
- Conclusions

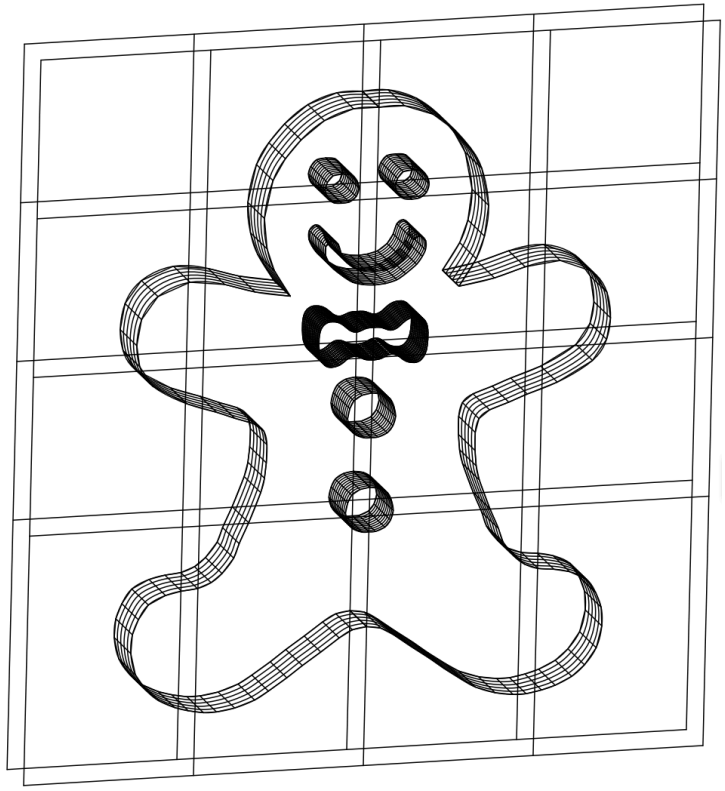
Motivation



Highly accurate representation of a user's facial expression when generating a high order mesh ($P = 5$)

High-order mesh generation

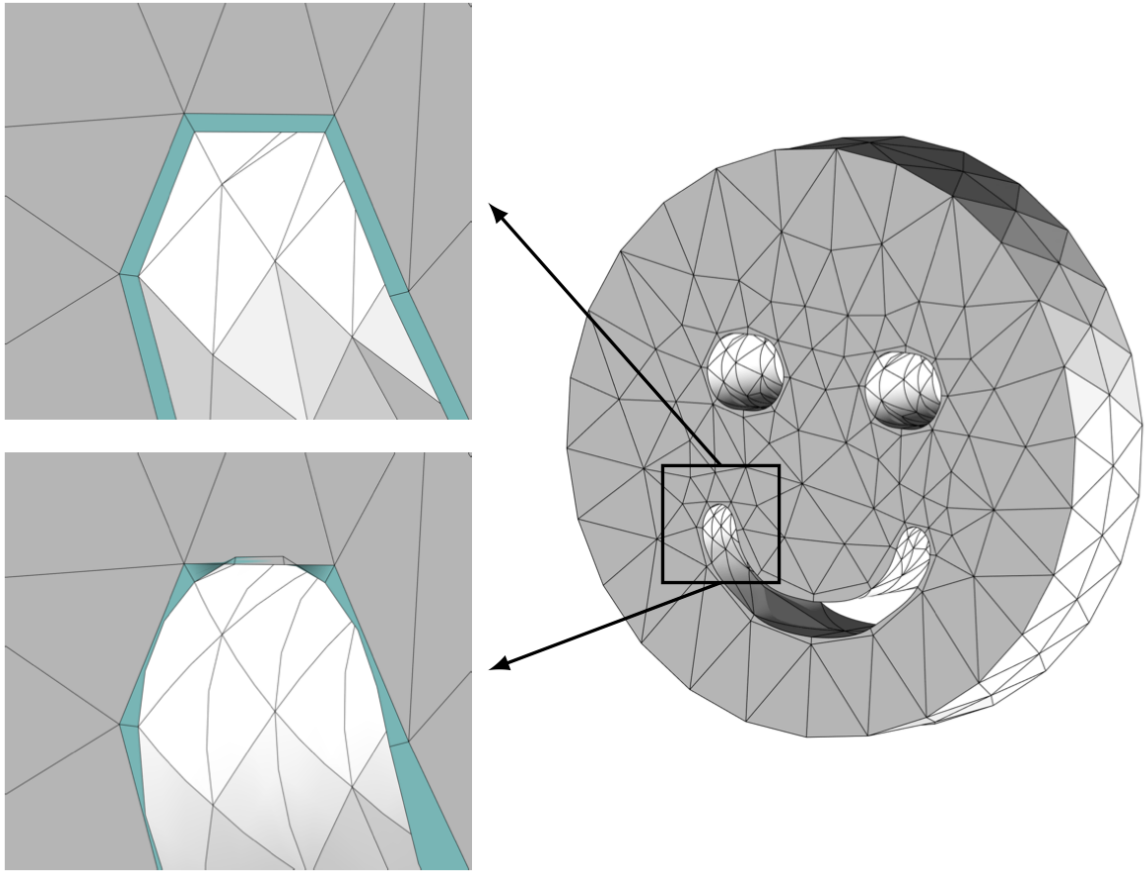
(in theory)



B-Rep

High-order mesh generation

Curving coarse meshes leads to invalid elements
Most existing MG packages cannot deal with this

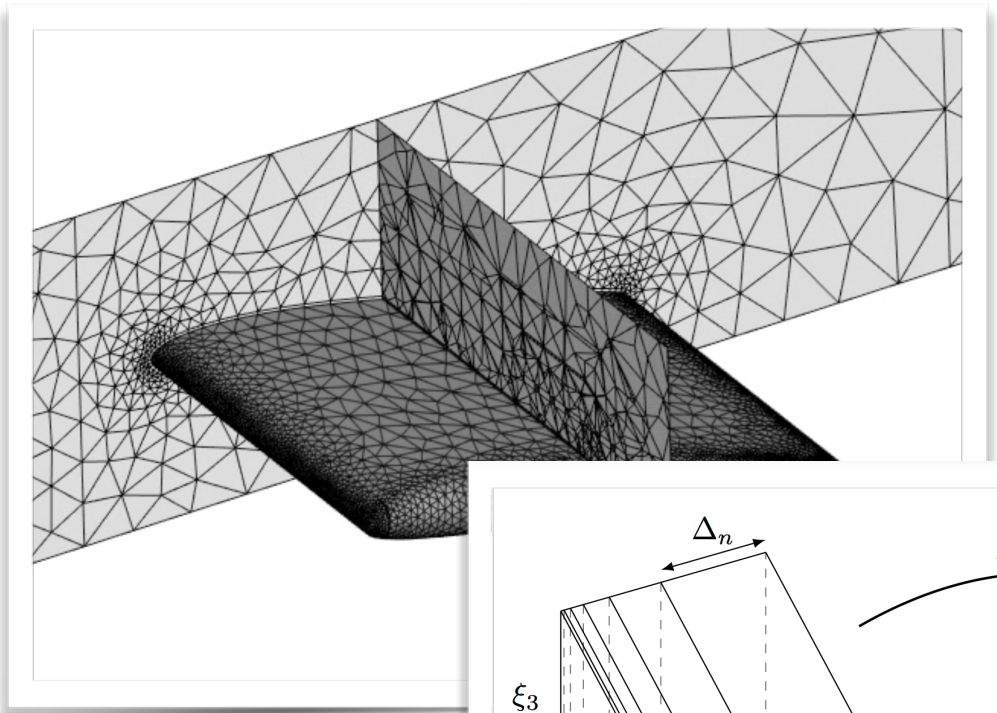


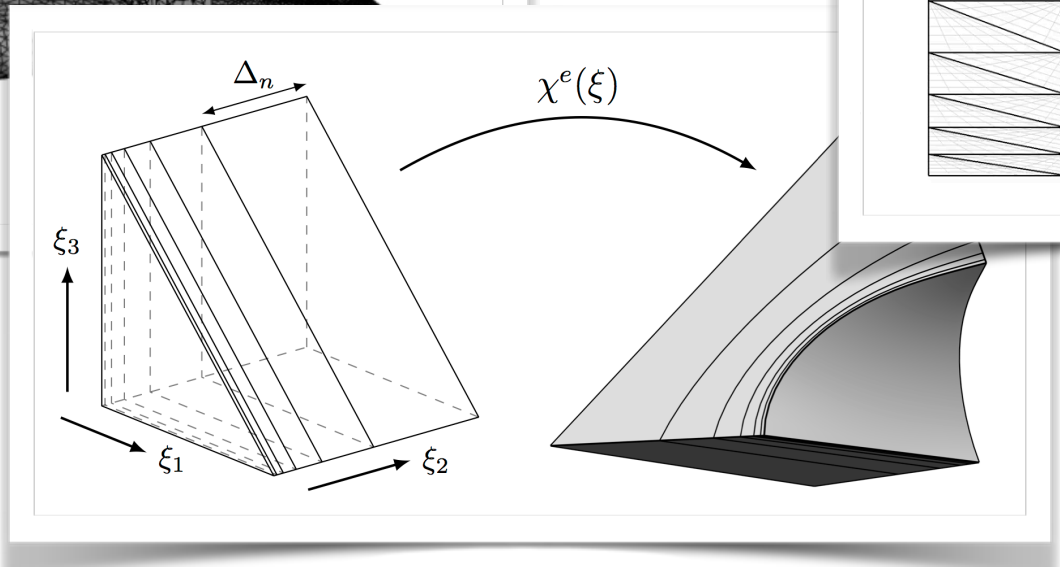
MG pipeline to date

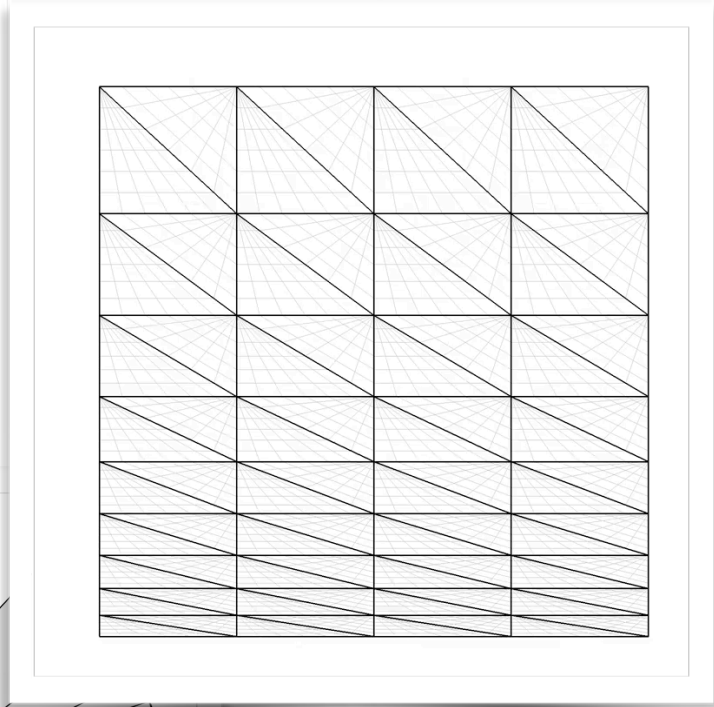
- Use commercial mesh generator for coarse straight-sided mesh (prism boundary, tet interior)
- Manipulate the mesh to make it high order
- Try to fix broken elements
- Pray
- Check whether the mesh is valid: if not, change mesh parameters and start over

Existing workflow

Linear mesh from Star-CCM+ → Convert to high-order → Output high-order mesh and/or correct





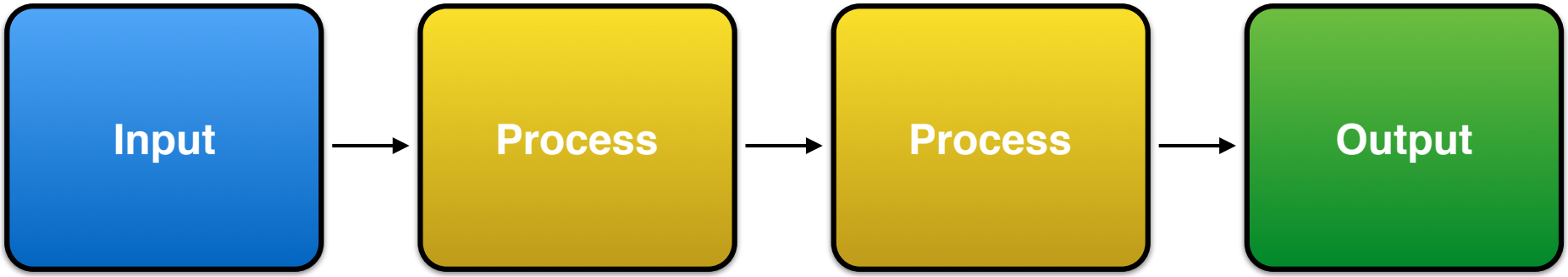


NekMesh goals

- Be capable of generating 3D curvilinear meshes of complex geometries with boundary layers
- CAD interface
- Be capable of optimising quality and/or untangling invalid meshes
- Support import from various external file formats
- Support high-order manipulation of linear meshes

Already in **MeshConvert**: **NekMesh** is an extension

Preprocessing



Nektar++
CAD
Nektar
Gmsh
Tecplot/Star
PLY
VTK
Semtex

High-order smoothing

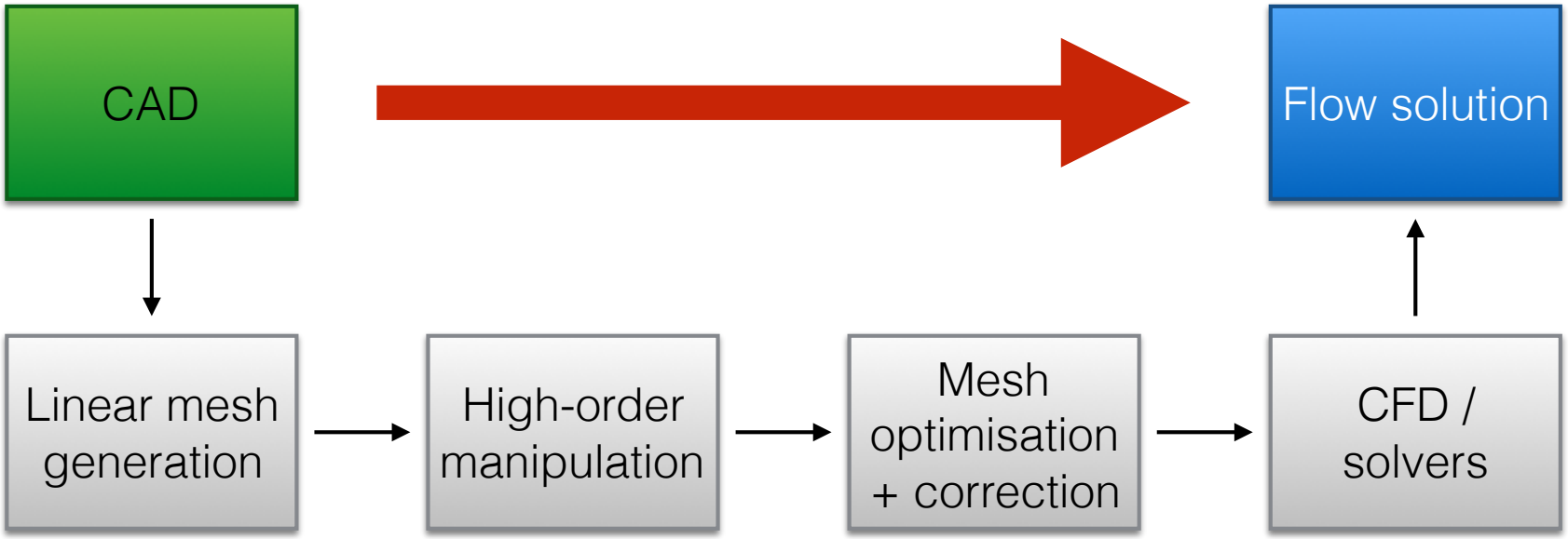
Boundary refinement

Nektar++
Gmsh
VTK (linear)

~~**MeshConvert:**~~ **NekMesh:** Utilises Nektar++ libraries with pipeline concept: makes ~~preprocessing~~ mesh generation easier

Towards a better MG solution

Single step process from CAD to flow solution
As few user parameters as possible
Preserve CAD data throughout



Linear mesh generation

- Use of heavily modified third party libraries

  ➡ Triangle and Tetgen

- Prismatic layer generation

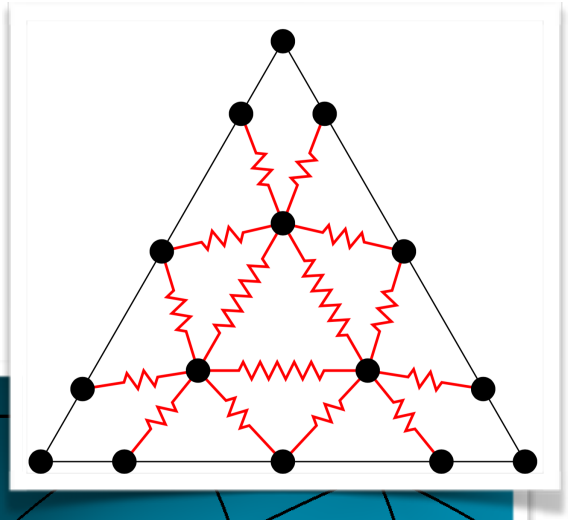
  ➡ In house code, designed for high-order

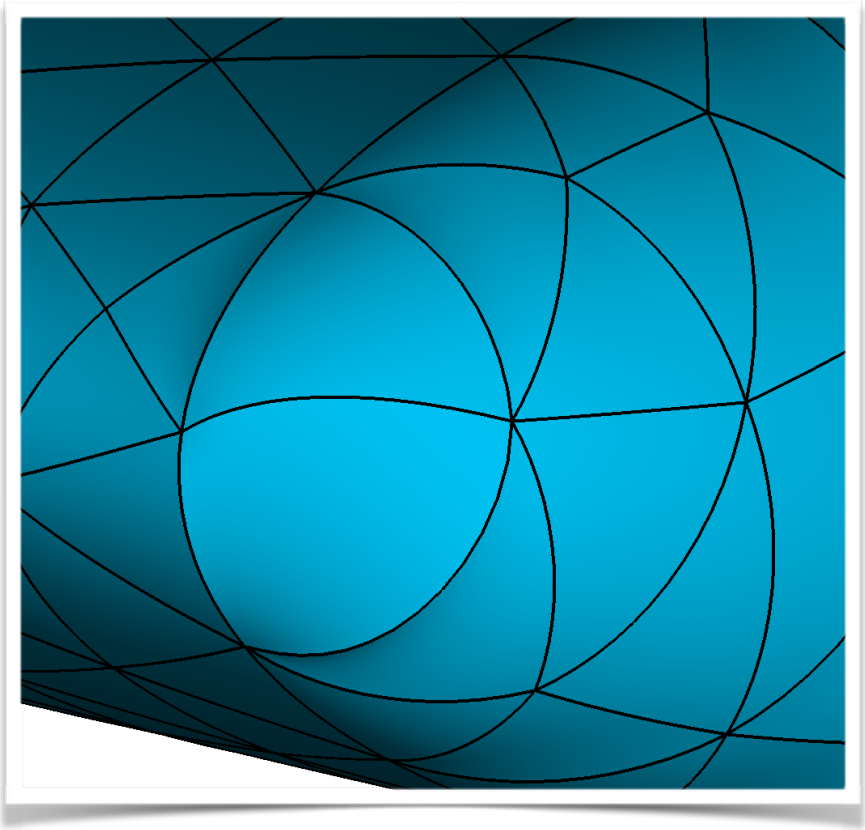
- All used with automated sizing specification for curvature-based refinement

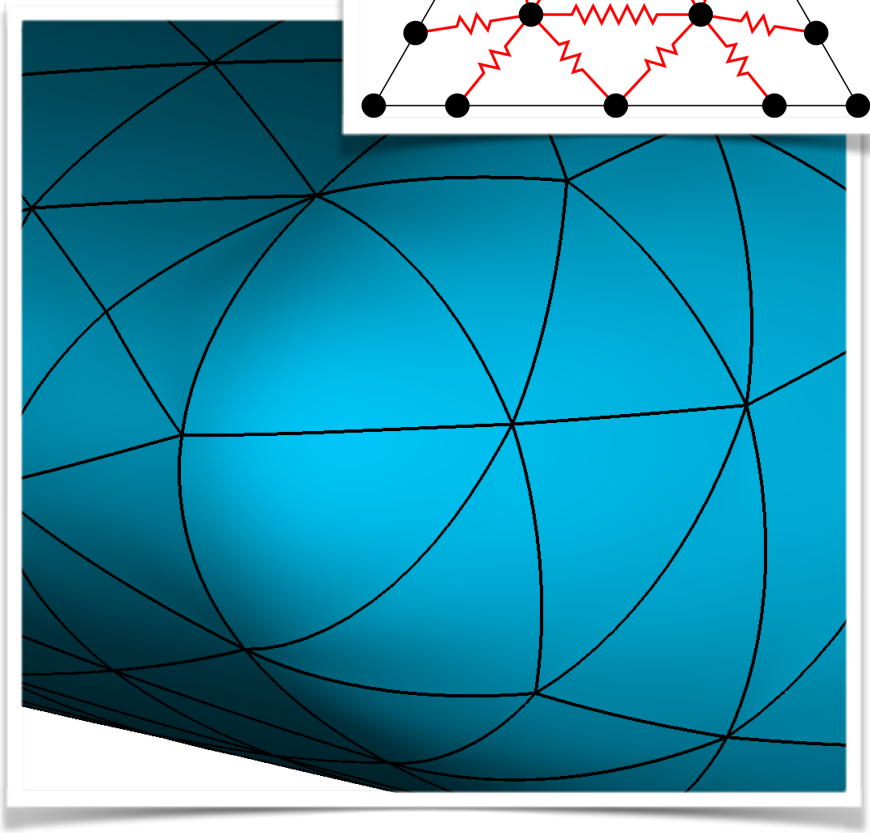
  ➡ Octree system (Turner et al. IMR 2015 research note)

High-order technologies

- Optimised high-order surface generation via spring analogy

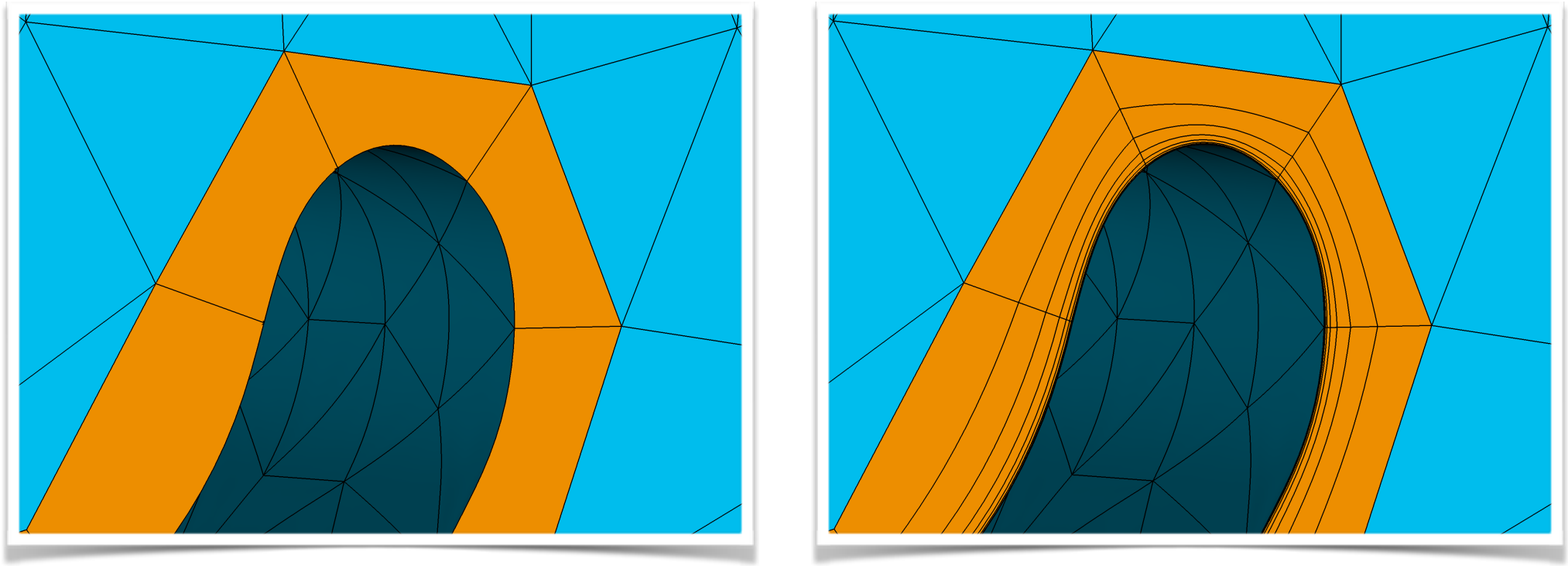






High-order technologies

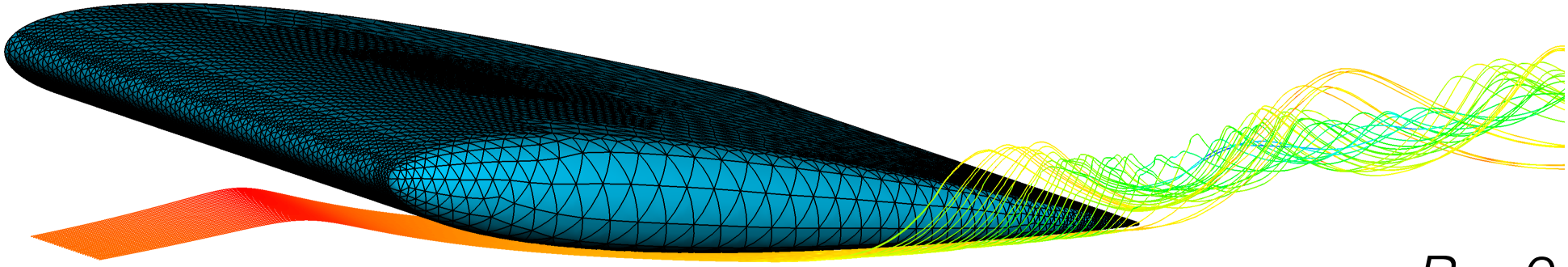
Isoparametric splitting of high-order boundary layers



Moxey et al., Comp. Meth. Appl. Mech. Eng **283** pp. 636-650 (2015)

NACA Wing

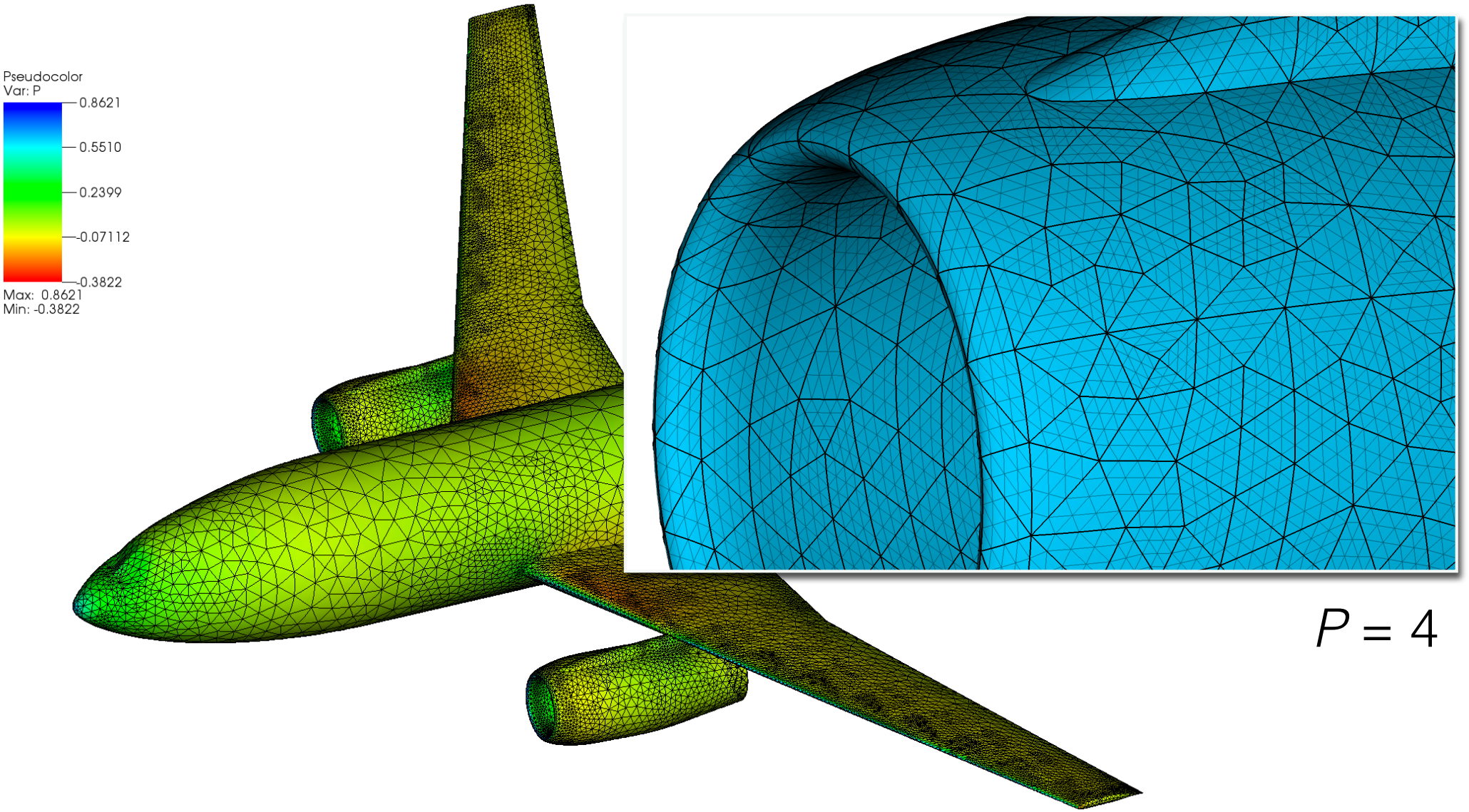
- NACA 0012, high AoA
- Prism/tet hybrid, curved BL anisotropy >1000:1
- Relevant physics



$P = 6$

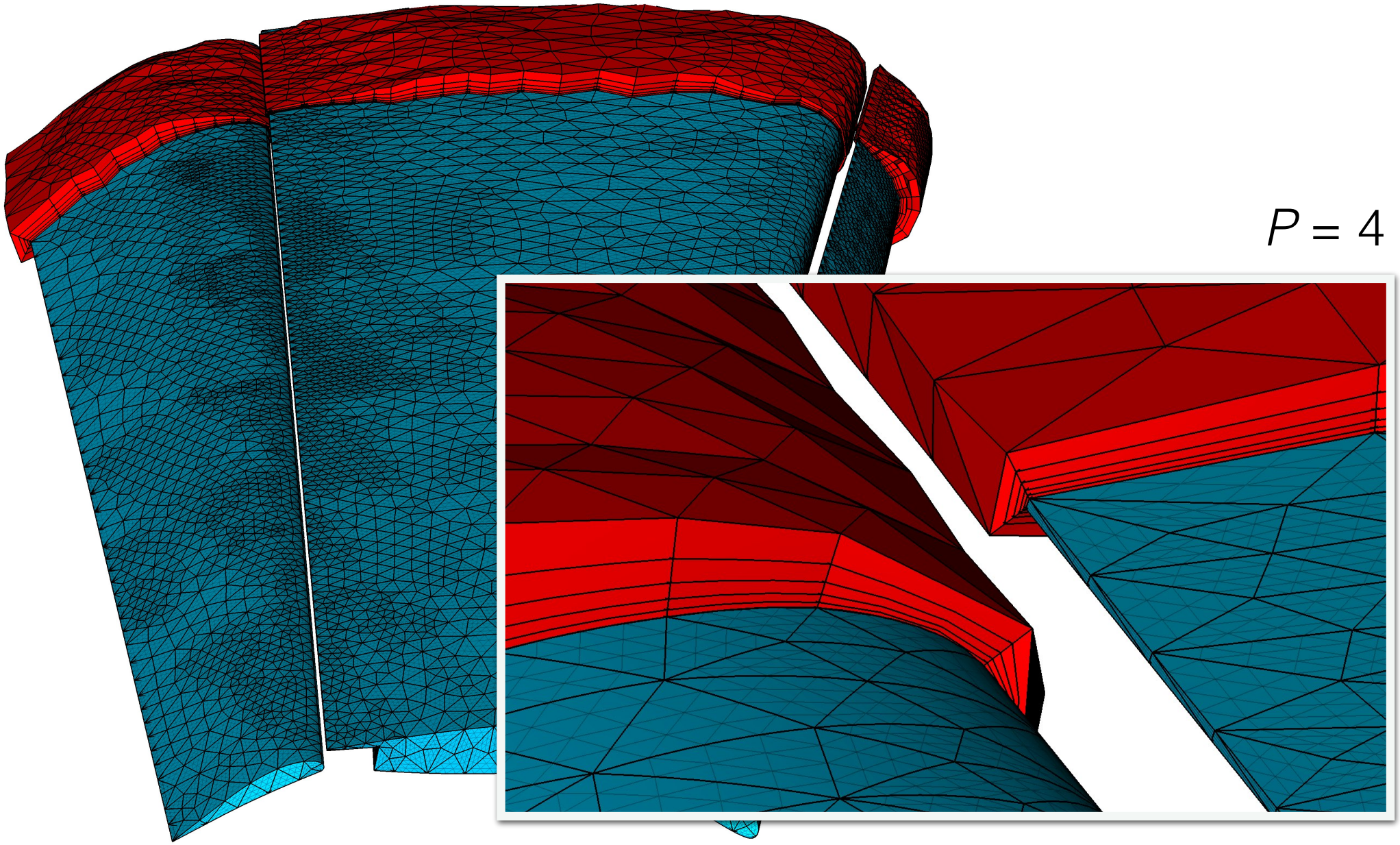
Re = 4.6M (experimental)

DLR F6

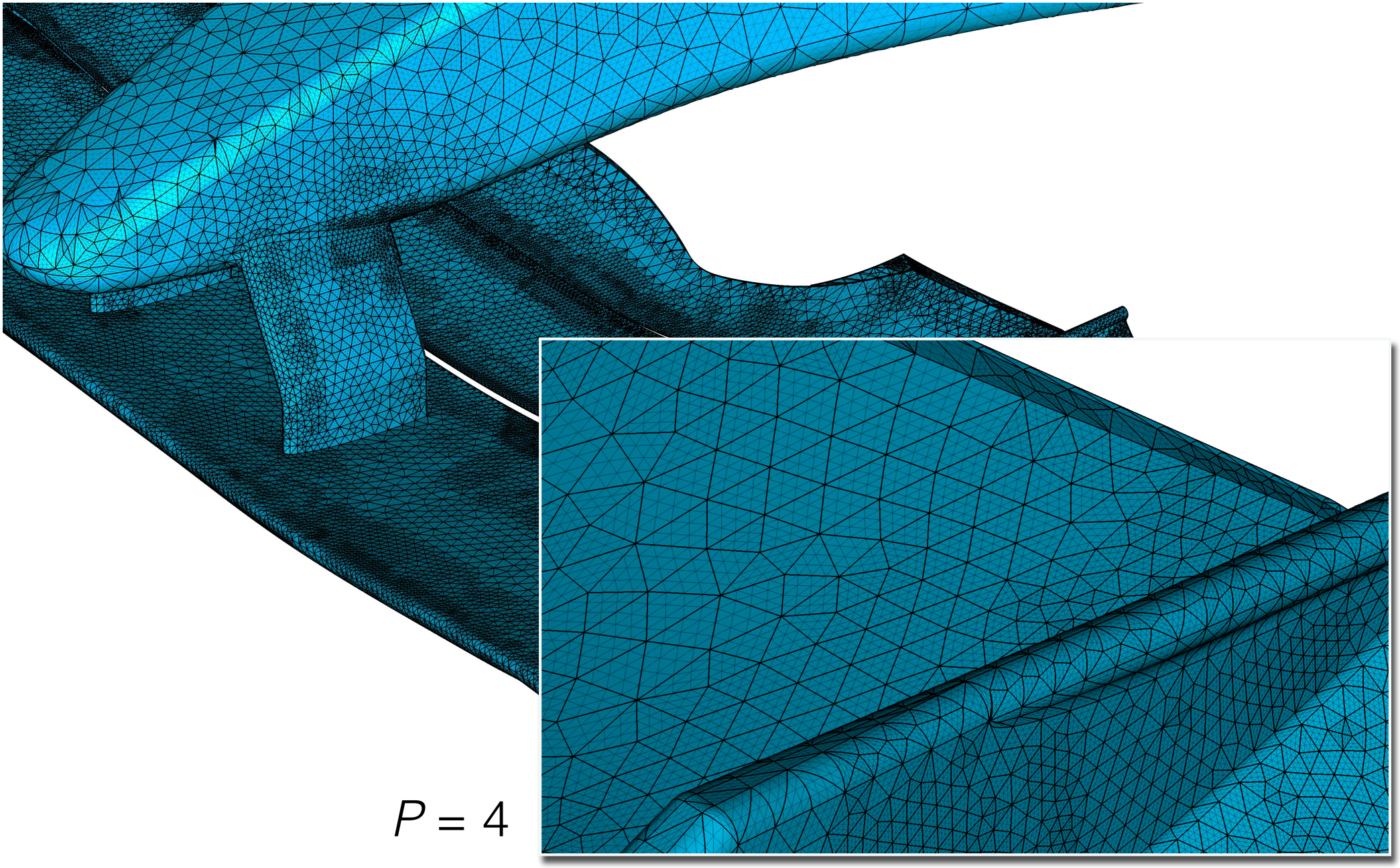

Pseudocolor
Var: P
0.8621
0.5510
0.2399
-0.07112
-0.3822
Max: 0.8621
Min: -0.3822

$P = 4$

NASA "Trap Wing"

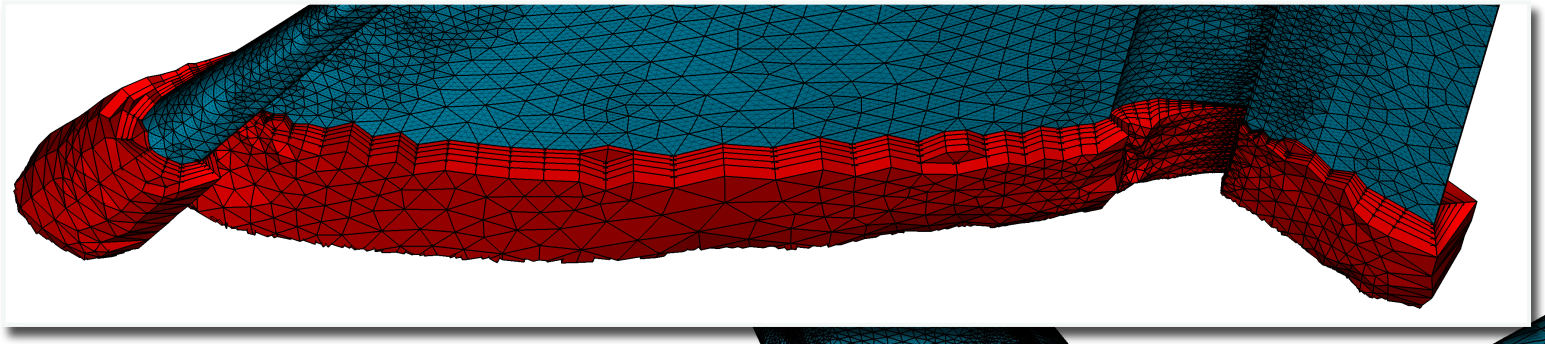


F1 Front wing

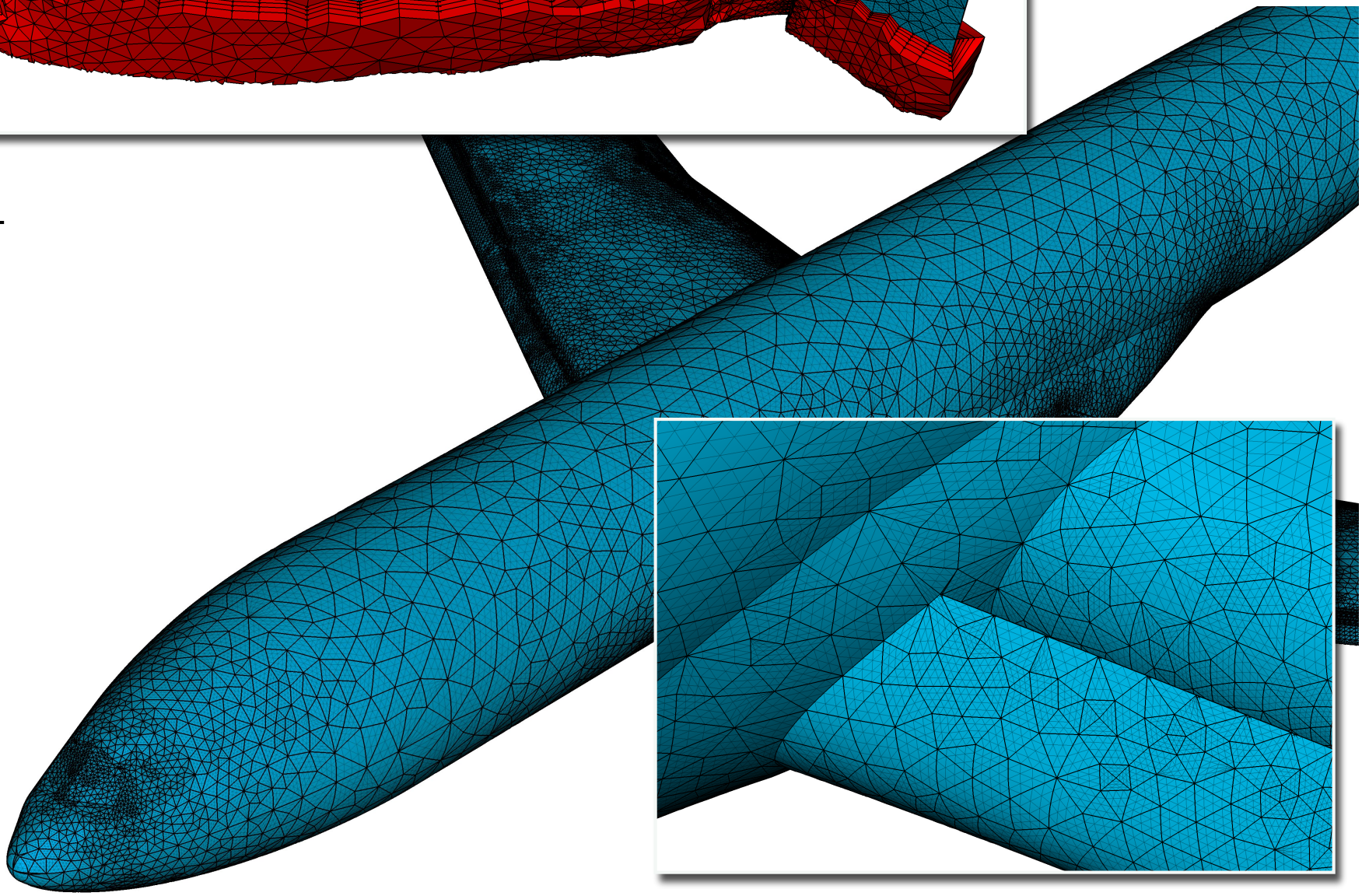


$P = 4$

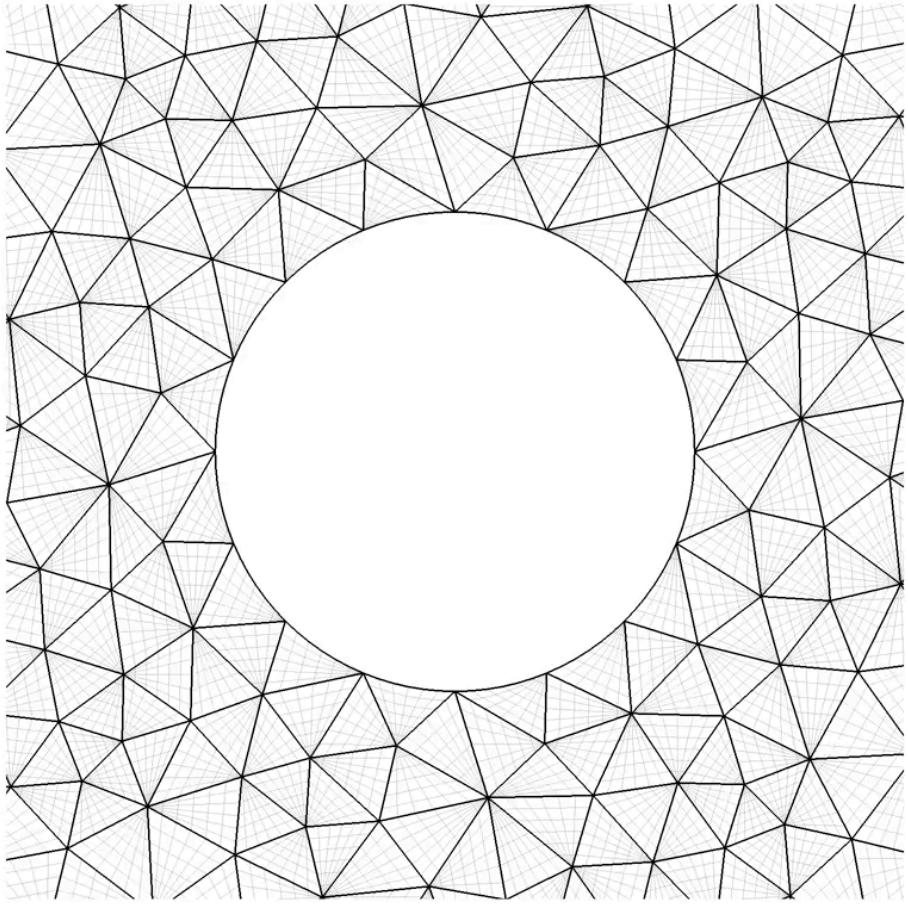
DLR F11



$P = 4$



Optimisation and untangling



Moxey et al., Comput. Aid. Des. **72** pp. 130-139 (2016)

- For optimisation we can use elasticity (elliptic PDE)

$$\nabla \cdot \mathbf{S} + \mathbf{f} = \mathbf{0} \quad \text{in } \Omega \qquad \mathbf{S} = \lambda \mathrm{Tr}(\mathbf{E})\mathbf{I} + \mu\mathbf{E}$$

$$\mathbf{u} = \mathbf{g} \quad \text{in } \partial\Omega \qquad \mathbf{E} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$$

- However this is fairly complex to implement
- See LinearElasticSolver
- Robustness? Could use nonlinear or thermoelastic

Work in progress: variational optimisation

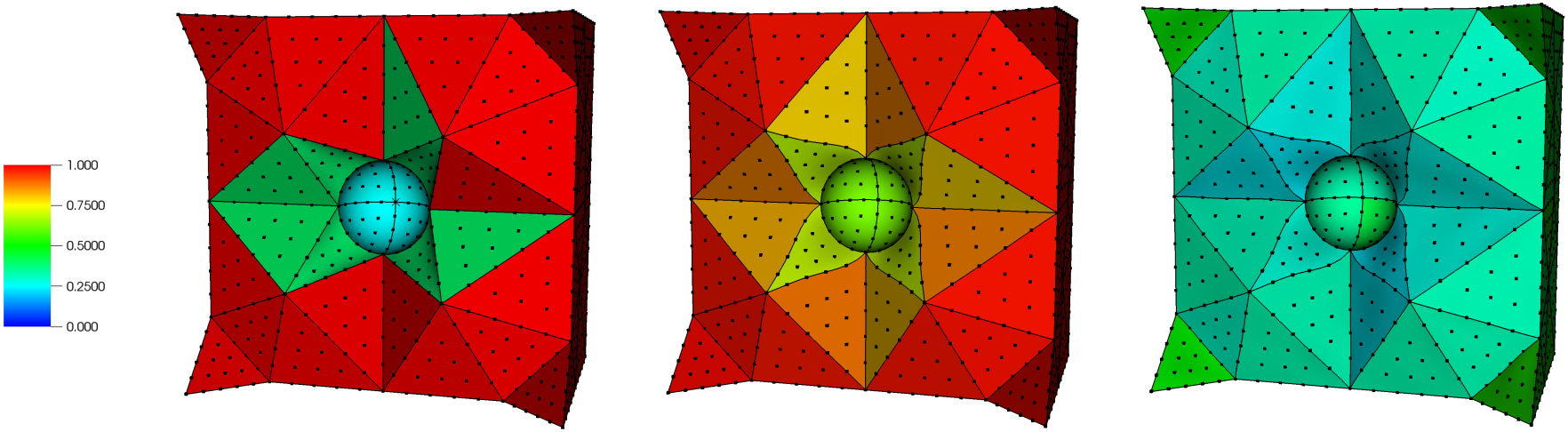
- Instead of using an elliptic operator, instead consider the variational form of the problem written as a functional

$$E(\mathbf{u}) = \int_{\Omega} W(\mathbf{u}, \nabla \mathbf{u}) d\mathbf{X}$$

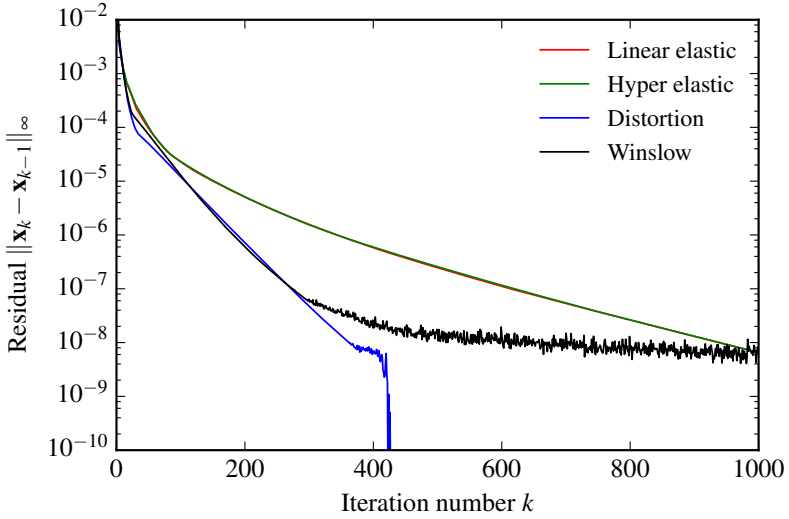
- Goal is to find the displacement $\boldsymbol{u}$ that minimises E

- Different choices of W recover existing schemes:
  - ➡ linear/nonlinear elasticity
  - ➡ Winslow equations
  - ➡ shape distortion metric

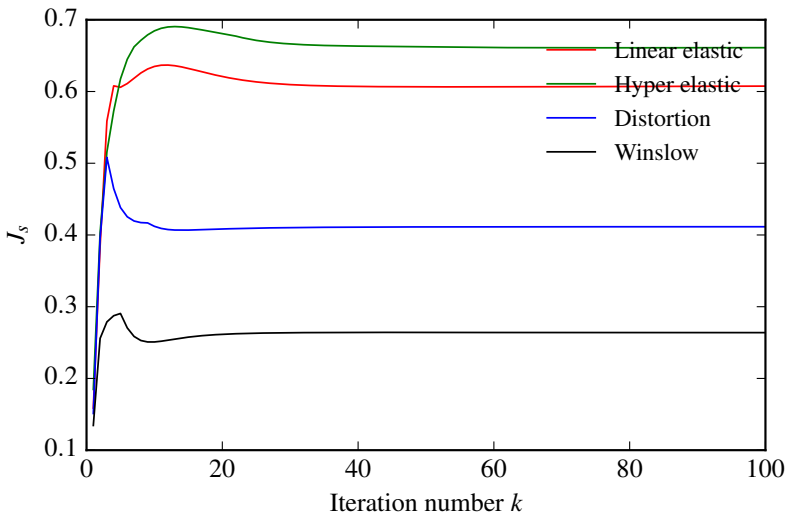
Results



(a) Original mesh (b) Optimised mesh with hyperelastic (c) Optimised mesh with Winslow



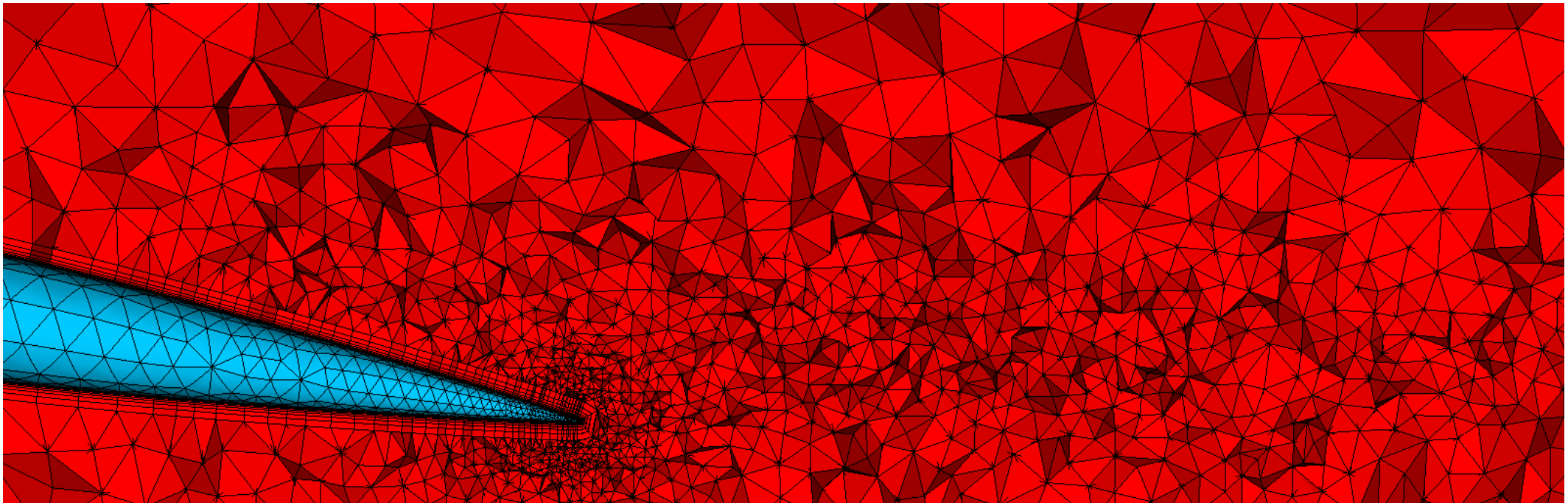
(d) Convergence of the residual



(e) Convergence of the Jacobian

Other developments

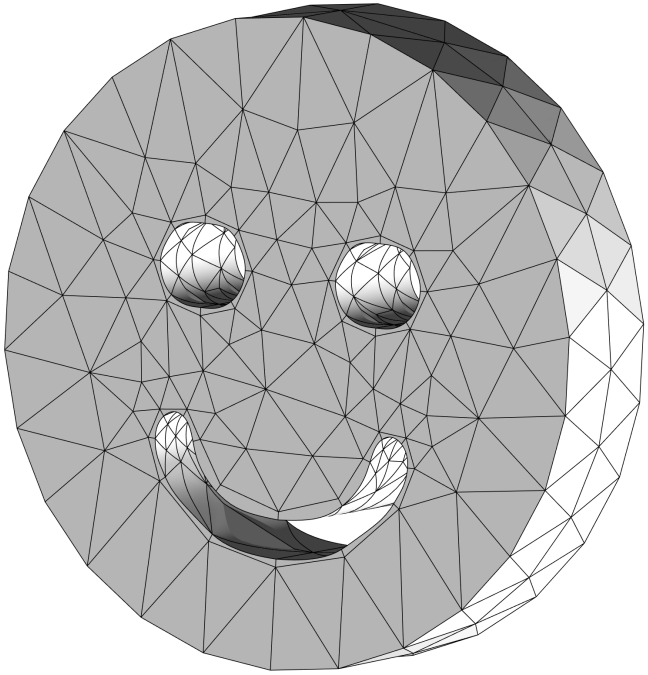
- Parallel (threading) capabilities
- Anisotropic mesh generation
- Higher automation through adaptivity



Caveats and using NekMesh

- Still relies on 'watertight' CAD
- Boundary layer generation still a work in progress
- Probably a number of fringe cases
- Ships with Nektar++ in master, release expected in v4.4, fixes/new features also merged semi-frequently
- Paper to appear in proceedings of ECCOMAS 2016
- If you think this could be useful for you then get in touch!

Thanks for listening!



@davidmoxey

d.moxey@imperial.ac.uk

www.nektar.info

Coming up...

- Now: 🍏☕🍎🍞
- After lunch, tutorials in SKEM 062 (downstairs, two floors):
  - ➡ **13:30** Installation / compilation triage
  - ➡ **14:00** ADRSolver
  - ➡ **16:00** Nekkloud
- **Tomorrow:** more tutorials in the morning, discussion session in the afternoon
- Thanks for attending!