NekMesh: an open-source high-order mesh generator

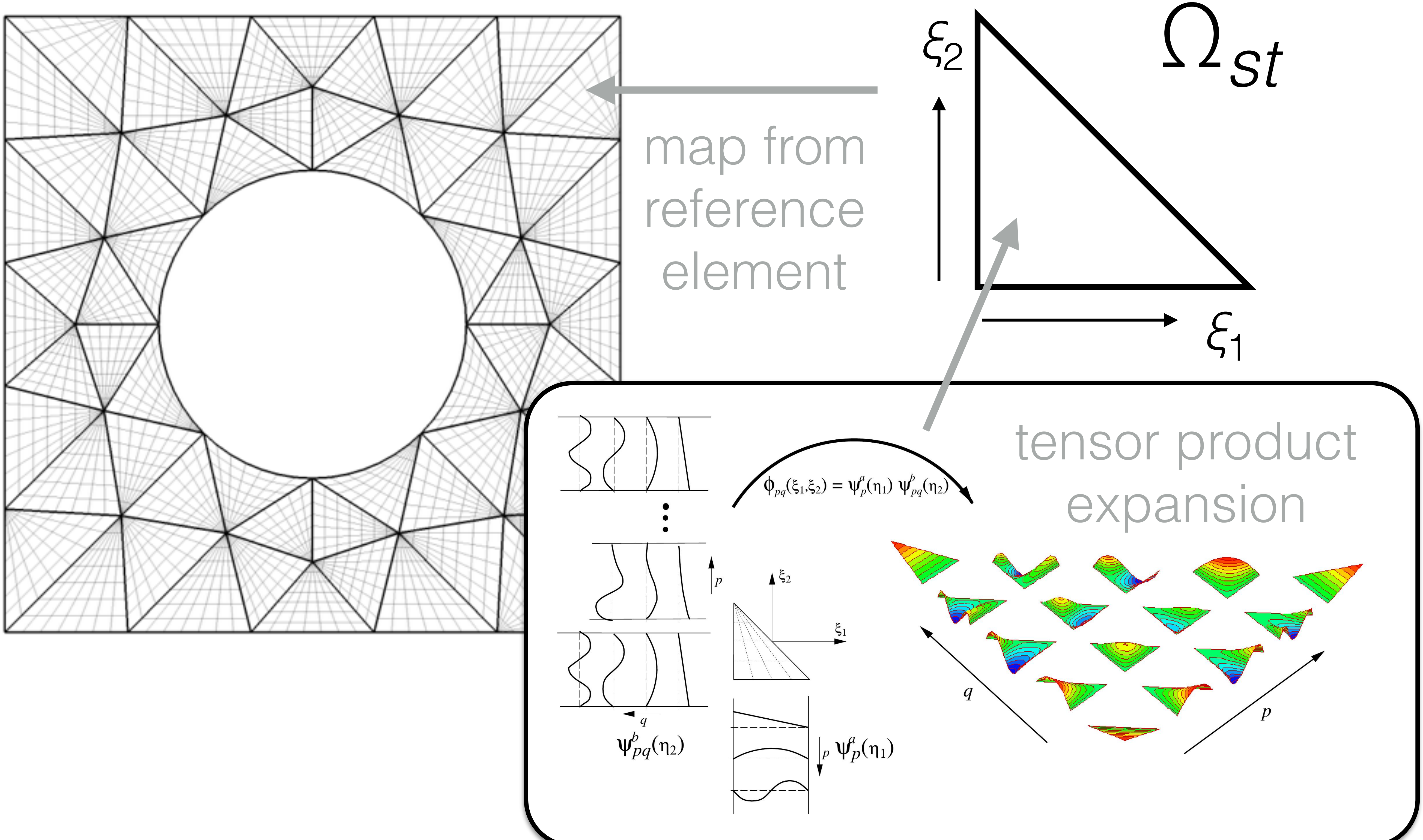
D. Moxey, M. Turner, J. Peiró
Department of Aeronautics, Imperial College London

DiPaRT 2016 Annual Meeting, Bristol, UK
22nd November 2016

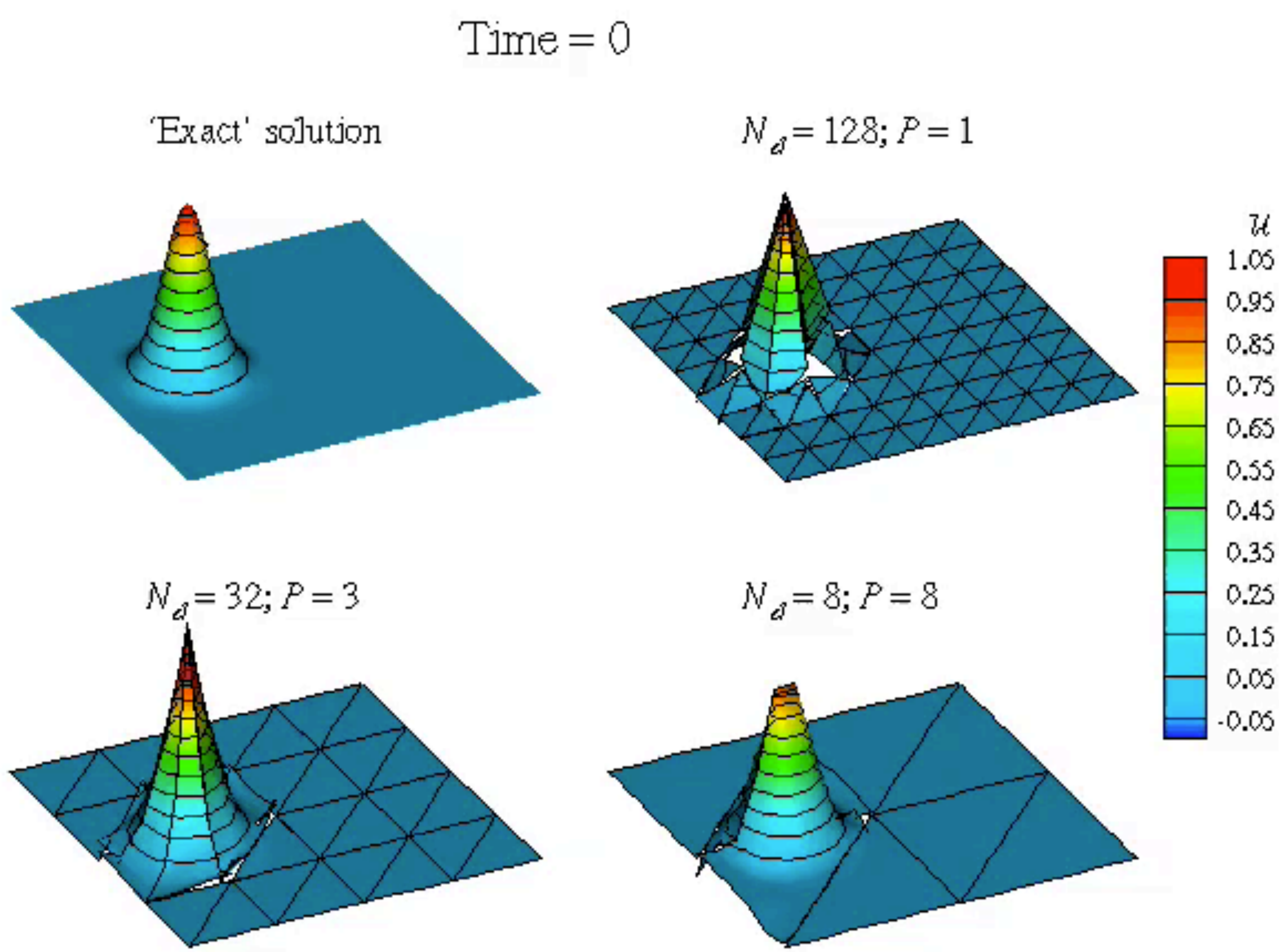
Overview

- The spectral/*hp* element method
- Challenges in high-order mesh generation
- Some results
- Conclusions

Spectral/*hp* element method



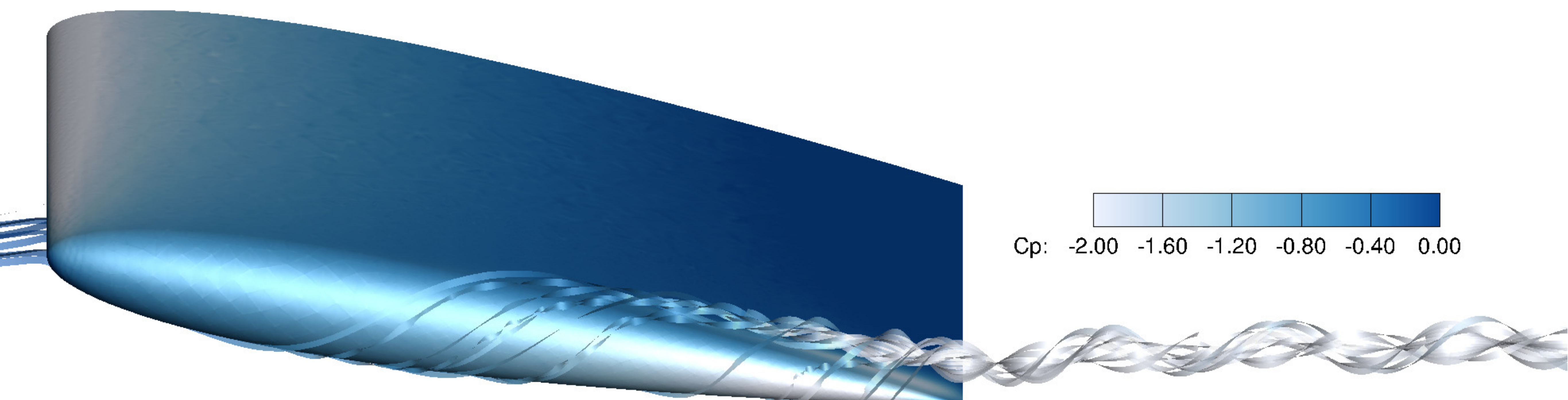
Why high-order methods?



Scalar transport of Gaussian bump
All cases: same number of degrees of freedom

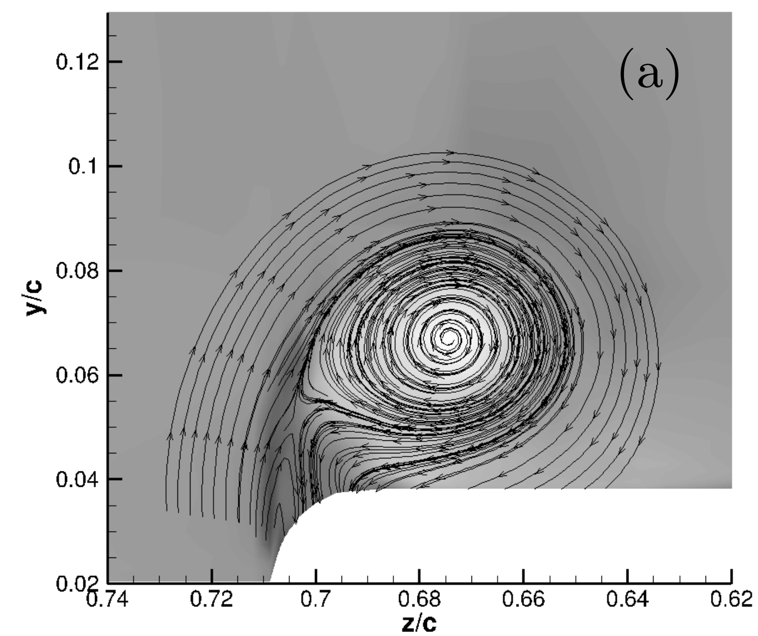
NACA 0012 example

- Simulations at Re = 1.2m (experimental 4.6m)
- Highly unsteady, vortex dominated
- SVV-LES formulation of incompressible NS

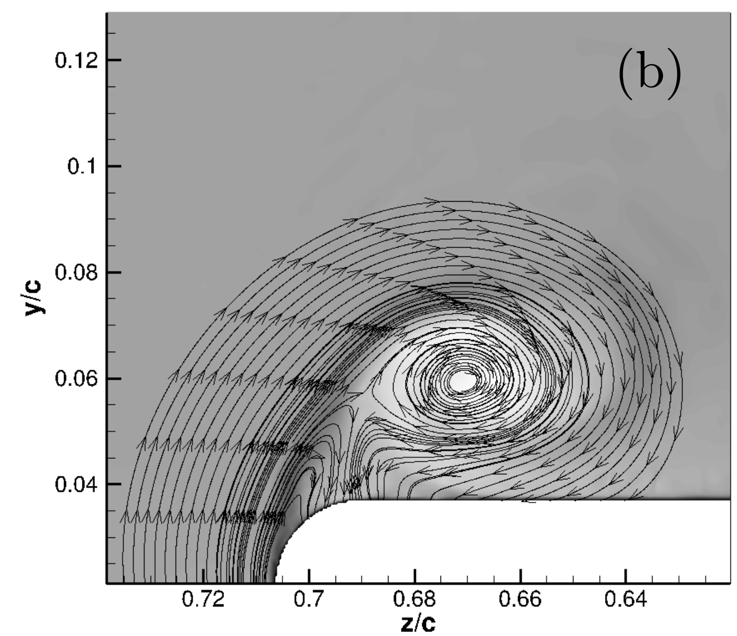


Lombard, Moxey, Hoessler, Dhandapani, Taylor and Sherwin
Implicit large-eddy simulation of a wingtip vortex, AIAA Journal **54** (2), 2016

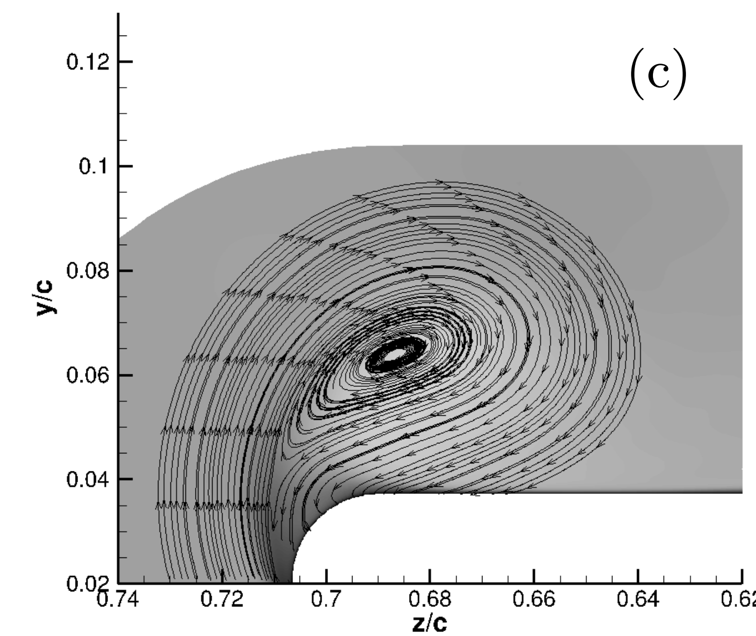
NACA 0012 example



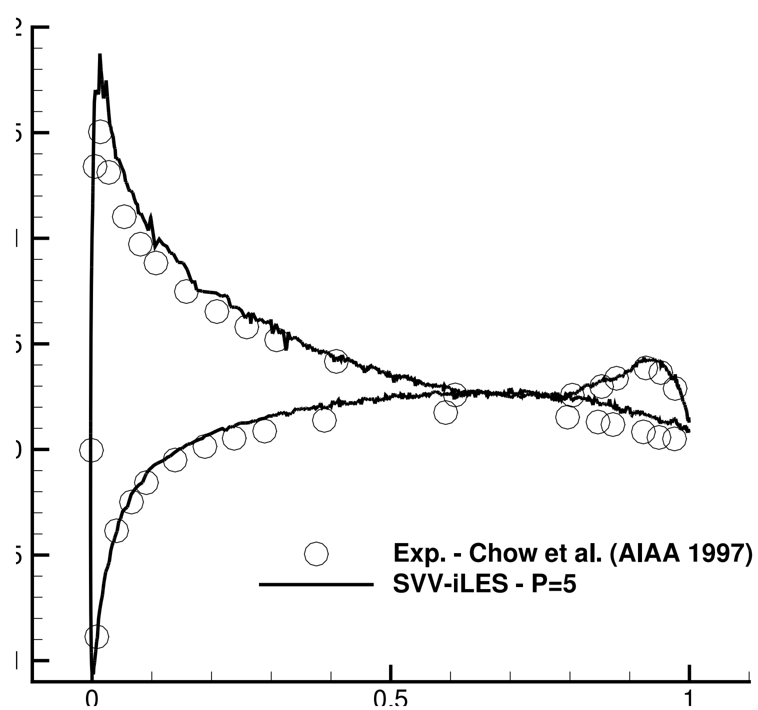
Exp.

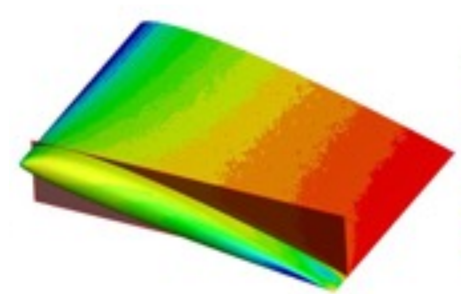


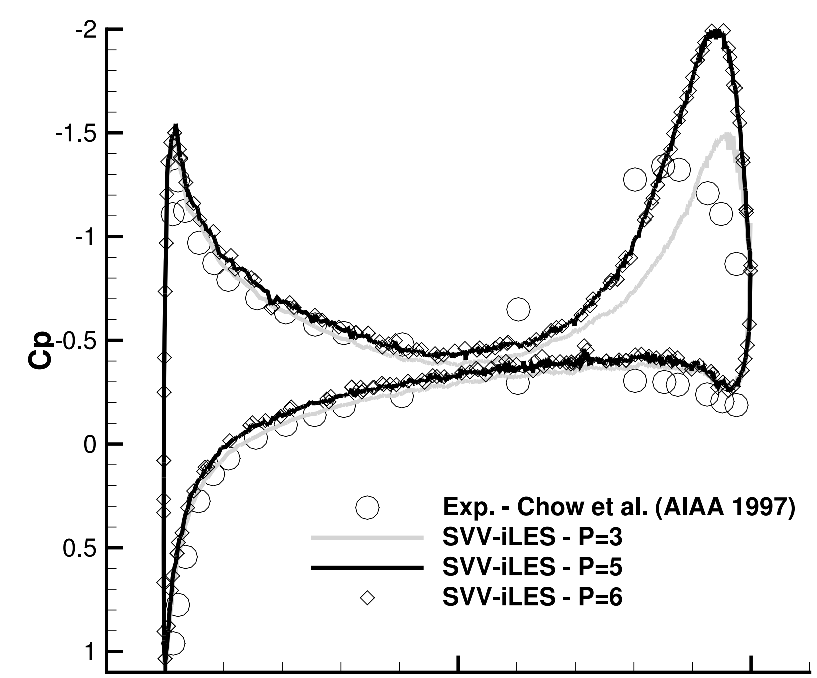
Nektar



LES. Uzun



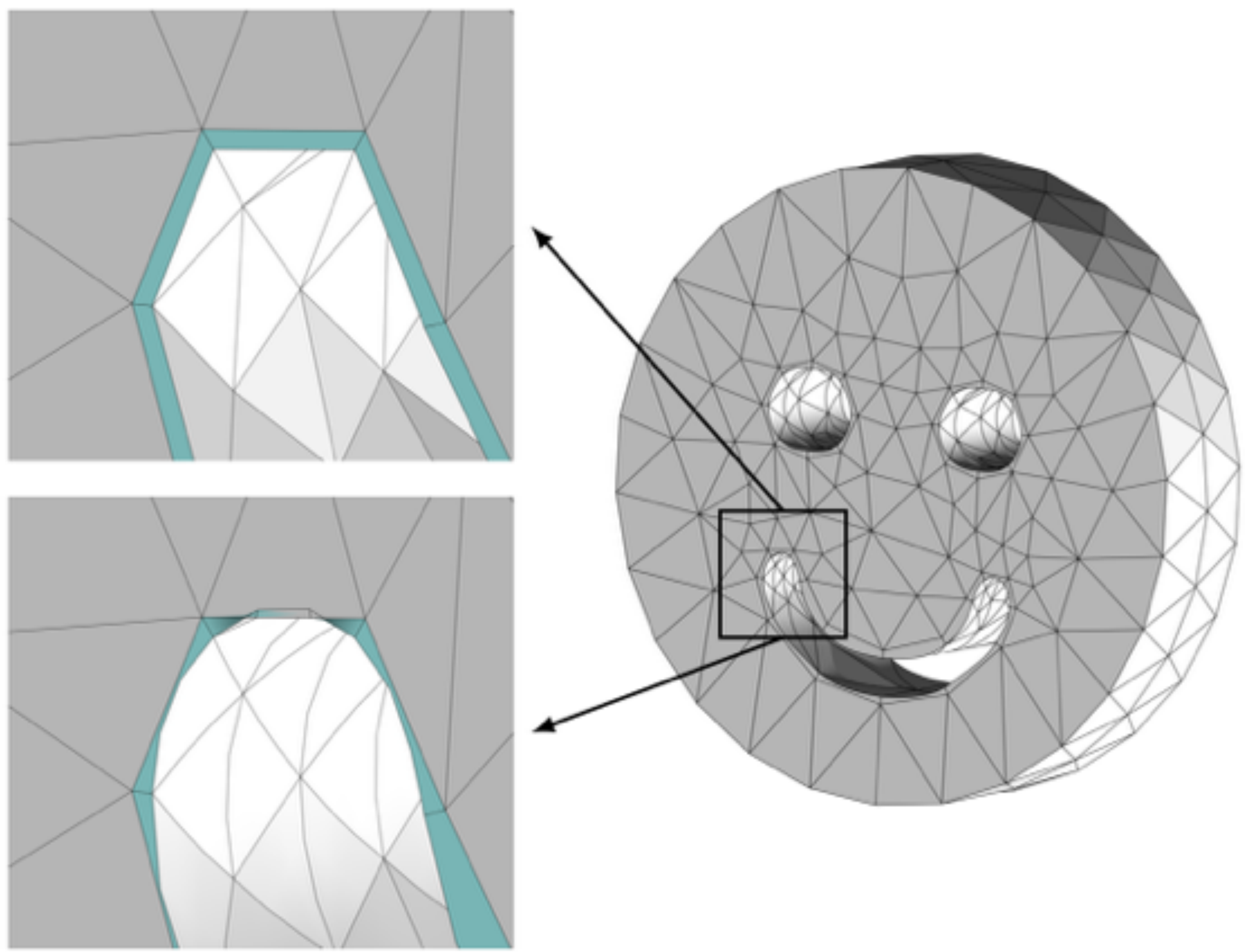




Lombard, Moxey, Hoessler, Dhandapani, Taylor and Sherwin
Implicit large-eddy simulation of a wingtip vortex, AIAA Journal **54** (2), 2016

High-order mesh generation

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



Our process / philosophy

- Keep things *modular:* each module does one thing
- Pass a common mesh between modules in a pipeline
- Focus on 3D: prismatic boundary layers + tetrahedral interior
- Try to minimise user parameters as much as possible
- Preserve CAD information throughout the process as much as possible
- Target high-order at every stage, from initial linear mesh generation to untangling/optimisation

NekMesh: workflows & modules

Input → **Surface mesh generation** → **Boundary layer generation** → **Volume mesh generation** → **Correction & optimisation** → **Output**

Input
- OpenCascade
- CFI (CADFix)
- Many others (Gmsh, Star-CCM+)

Surface mesh generation
- In-house code based on octree + triangle
- CFI (CADFix)

Boundary layer generation
- In-house code
- Star-CCM+

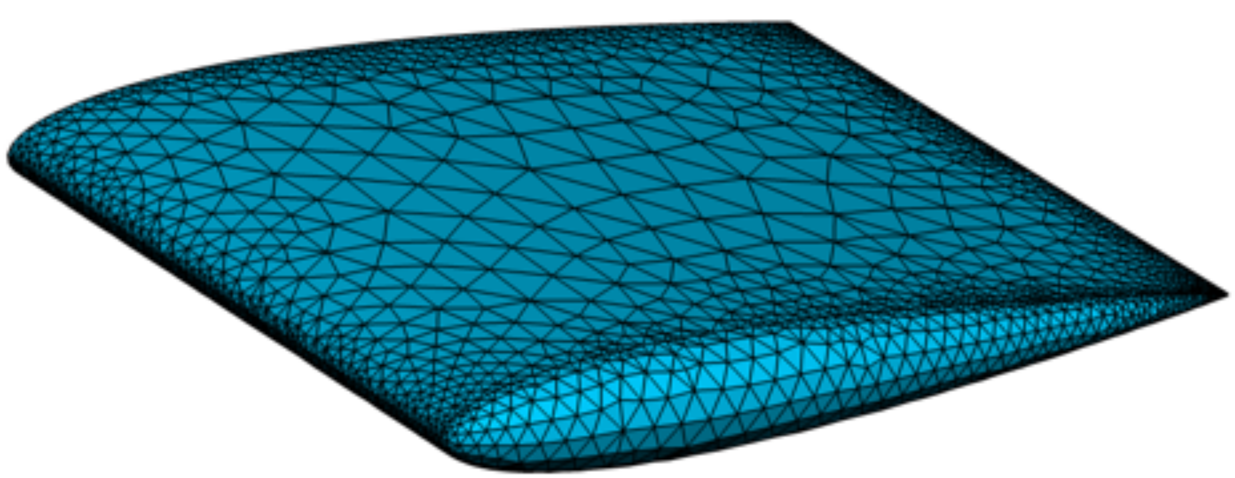
Volume mesh generation
- In-house code
- Star-CCM+
- CFI (CADFix)

Correction & optimisation
- In-house tools

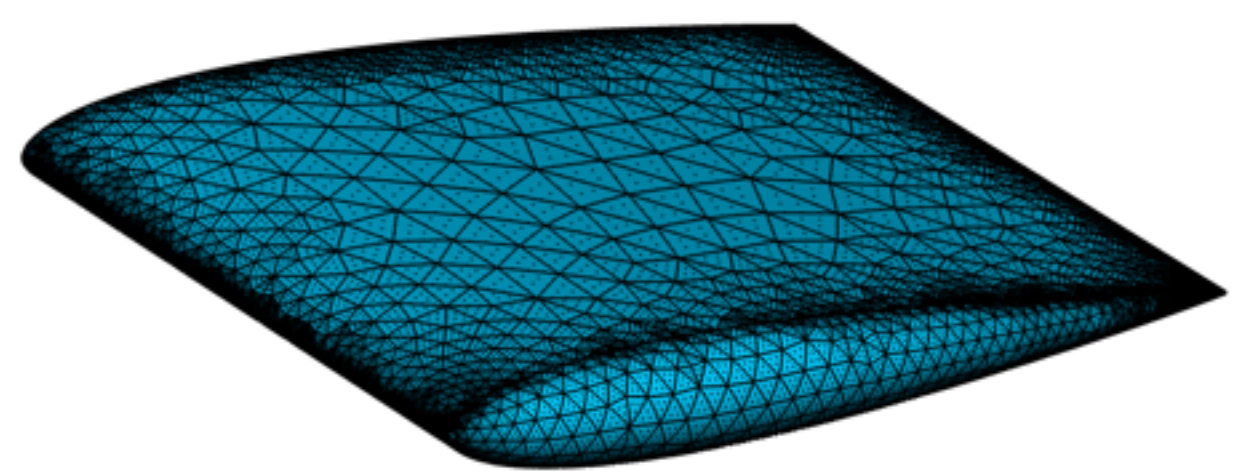
Output
- Nektar++ format
- Gmsh format
- +others?

Alternative CAD engines

- OpenCASCADE and own in-house mesh generation is quite good but very complex cases can be difficult
- For industrial cases we have integrated CFI CAD engine + linear mesher: can now use this to generate high-order meshes and help with CAD issues



Linear mesh from CFI

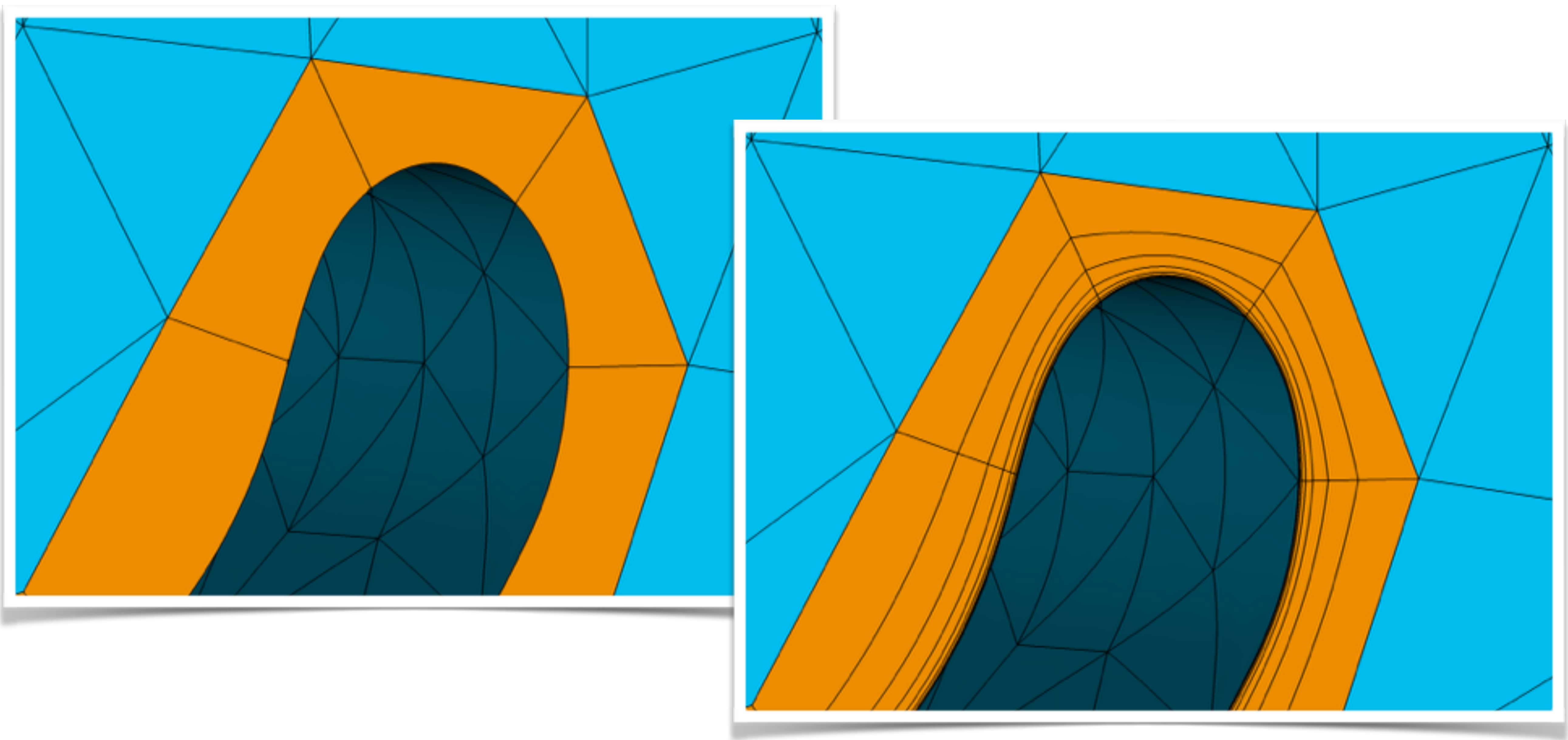


Converted to high-order by NekMesh

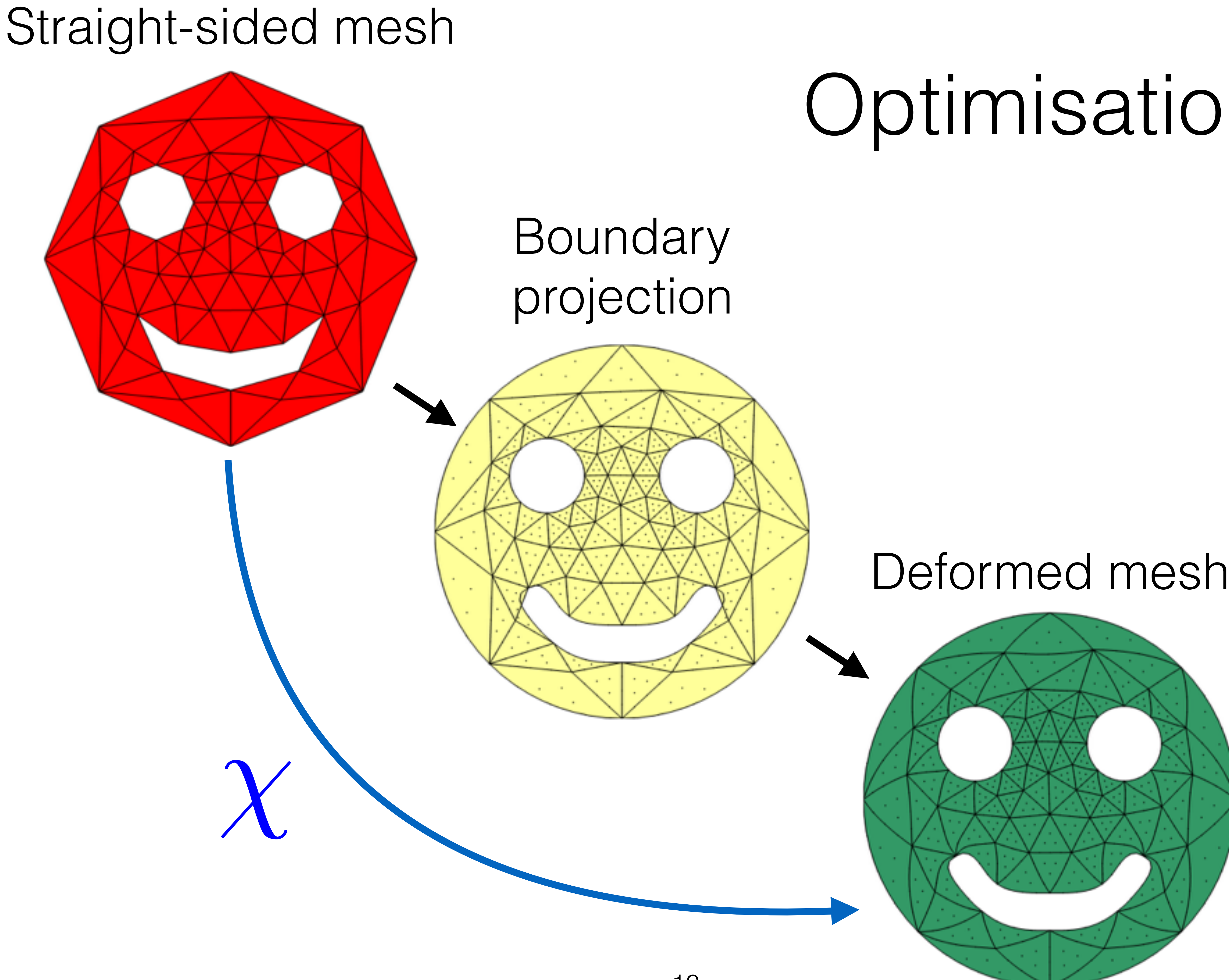
High-order technologies

Isoparametric splitting of high-order boundary layers

$y^+ < 1$



Optimisation



Current approaches

PDE solutions

- Non-linear elasticity (Persson & Peraire 2009)
- Linear elasticity (Xie et al 2013; Hartmann & Leicht 2015)
- Thermo-elasticity (Moxey et al 2015)
- Winslow (Fortunato & Persson 2016)

Direct optimisation

- Log barrier optimisation (Toulorge et al 2013)
- Distortion metric (Roca et al 2014)

Note: list of references is not exhaustive

Variational approach

We borrow ideas from (linear) variational grid generation where the mapping problem is cast as:

$$\text{Find } \min_{\chi} \mathcal{E}(\chi) \qquad \mathcal{E}(\chi) = \int_{\Omega} W(\chi, \nabla\chi)\, \mathrm{d}\mathbf{y}$$

Through an appropriate choice of W we encompass the PDE and optimisation methods in a single framework

M. Turner, J. Peiró, D. Moxey, *A variational framework for high-order mesh generation*, 25th International Meshing Roundtable, Washington DC, 2016.

Choice of functional

$$\mathbf{F} = \nabla\chi \qquad J = \det\mathbf{F}$$

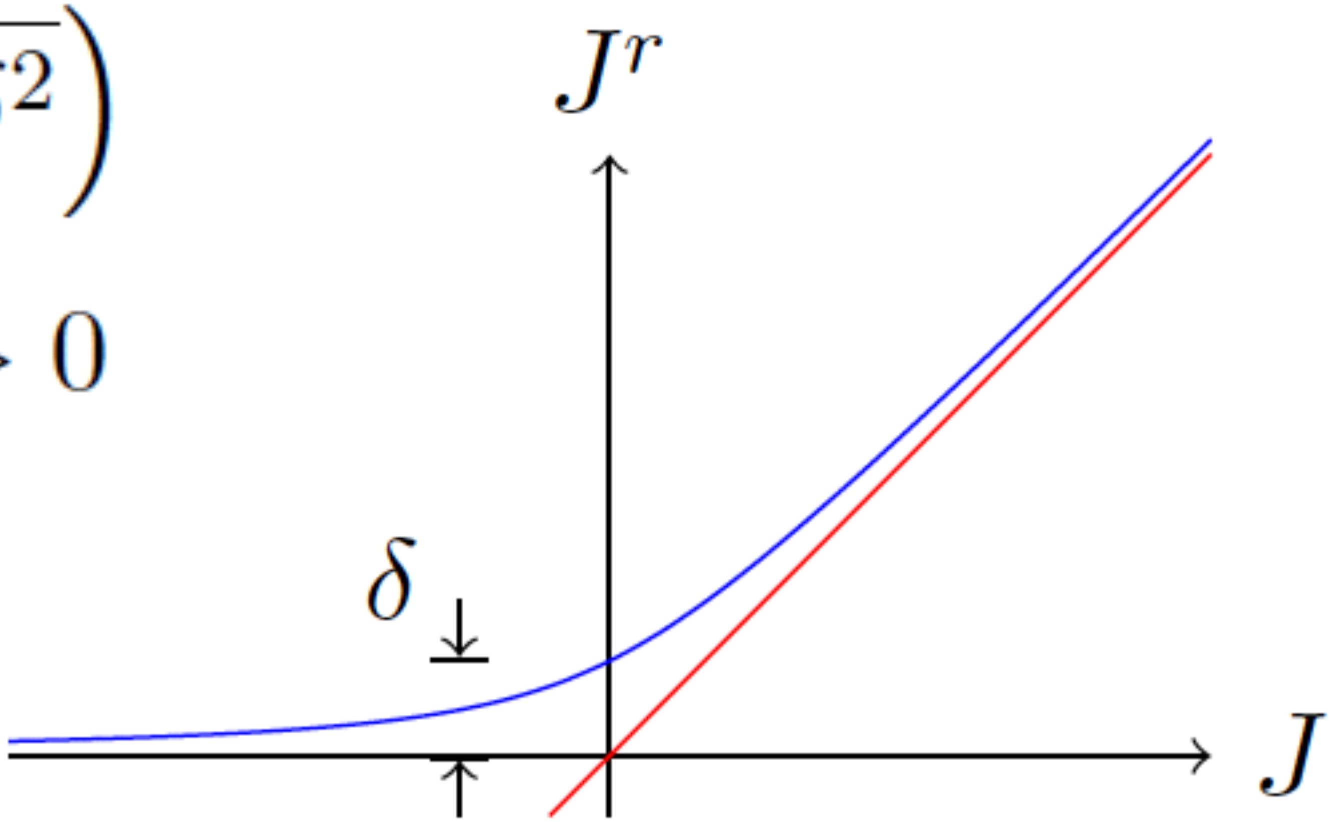
- Linear elasticity: $W = \dfrac{\kappa}{2}(\ln J)^2 + \mu\,\mathbf{E}:\mathbf{E}; \quad \mathbf{E} = \dfrac{1}{2}(\mathbf{F}^t\mathbf{F} - \mathbf{I})$
- Non-linear elasticity: $W = \dfrac{\mu}{2}(\mathbf{F}:\mathbf{F} - 3) - \mu\ln J + \dfrac{\lambda}{2}(\ln J)^2$
- Winslow: $W = J^{-1}\,(\mathbf{F}:\mathbf{F})$
- Distortion: $W = \dfrac{1}{d}|J|^{-d/2}(\mathbf{F}:\mathbf{F})$

Invalid mesh: min J_s <0

- Potentially $\mathbf{W}$ is not physical: e.g. $1/J$, $\log(J)$
- Regularisation (Garanzha 2004) which forces a positive small Jacobian:

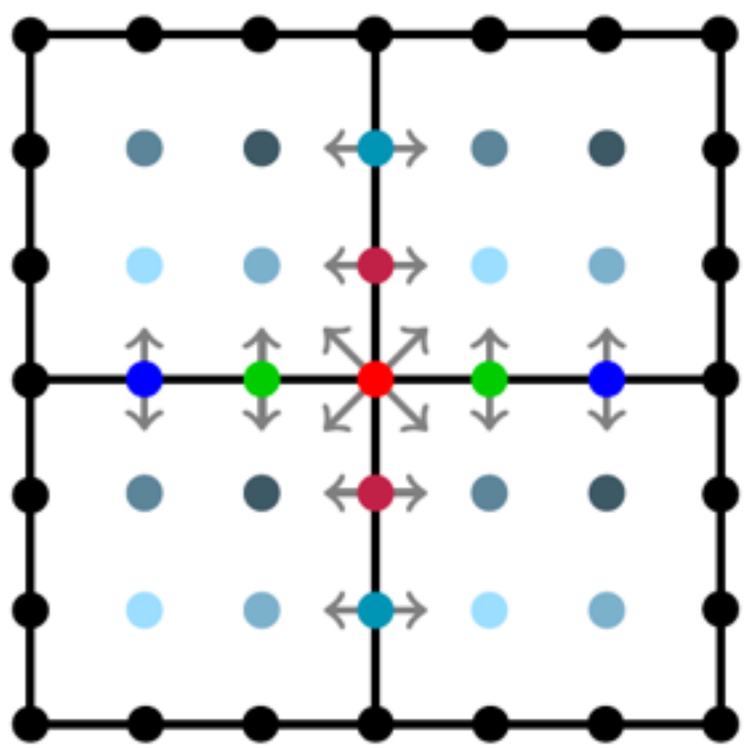
$$J^r = \frac{1}{2}\left(J + \sqrt{J^2 + 4\delta^2}\right)$$

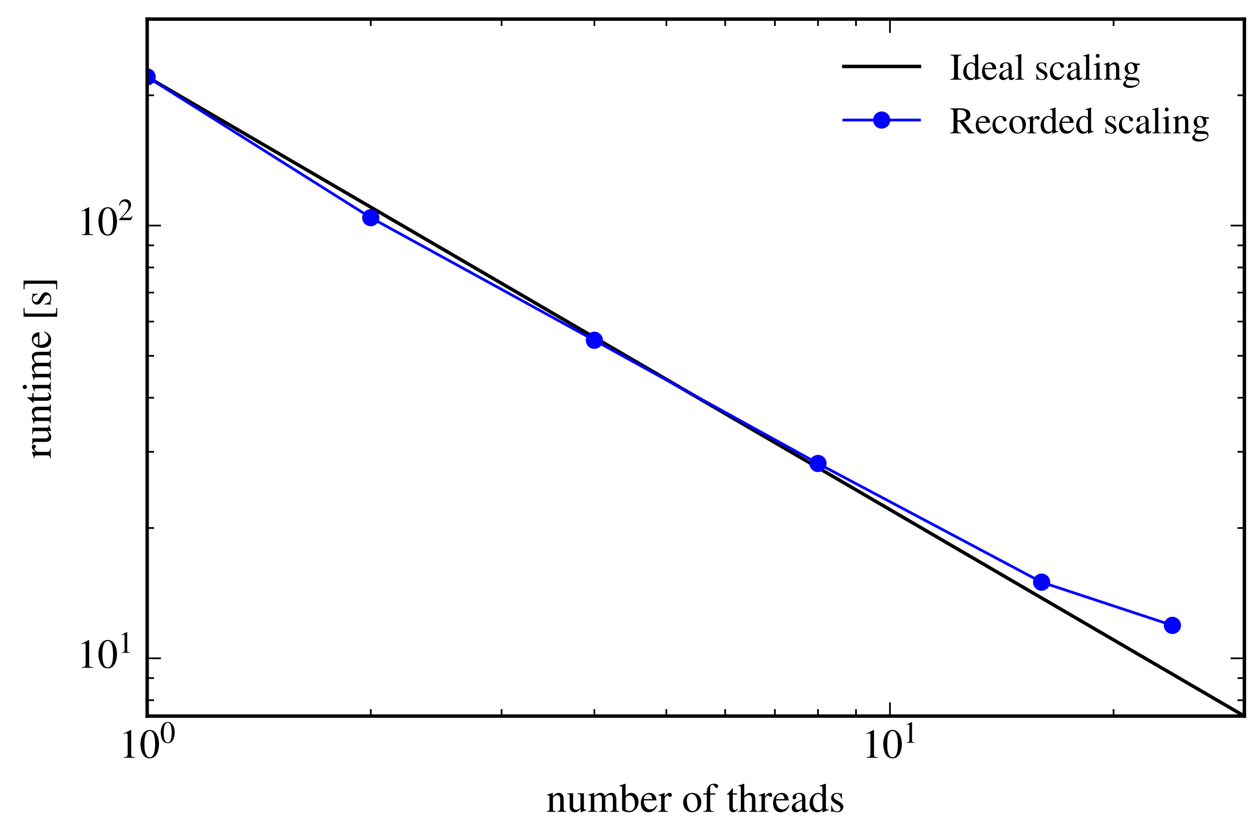
$$\delta(J): \quad \delta \Rightarrow \varepsilon \quad J \Rightarrow > 0$$



Numerical implementation

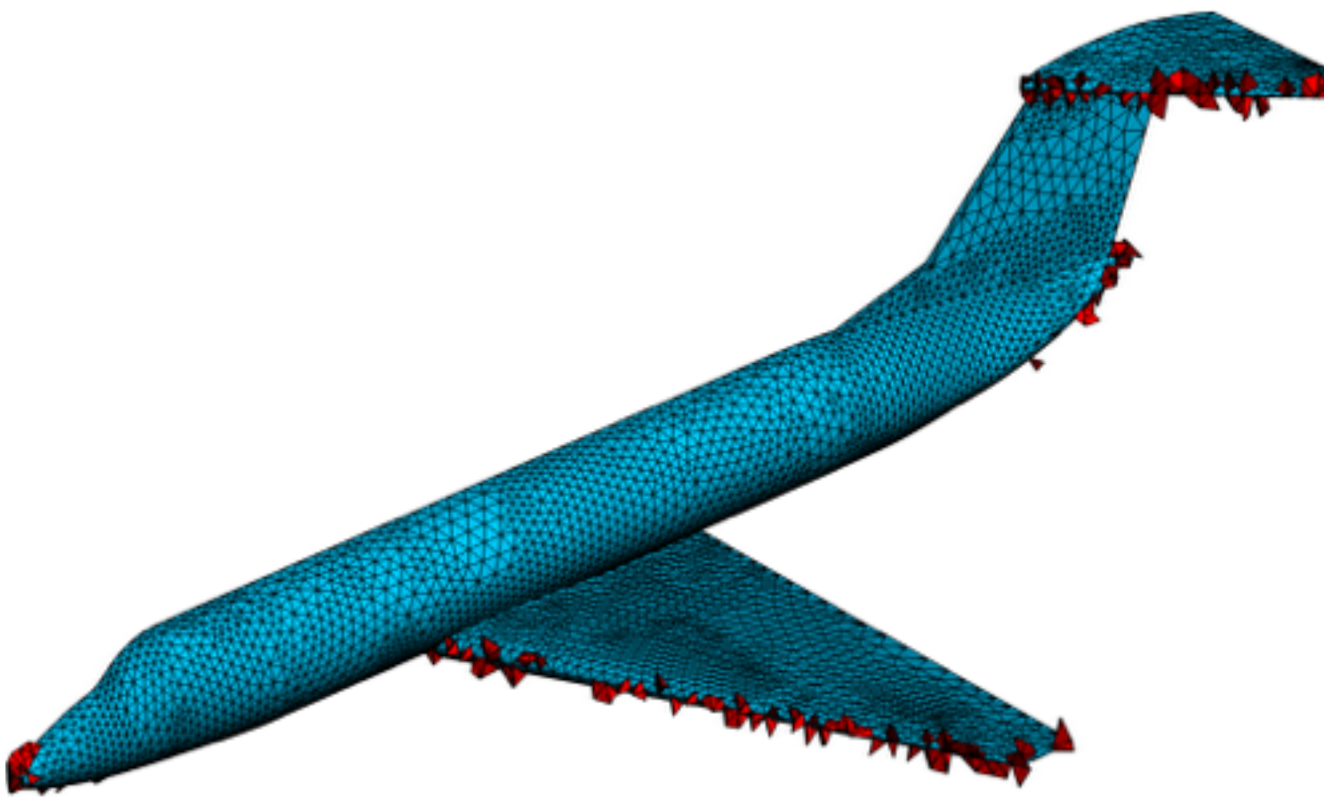
Very efficient parallel implementation with a simple colouring scheme + Newton-based node-by-node optimisation scheme



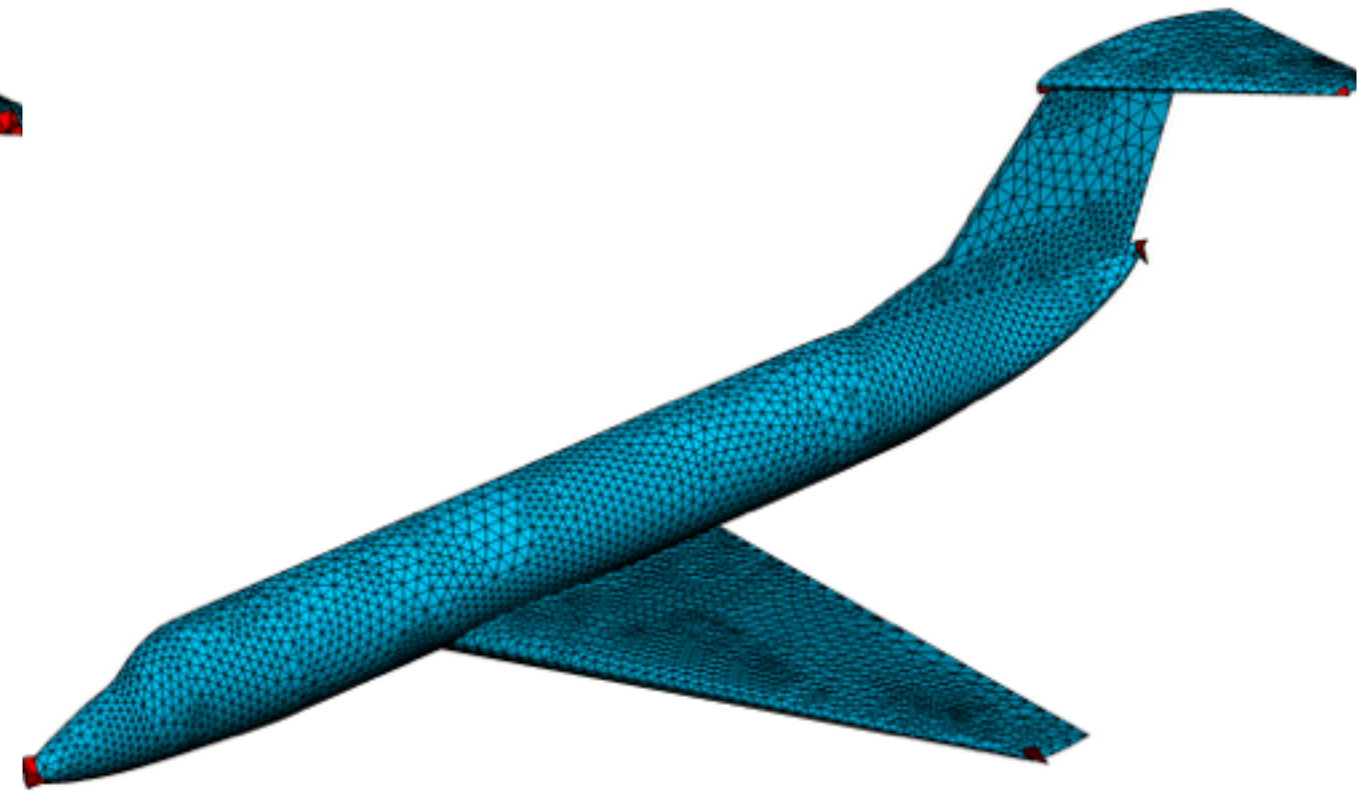


$\approx$ 375,000 DoF

Example: jet configuration

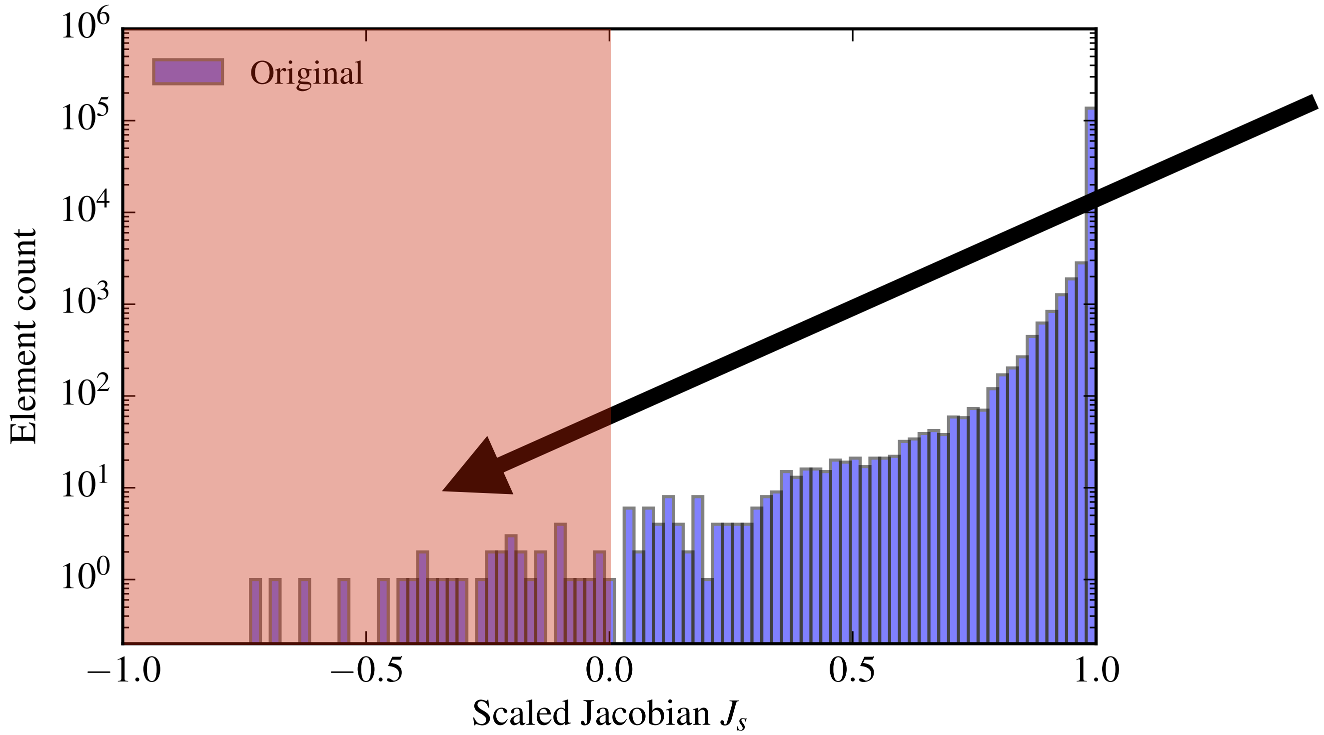


Before optimisation
$J < 0.5$

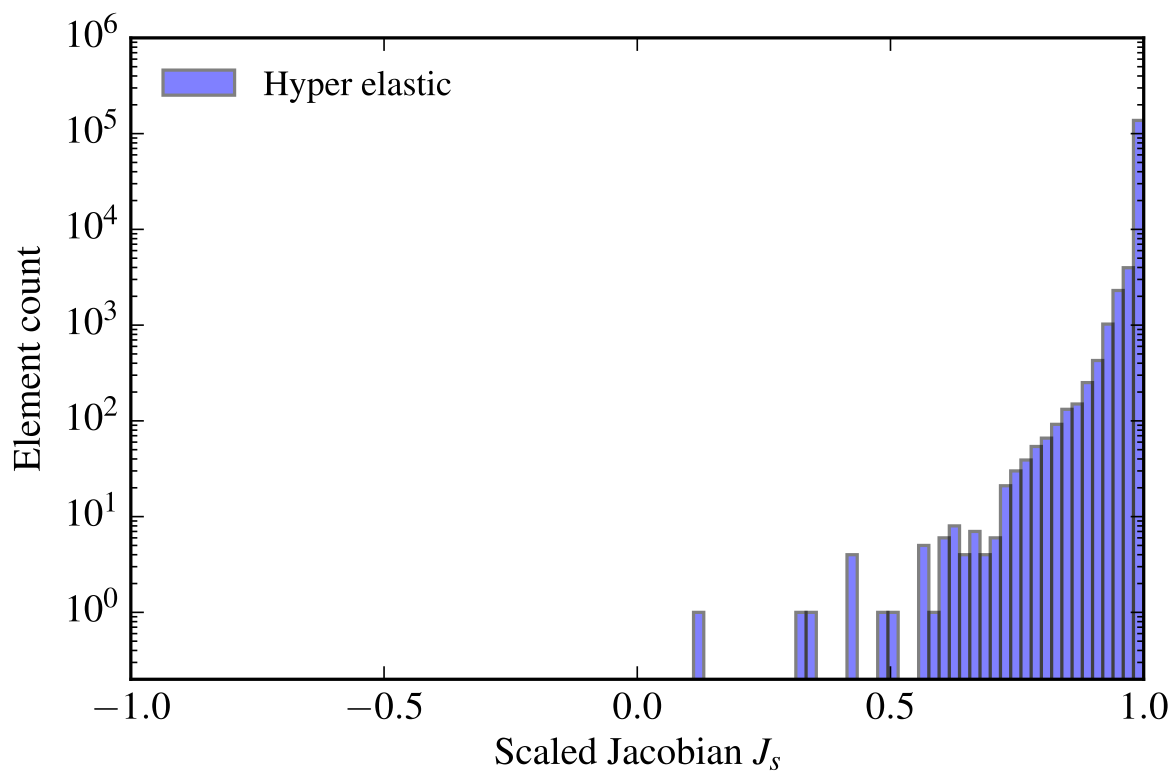


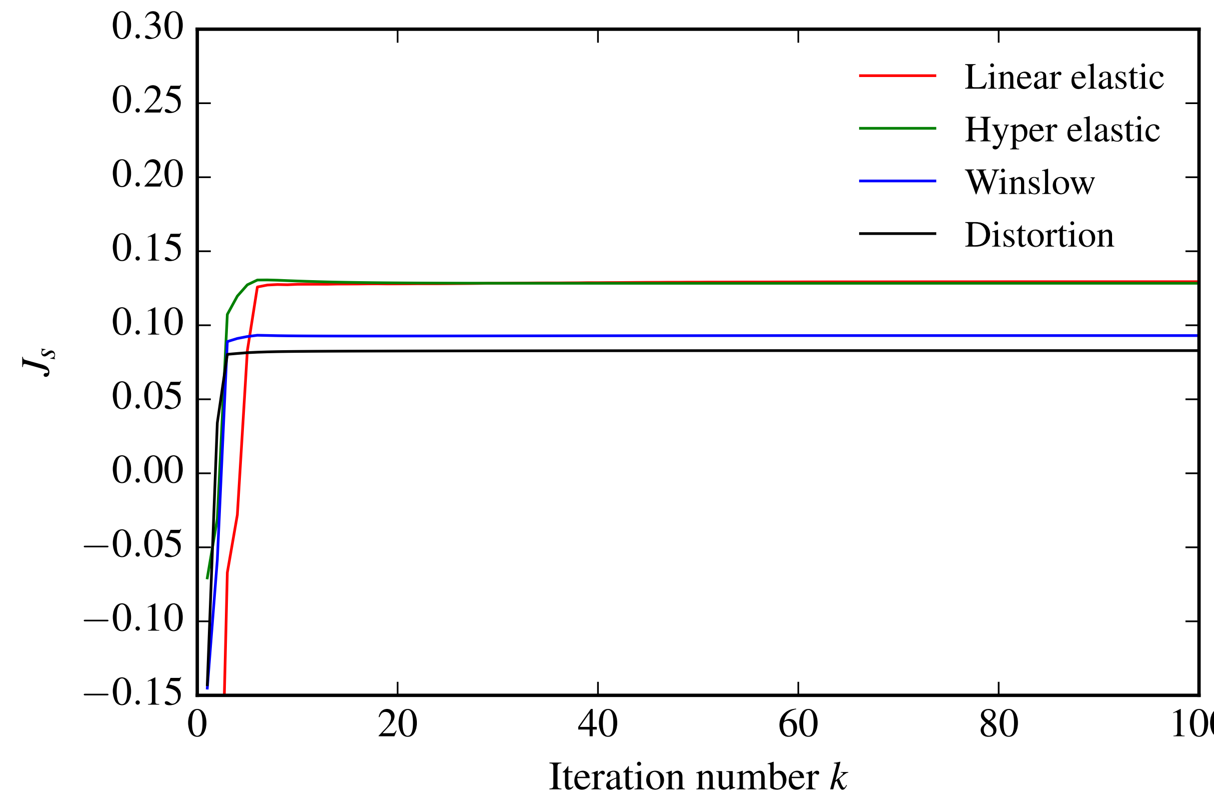
After

Example: jet configuration

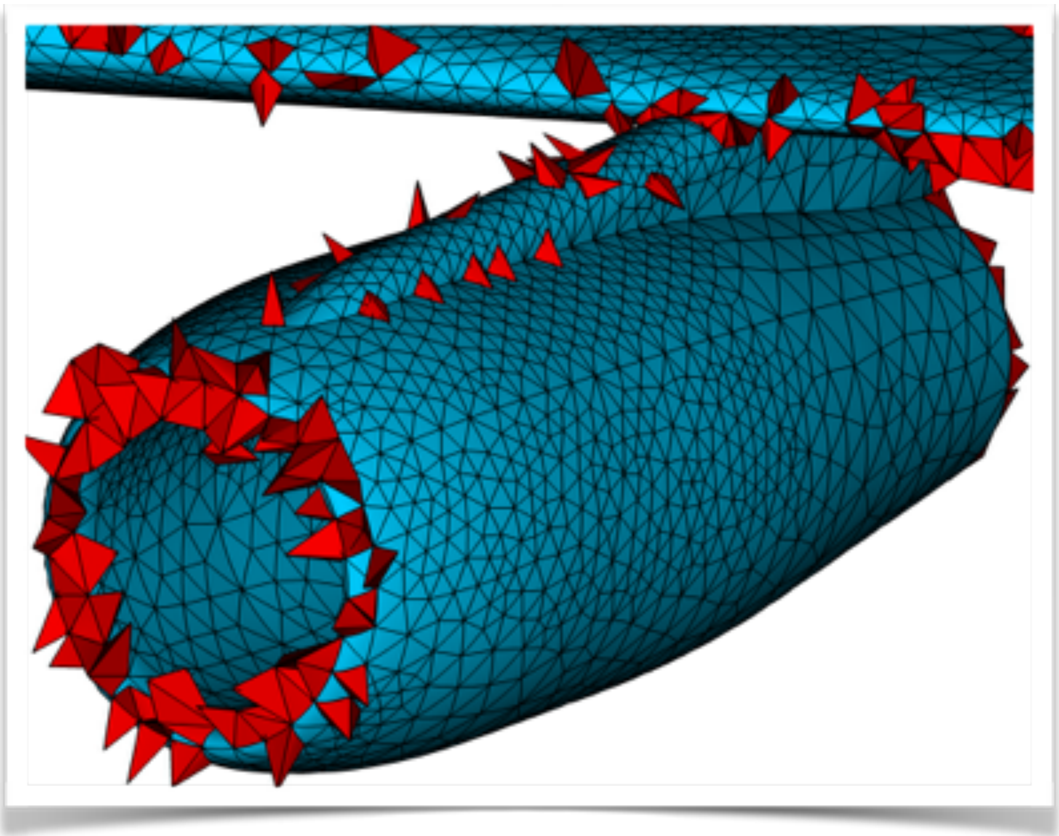


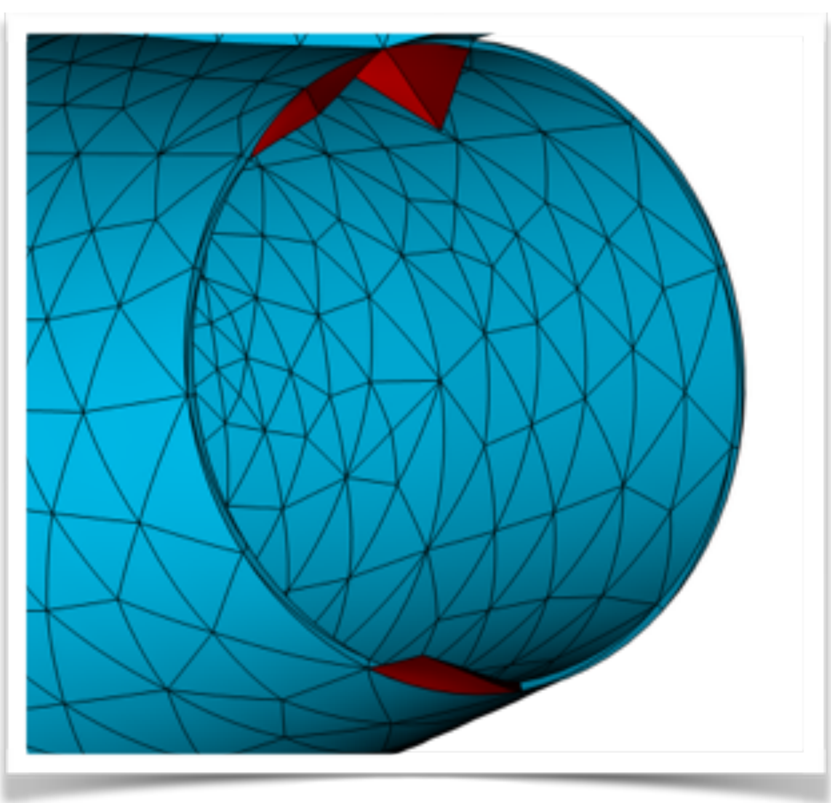
Lots of invalid elements

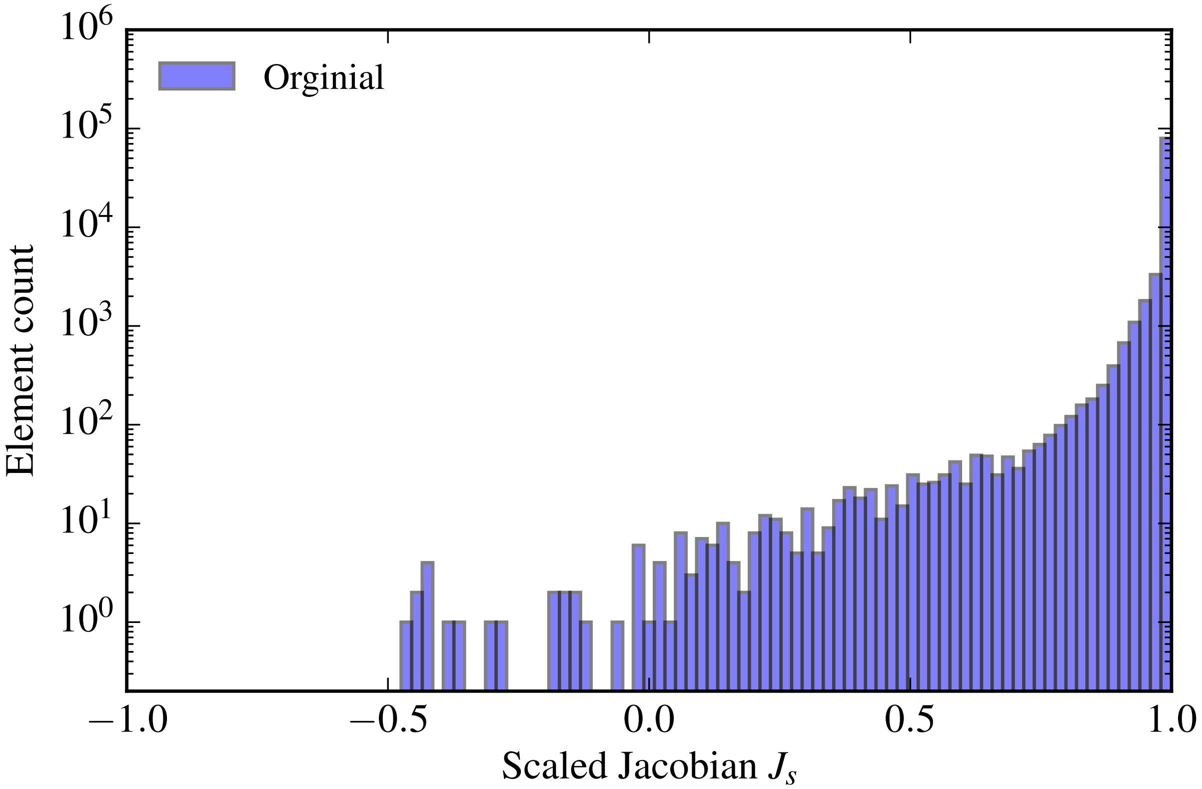


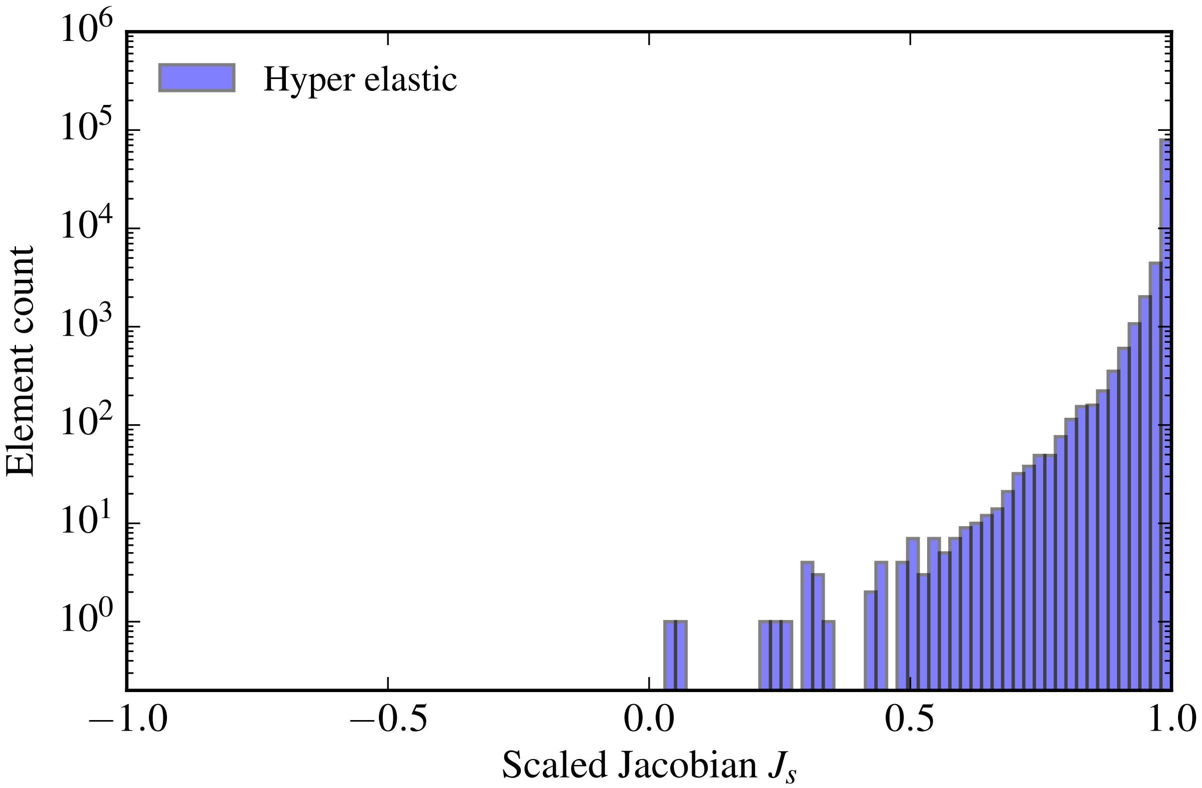


Example: DLR F6 engine

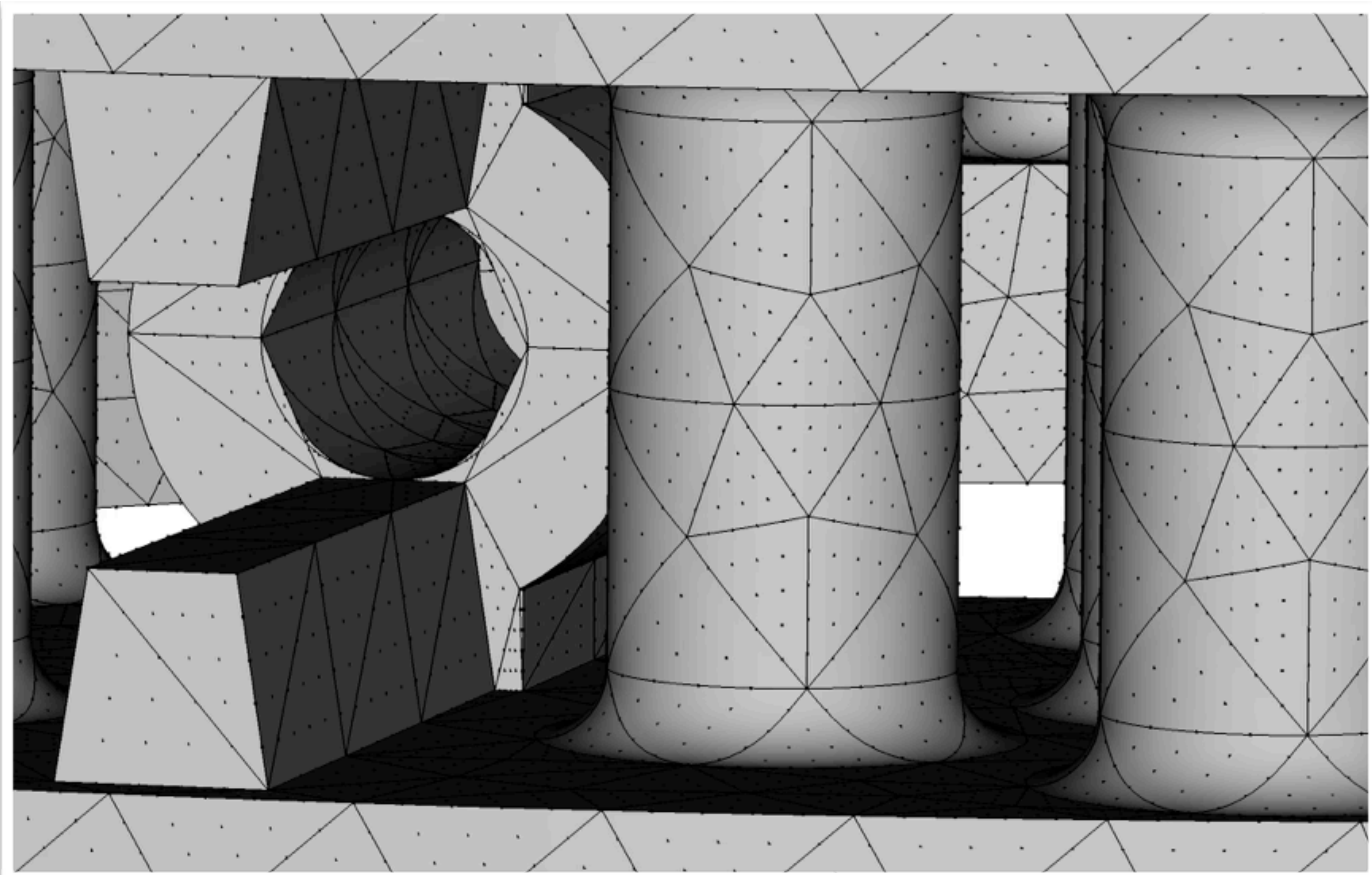








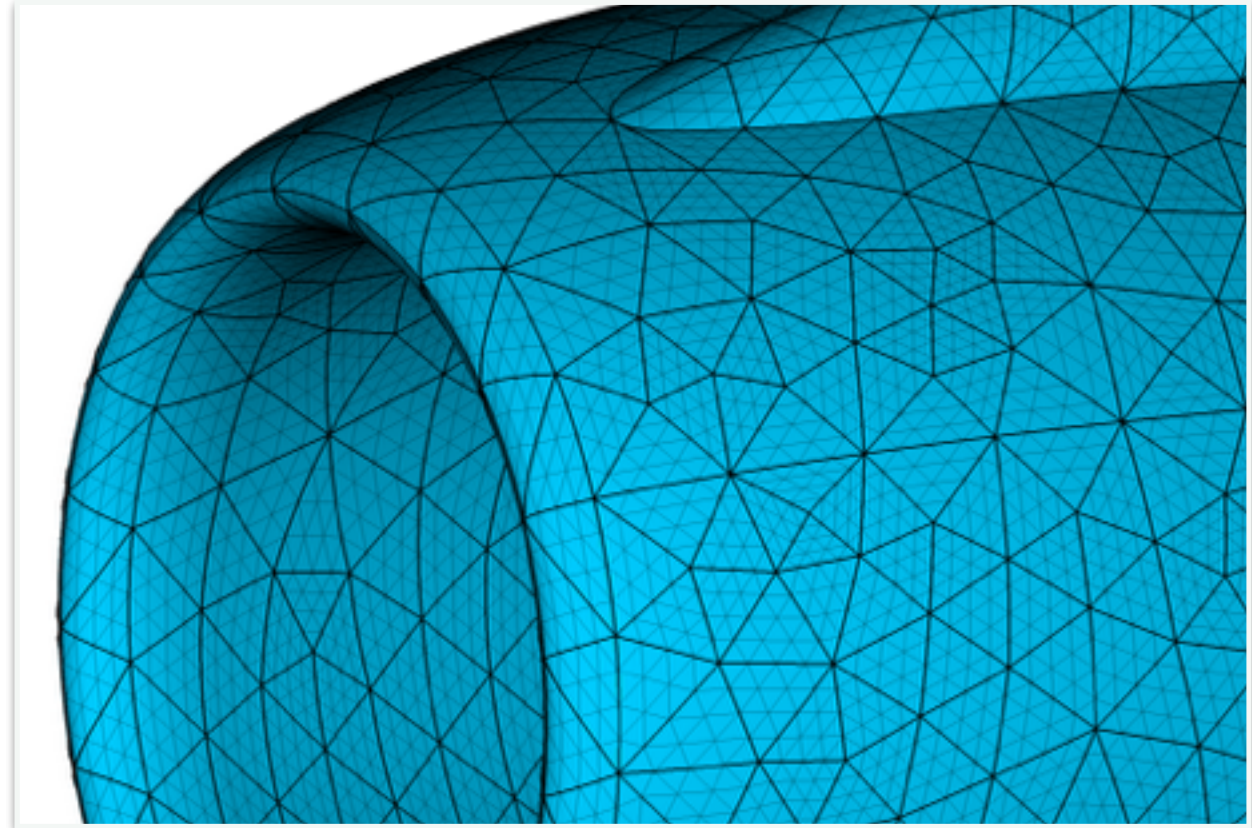
Surface mesh optimisation

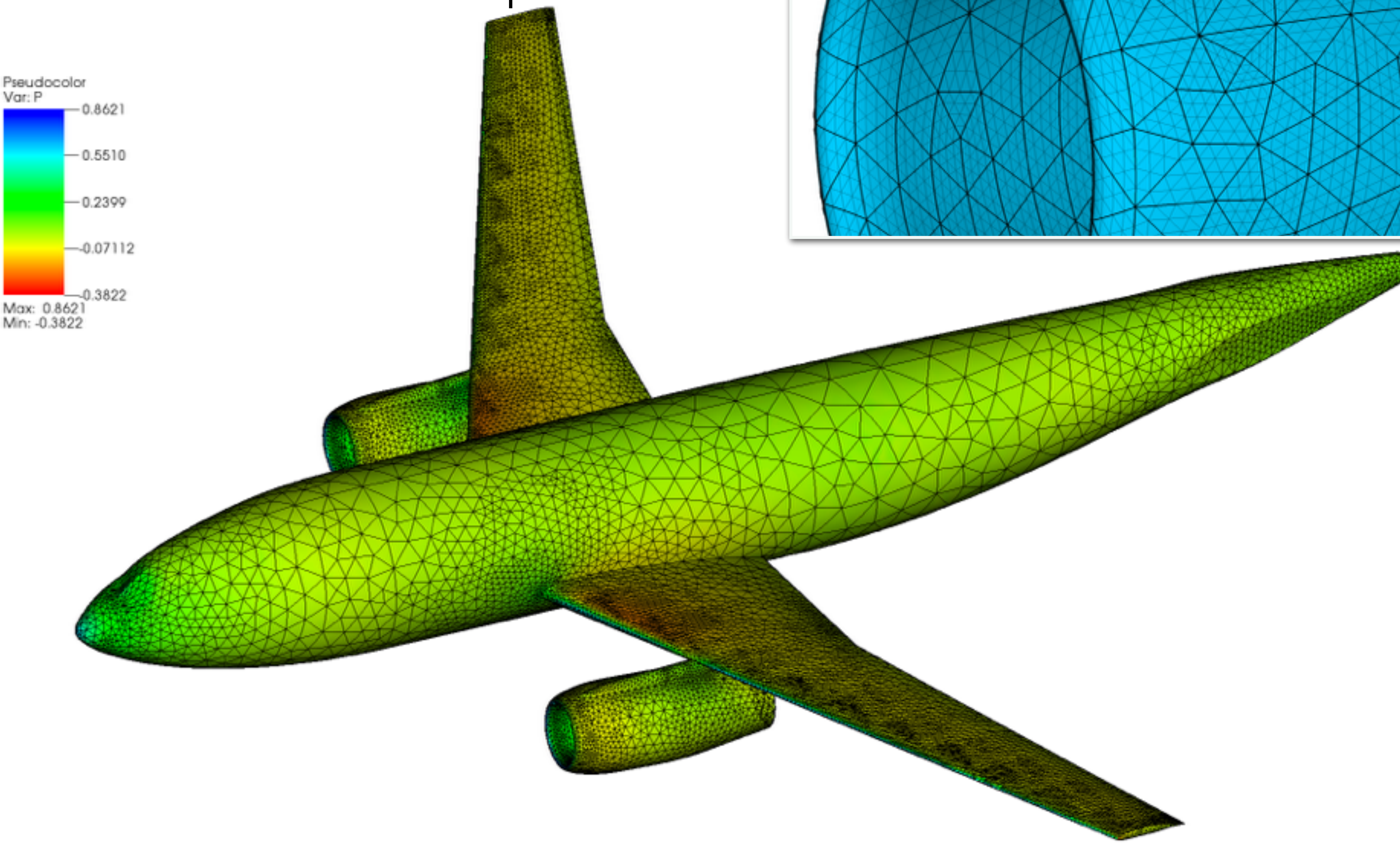


DLR F6

$P = 4$

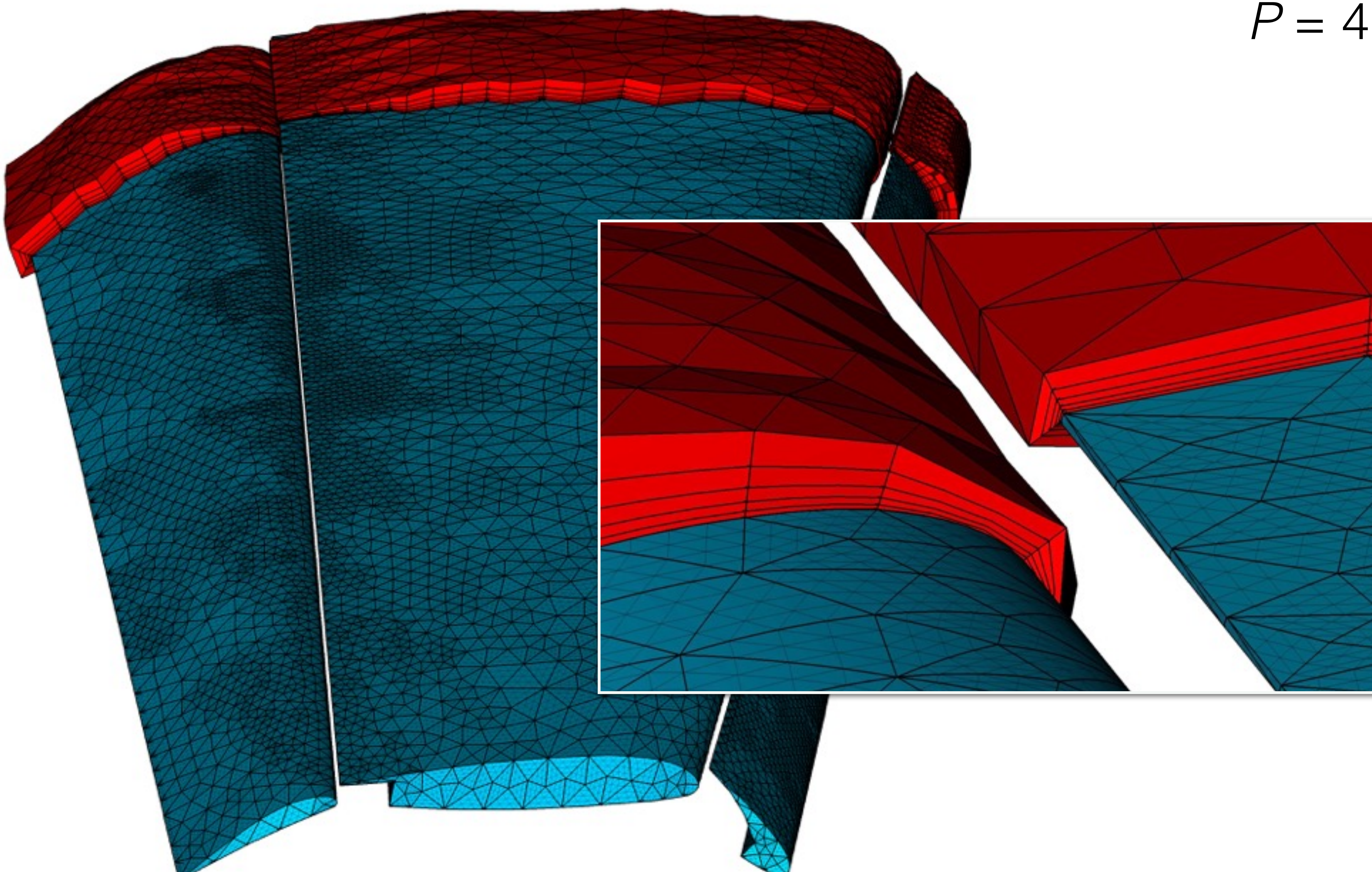
Contours: surface pressure





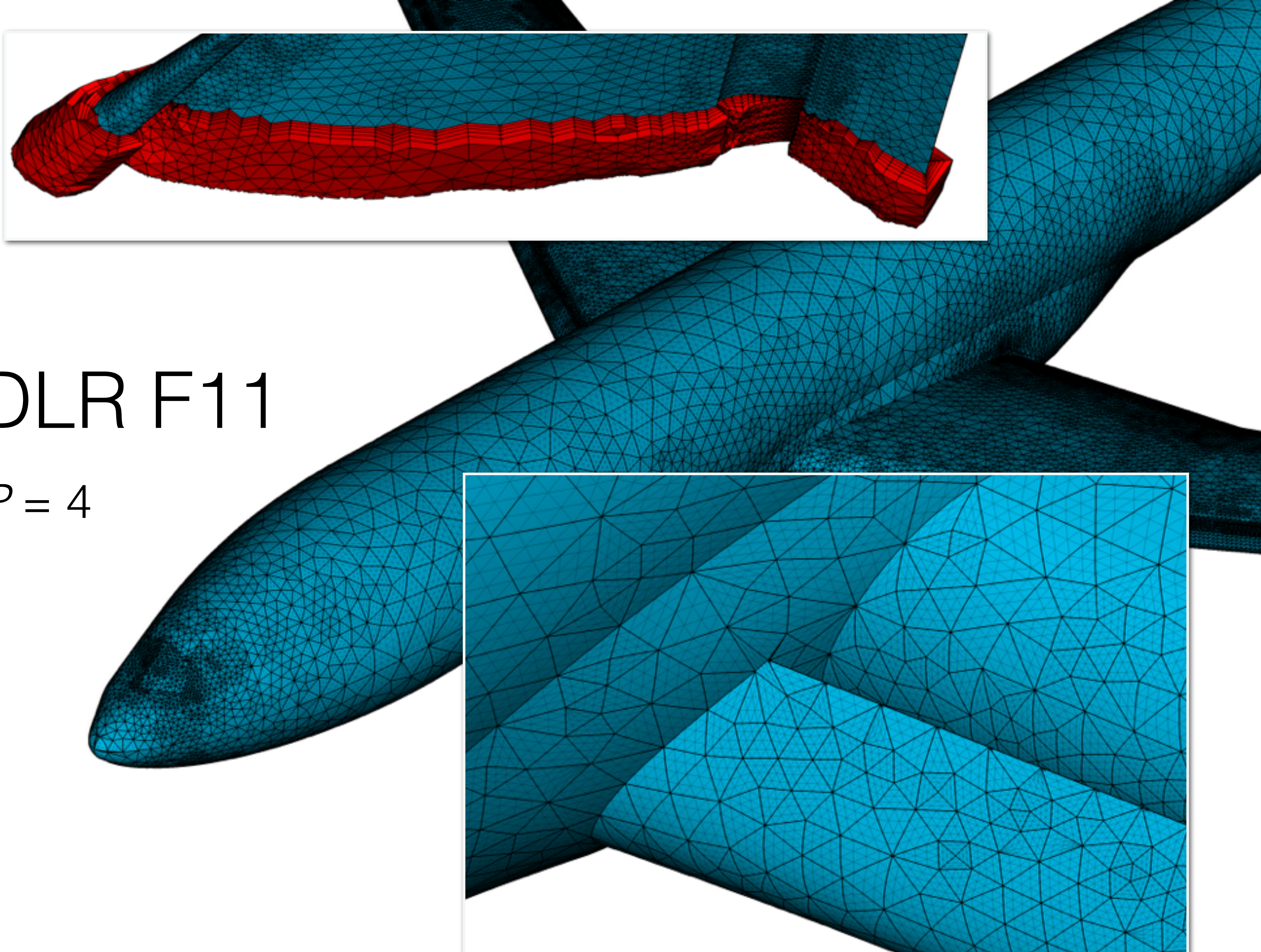
NASA "Trap Wing"

$P = 4$



DLR F11

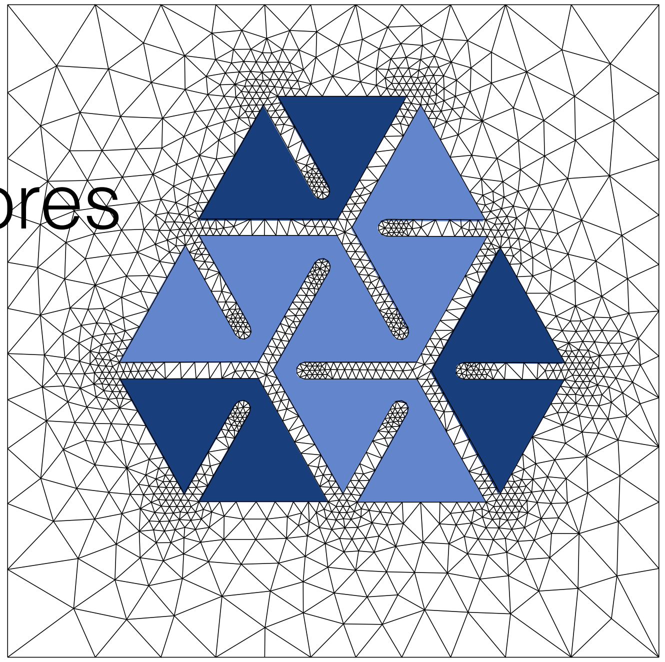
$P = 4$



Nektar++ high-order framework

Framework for spectral(/hp) element method:

- Dimension independent, supports CG/DG/HDG
- Mixed elements (quads/tris, hexes, prisms, tets, pyramids) using hierarchical modal and classical nodal formulations
- Solvers for (in)compressible Navier-Stokes, advection-diffusion-reaction, shallow water equations, ...
- Parallelised with MPI, tested scaling up to ~10k cores



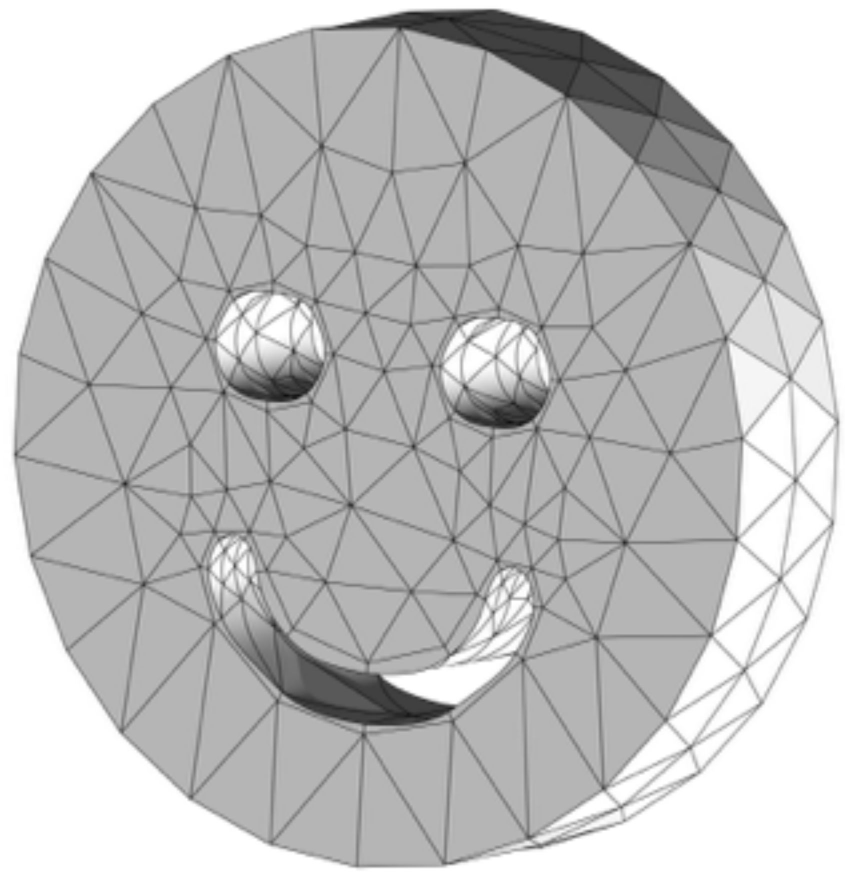
http://www.nektar.info/

nektar-users@imperial.ac.uk

Ongoing and future work

- Variational optimisation for hybrid meshes: triangles and quadrilaterals; tetrahedra, hexahedra, pyramids and prisms
- Mesh adaptation: Incorporate mesh control via the functional
- Open-source code NekMesh release with packages
- Variational boundary-layer mesh generation?
- High-order aware “linear” mesh generation: Incorporate criteria to accommodate high-order mesh information

Thanks for listening!



@davidmoxey

d.moxey@imperial.ac.uk

www.nektar.info   https://gitlab.nektar.info/nektar/nektar