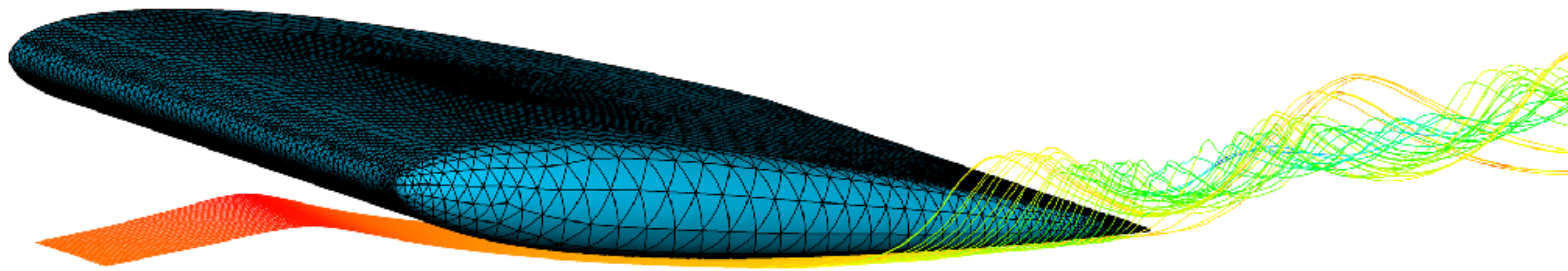
NekMesh: new features and functionality

David Moxey
College of Engineering, Mathematics and Physical Sciences, University of Exeter

Michael Turner, Julian Marcon and Joaquim Peiró
Department of Aeronautics, Imperial College London

Nektar++ Workshop, London, UK
14th June 2017



What is NekMesh?

- **A high-order unstructured mesh generator for complex geometries for arbitrarily high orders**

- Powerful high-order processing technologies
  - ➡ correction and mesh optimisation
  - ➡ boundary layer refinement
  - ➡ spherigon smoothing, extrusion, …

- Support for reading loads of formats as well as writing formats (e.g. can do Nek5000 to Gmsh conversion)

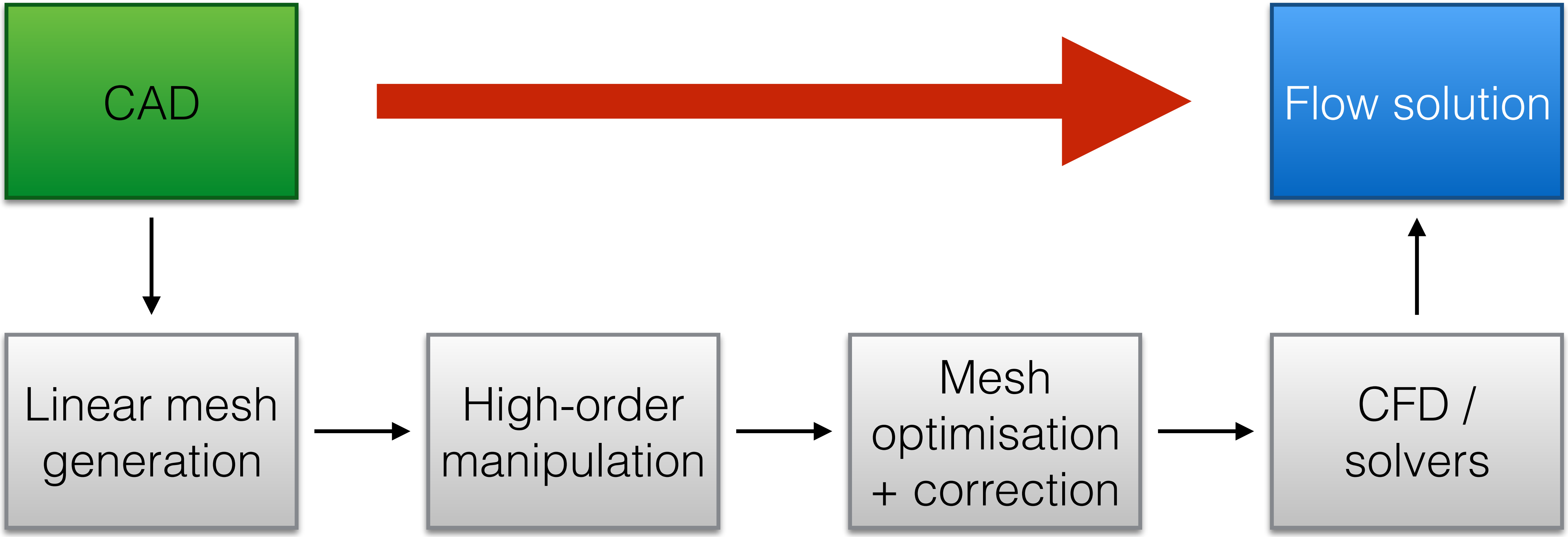
- Previously called MeshConvert **but no longer a converter**

Philosophy

Single step process from CAD to flow solution

As few user parameters as possible - automatic curvature refinement

Preserve CAD throughout



Last 12 months

NekMesh:

- Modify curve module to allow for spline input (!628)
- Add STL surface writer module (!668)
- New module for inserting an alternate high-order surface into the working mesh (!669)
- Add curve projection routines to CAD system (!697)
- Extensive clean-up of NekMeshUtils/MeshElements and extension of makeorder to consider CAD information (!698)
- Improvements to mesh linearisation module (!659)
- Add support for Gmsh high-order output (!679)
- Move CAD classes to factory format (!676)
- Add module to check topology of the mesh along with boundary connectivity to detect problems such as hanging nodes (!691)
- Add option to `linearise` module to linearise only prisms (!688)
- Add reader for Nek5000 mesh files (!680)
- Add option to `linearise` to use element quality (!690)
- Add flag to `insertsurface` process for non-conforming geometries (!700)
- Bug fix to get two meshgen regression tests working (!700)
- Remove libANN in deference to boost::geometry (!703)
- Refactor library to use NekMesh modules for CAD generation (!704)
- Add `varopti` process module to optimise meshes (!711)
- Add a mesh extract option to the linearise module to visualise the result (!712)
- 2D to 3D mesh extrusion module (!715)
- Add new two-dimensional mesher from NACA code or step file (!720)
- Fix inverted boundary layer in 2D (!736)
- More sensible element sizing with boundary layers in 2D (!736)
- Change variable names in mcf file to make more sense (!736)
- Fix issues in varopti module so that in can be compiled without meshgen on (!736)
- Replace LAPACK Eigenvalue calculation with handwritten function in varopti (!738)
- Improved node-colouring algorithm for better load-balancing in varopti (!738)
- Simplified calculation of the energy functional in varopti for improved performance (!738)

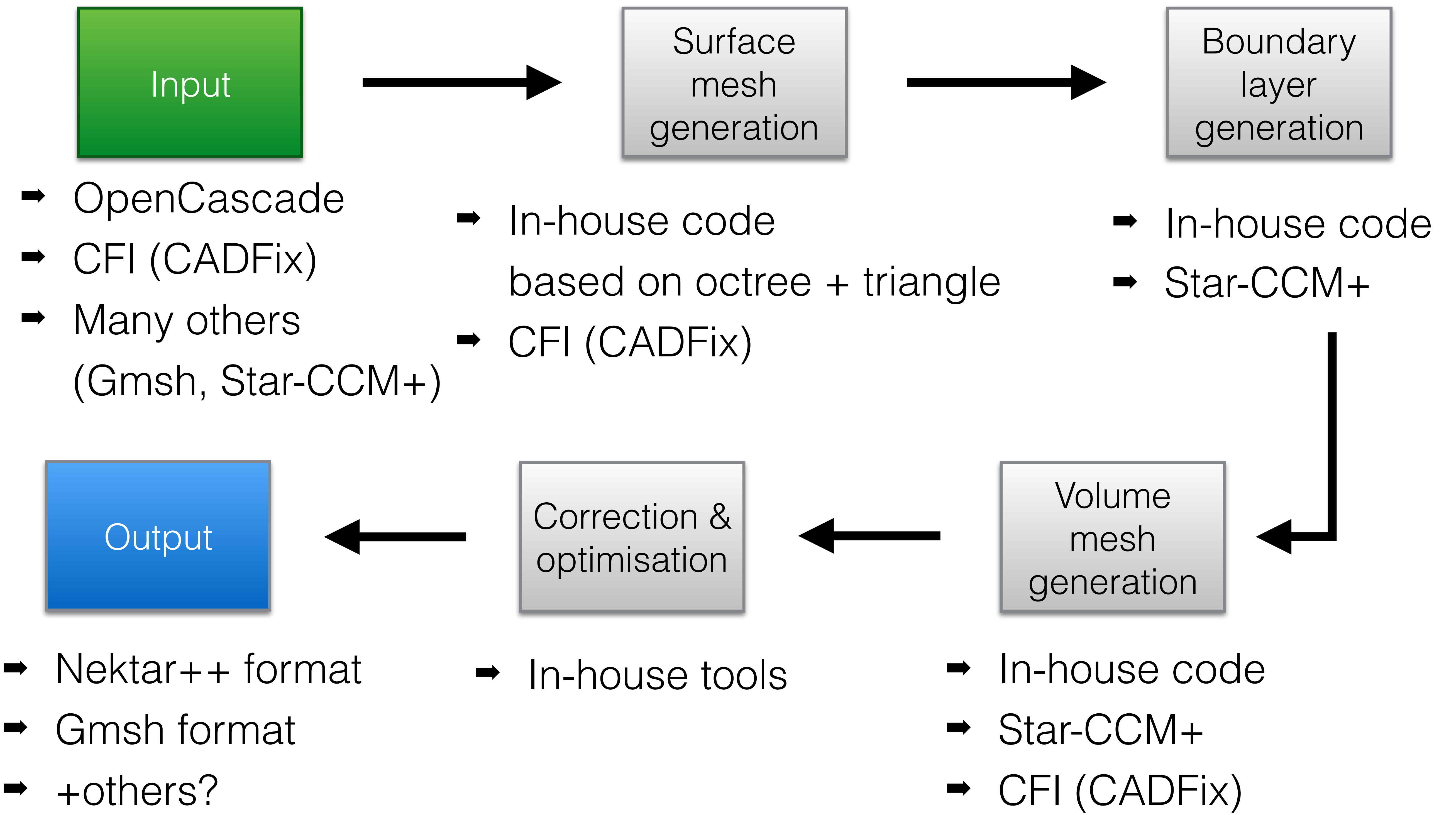
We've been busy!

One of the most active areas of development in Nektar++

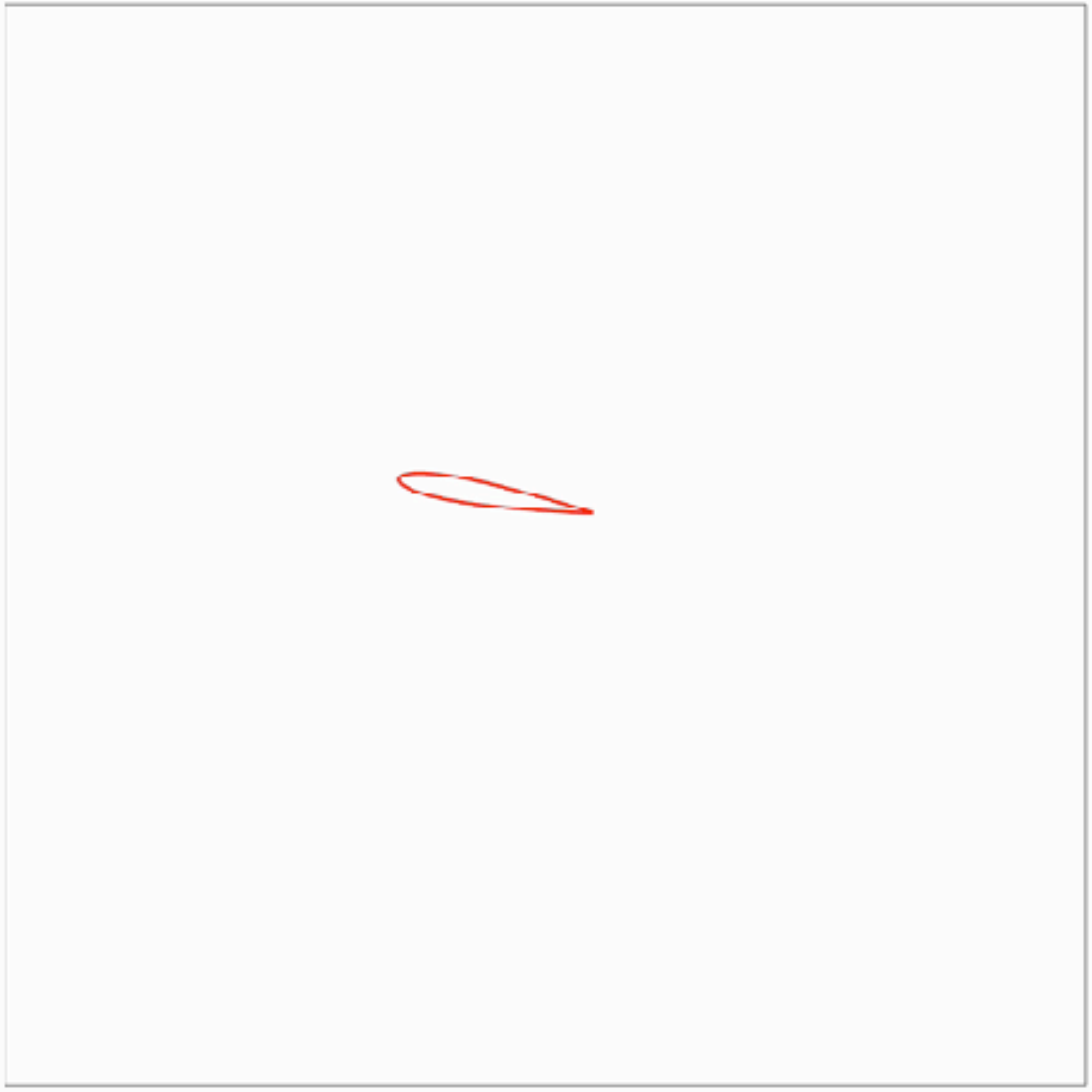
Lots of new functionality and bug fixes

Not everything in v4.4 - you need to run `master`

Pipeline approach

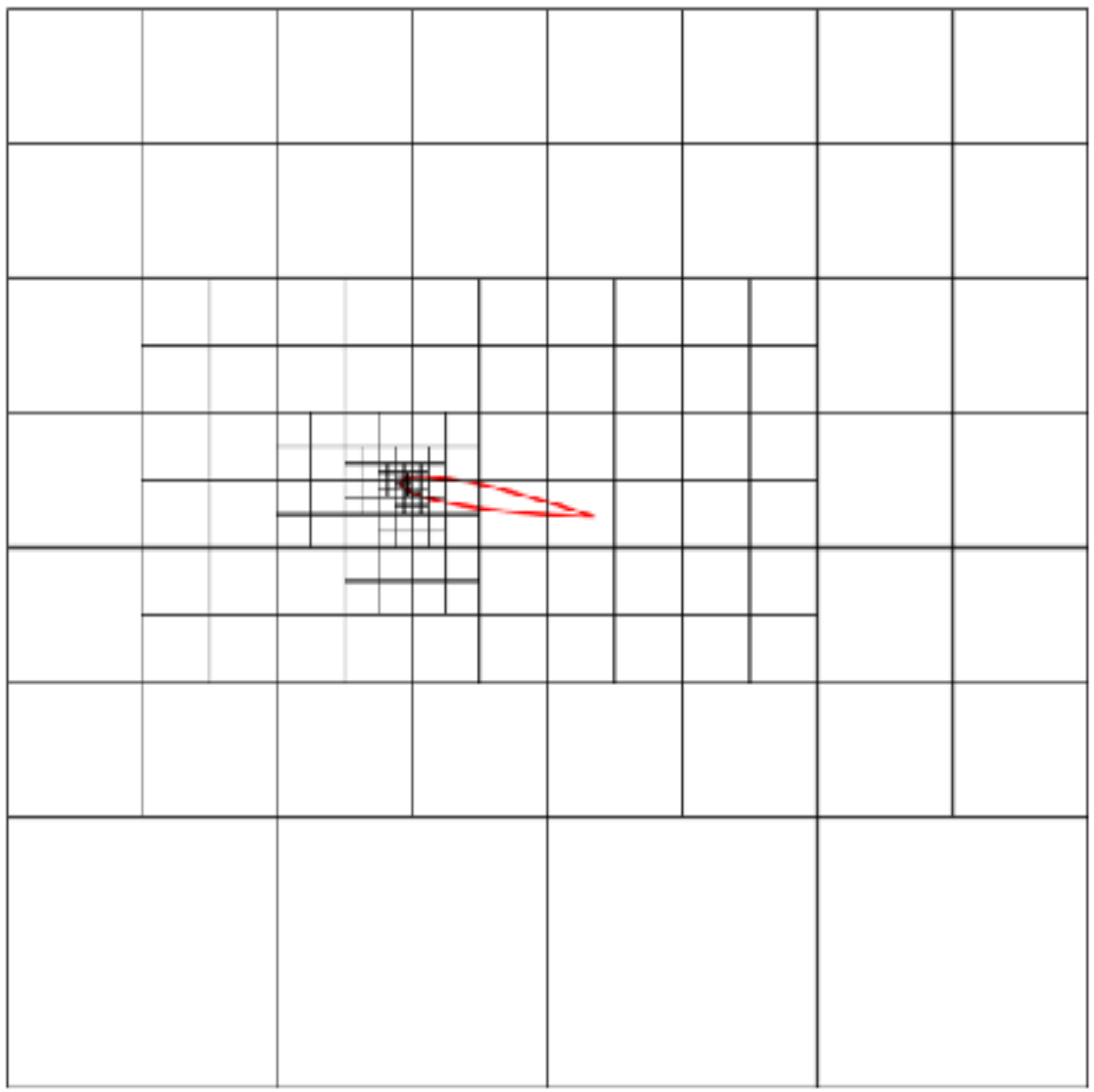


Generating a mesh from CAD



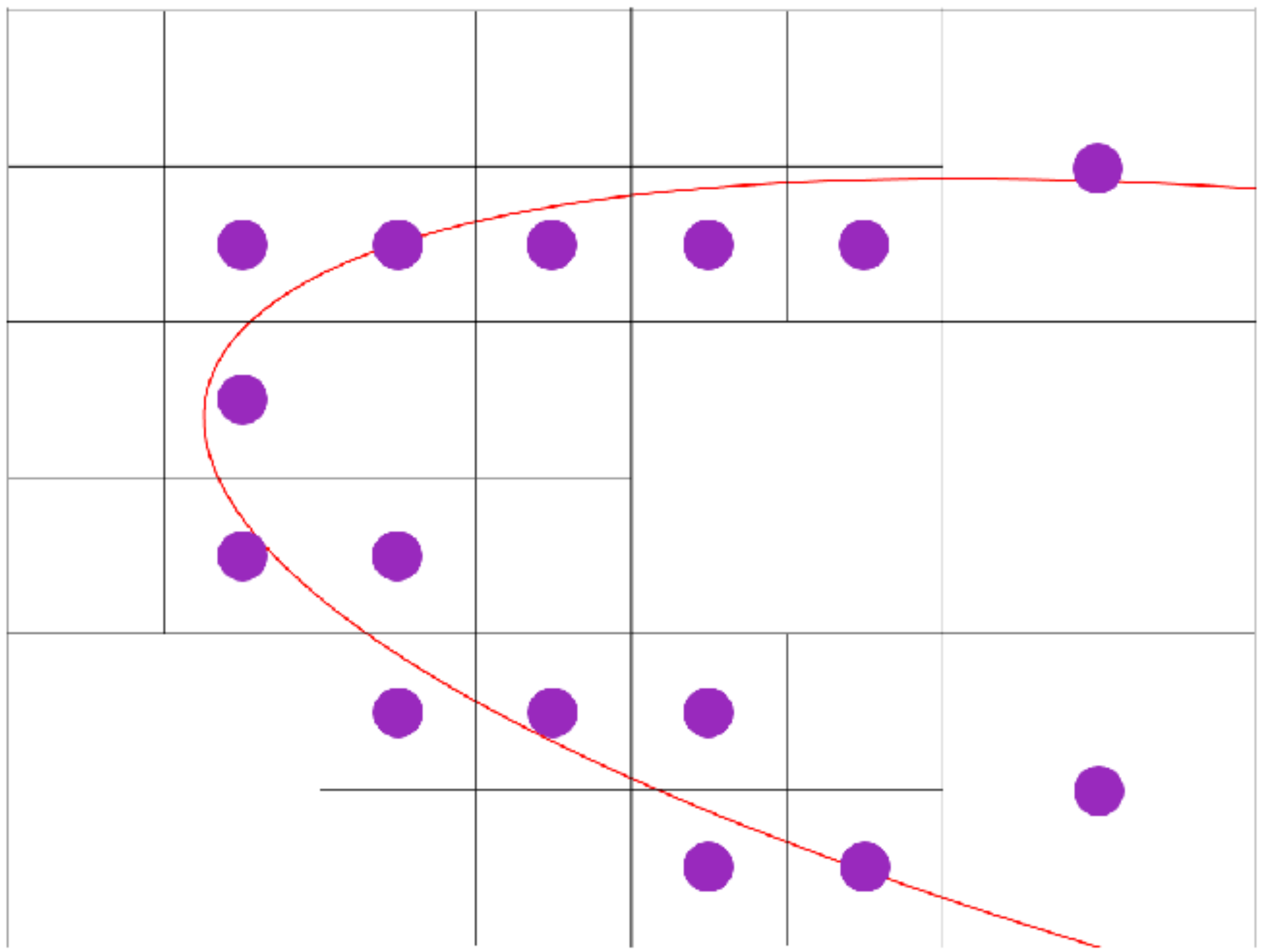
Constructing an octree to refine around curvature

Generating a mesh from CAD



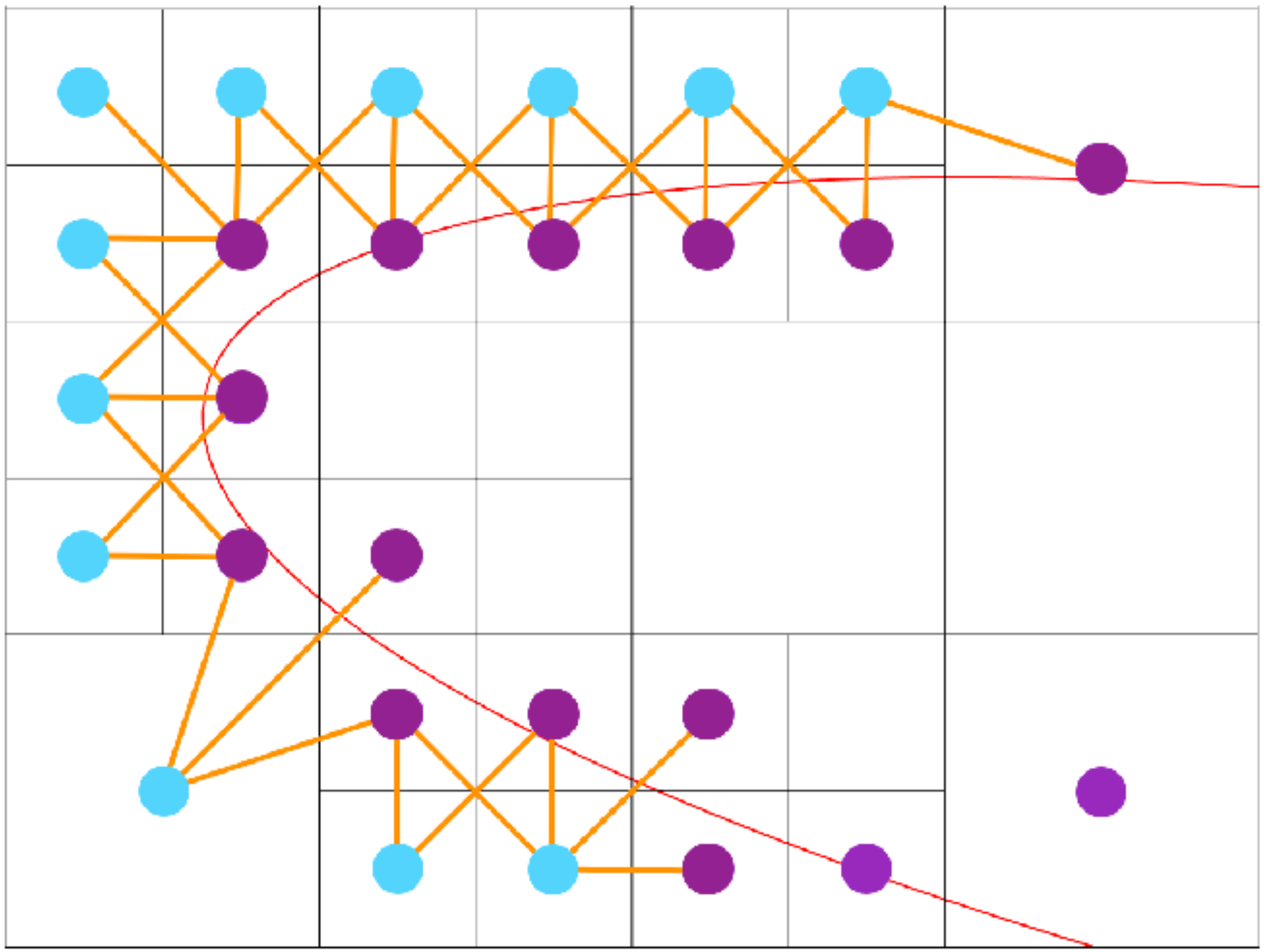
Smoothing the octree

Generating a mesh from CAD



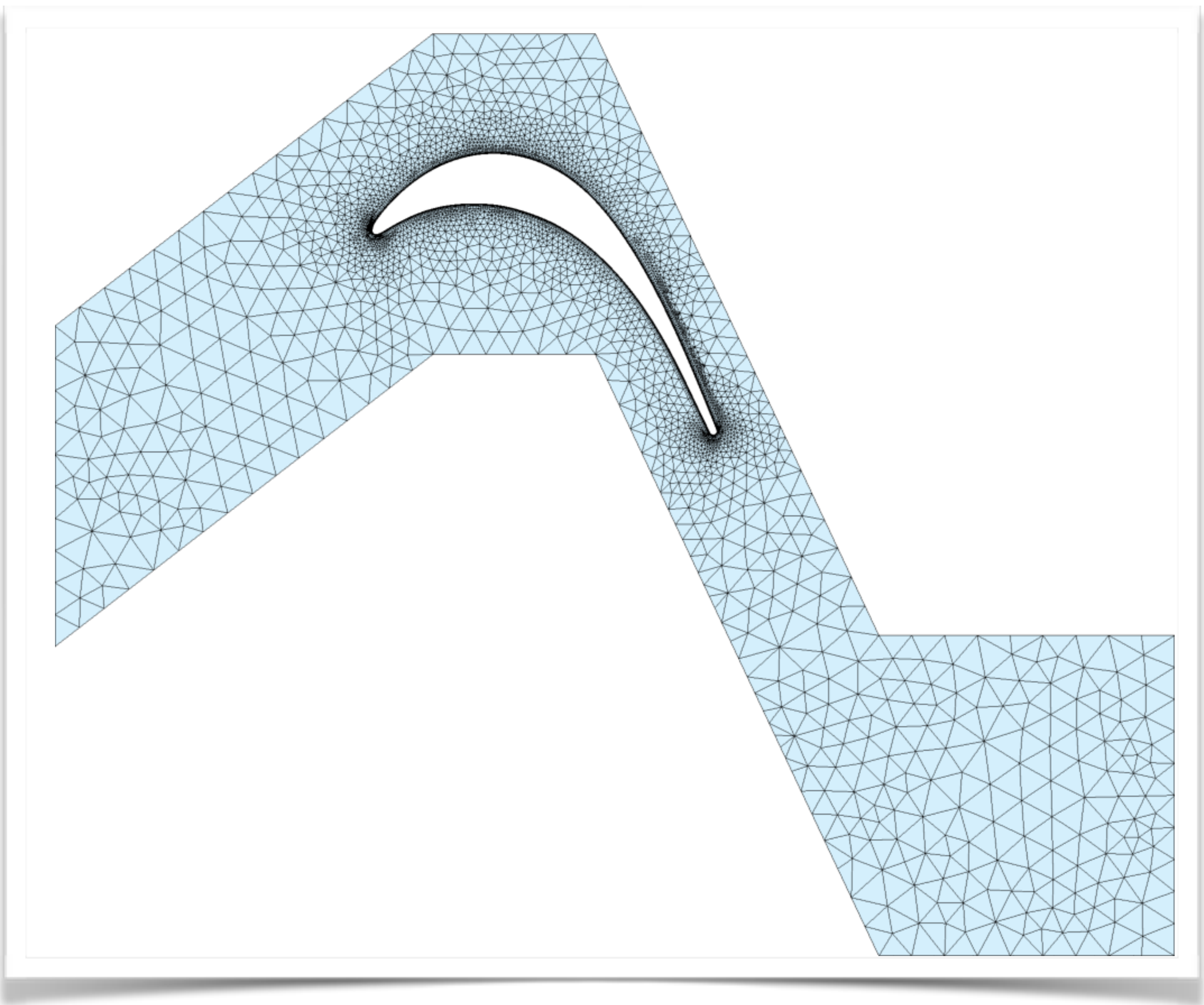
Use curvature to define mesh spacing $\delta(R, \varepsilon)$

Generating a mesh from CAD



Propagate mesh curvature onto interior of domain

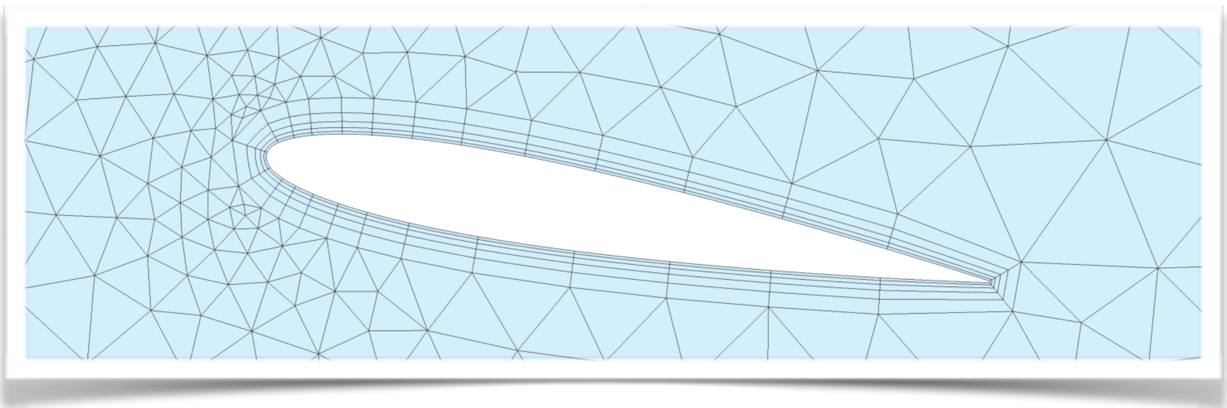
2D mesh generation

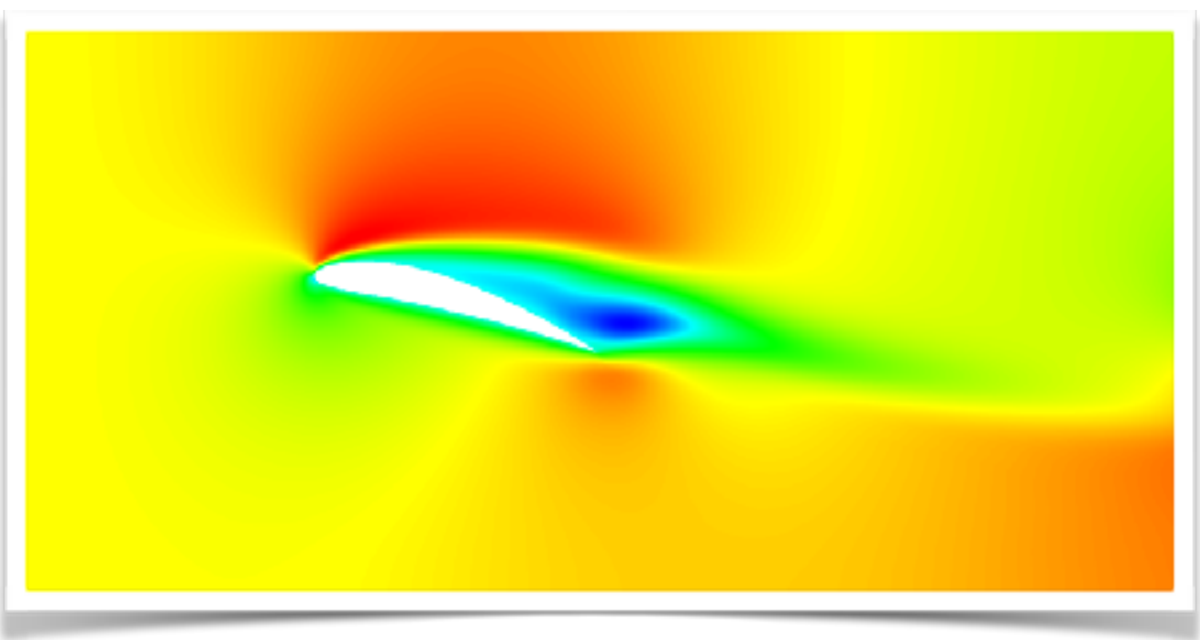


T106C turbine blade

- NekMesh now supports reading 2D STEP files and generating meshes
- Boundary layer generation
- Periodic edge support
- Wake refinement through line sources

2D mesh generation





NACA0012 and NACA4412, 5° AoA

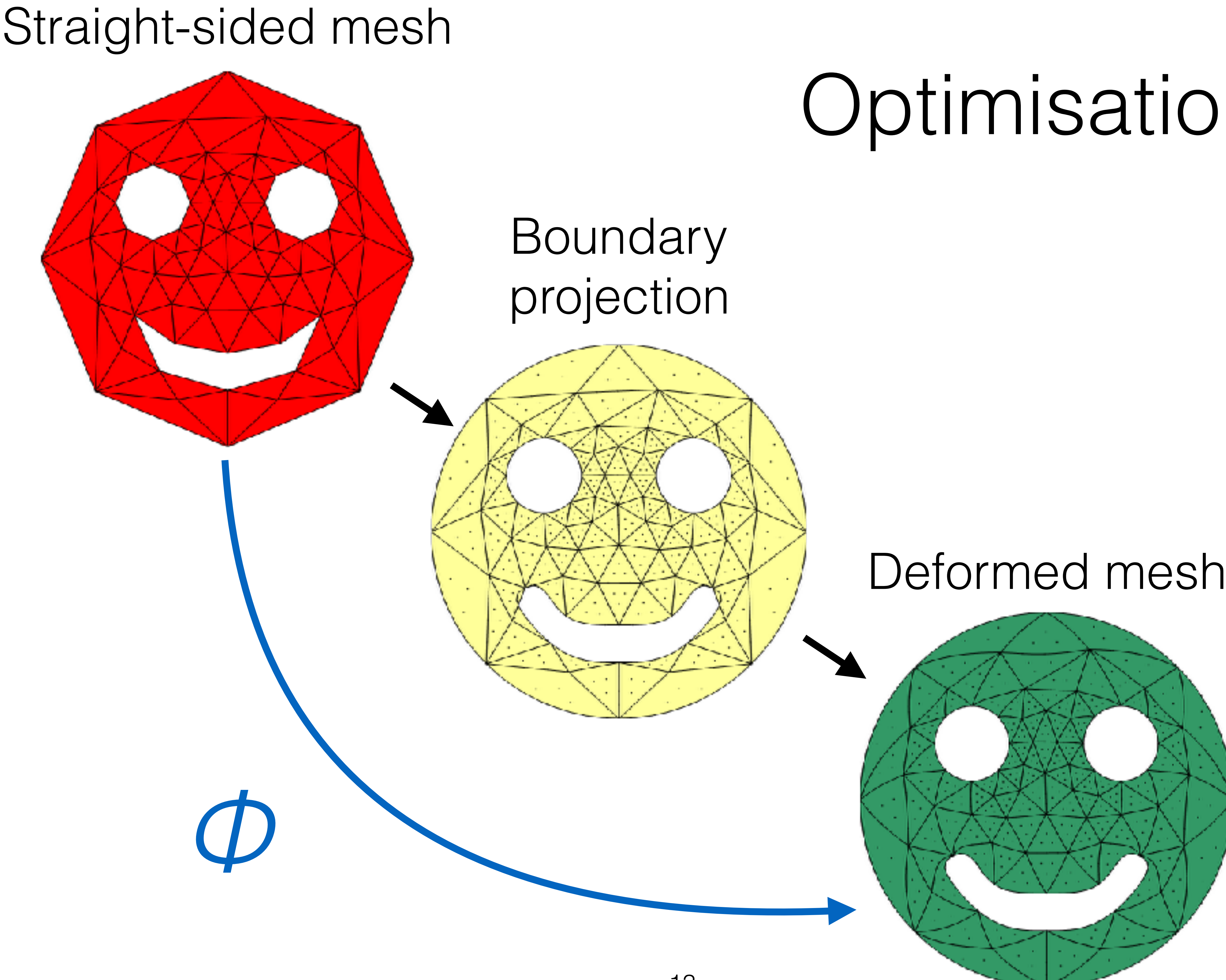
- Also, you don't need a STEP anymore
- Simple .geo Gmsh reader for 2D geometries
- Or just enter a 4-digit NACA code, far-field and AoA to generate a NACA geometry

MCF Input format

```xml
<NEKTAR>
    <MESHING>
        <INFORMATION>
            <I PROPERTY="CADFile" VALUE="0012"       />
            <I PROPERTY="MeshType" VALUE="2DBndLayer" />
        </INFORMATION>
        <PARAMETERS>
            <P PARAM="MinDelta" VALUE="0.1"          />
            <P PARAM="MaxDelta" VALUE="5"            />
            <P PARAM="EPS"      VALUE="0.1"          />

            <P PARAM="Order"    VALUE="4"            />

            <P PARAM="BndLayerSurfaces"  VALUE="5-6"  />
            <P PARAM="BndLayerThickness" VALUE="0.05" />

            <P PARAM="Xmin"     VALUE="-1.0"         />
            <P PARAM="Ymin"     VALUE="-2.0"         />
            <P PARAM="Xmax"     VALUE="3.0"          />
            <P PARAM="Ymax"     VALUE="2.0"          />
            <P PARAM="AOA"      VALUE="15.0"         />
        </PARAMETERS>
        <BOOLPARAMETERS>
            <P VALUE="VariationalOptimiser"          />
        </BOOLPARAMETERS>
    </MESHING>
</NEKTAR>
```

- Store options for generation inside MCF
- Will pre-load some processing modules
- Example shows NACA0012 generation

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)

Variational approach

Instead of viewing problem as a PDE, use calculus of variations: recast as an **integral (energy) minimisation** instead, where we solve the problem

$$\text{find } \min_{\phi} \mathscr{E}(\phi) = \min_{\phi} \int_{\Omega_I} W(\nabla \phi)\, dy$$

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

M. Turner, J. Peiró, D. Moxey, *A variational framework for high-order mesh generation*, Procedia Engineering **82** 127-135 (2016)

Choice of functional

$$\mathsf{F} = \nabla\phi \qquad J = \det \mathsf{F}$$

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

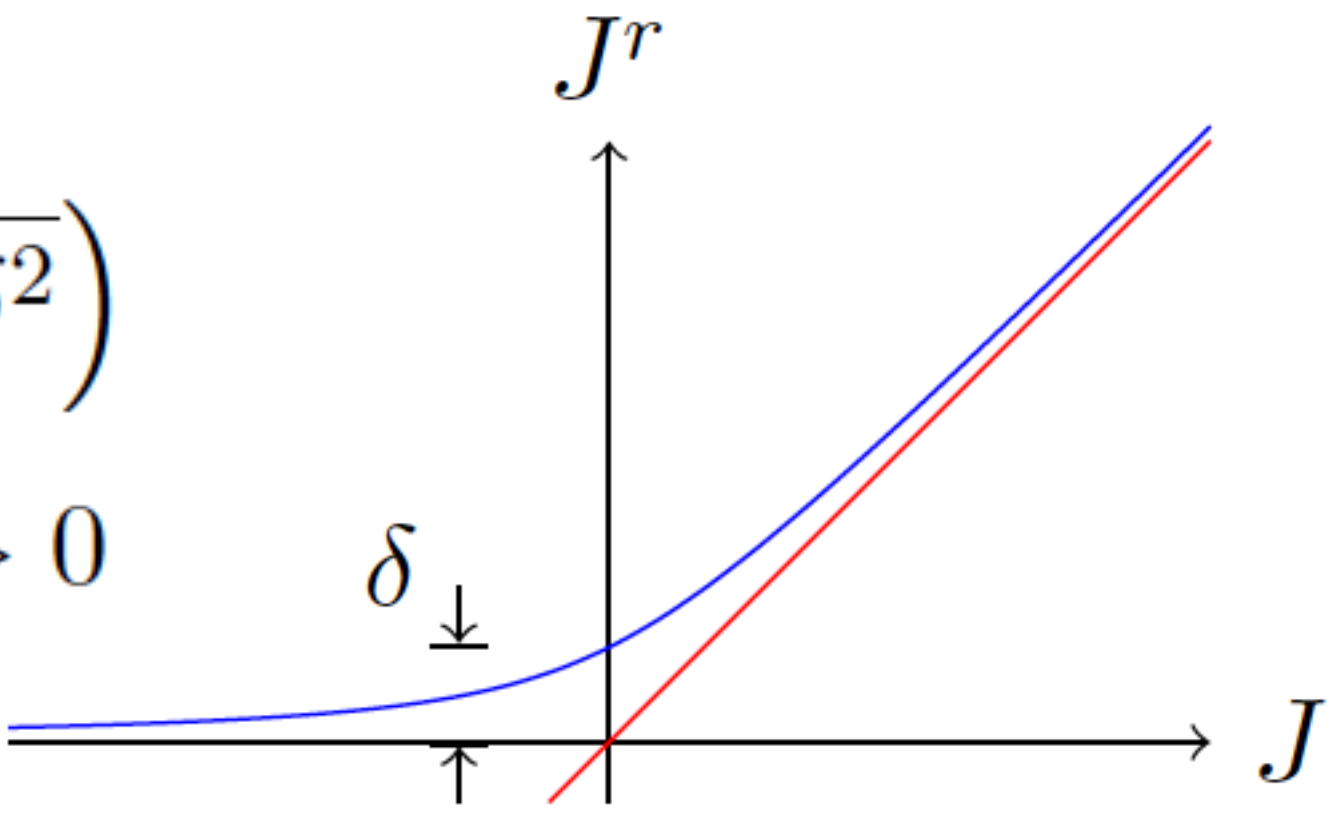
This technique encapsulates most currently available

Invalid mesh: min J_s <0

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

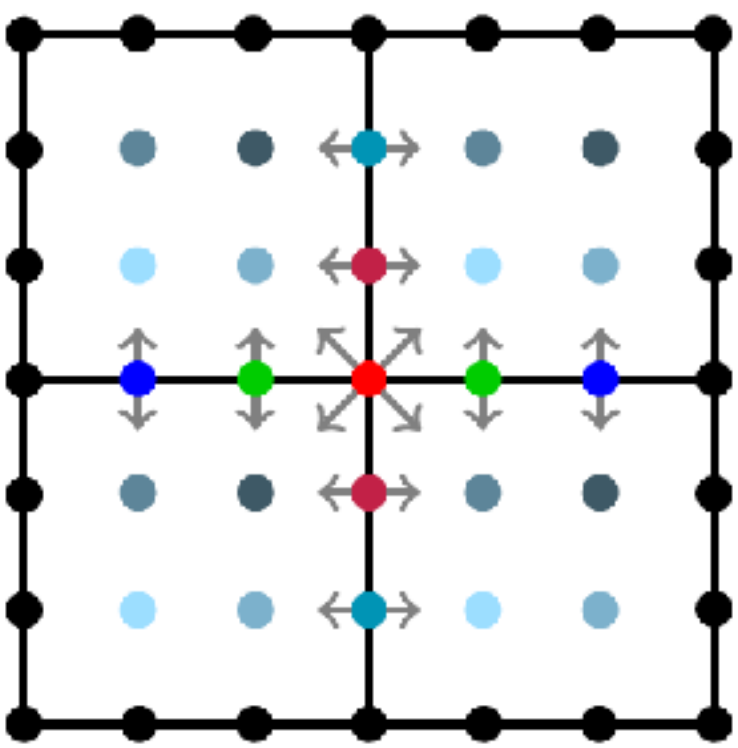
$$J^r = \frac{1}{2}\left(J + \sqrt{J^2 + 4\delta^2}\right)$$

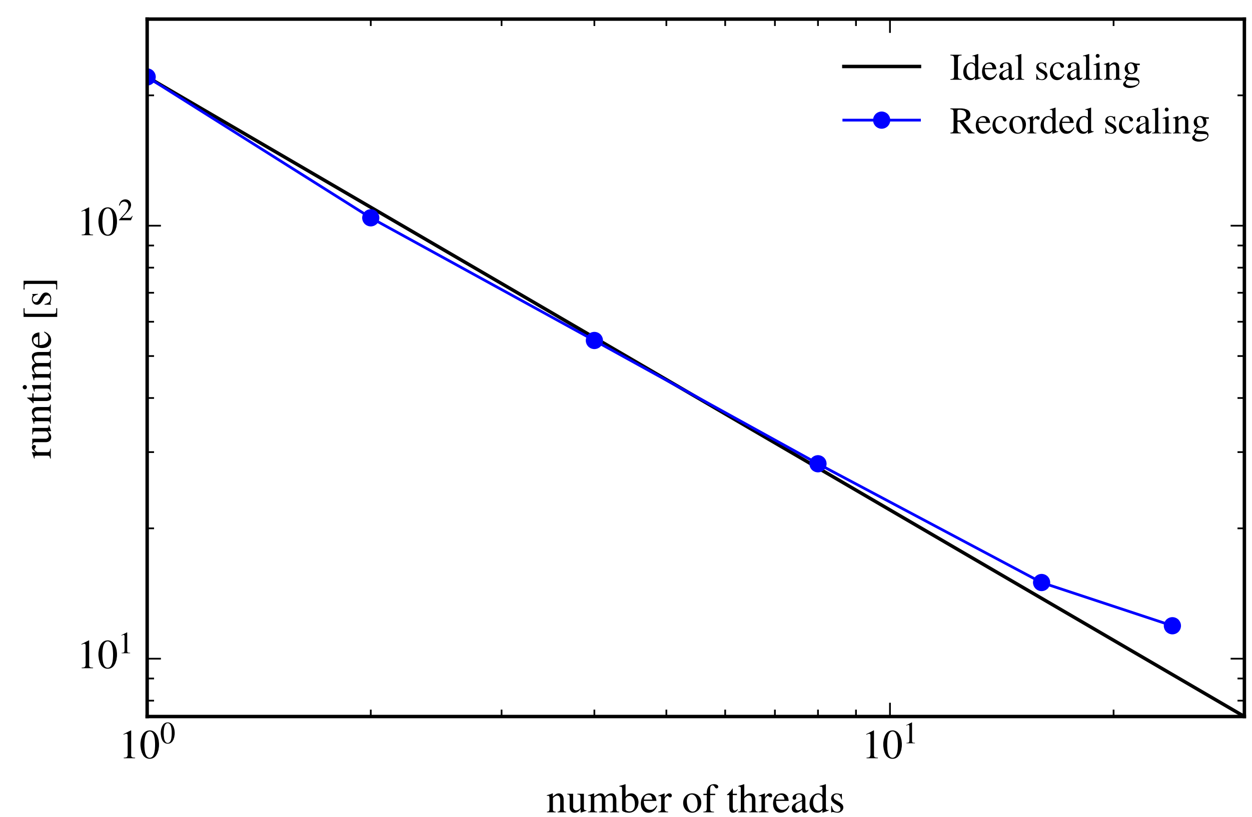
$$\delta(J): \quad \delta \Rightarrow \varepsilon \quad J \Rightarrow > 0$$



Parallelisation

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

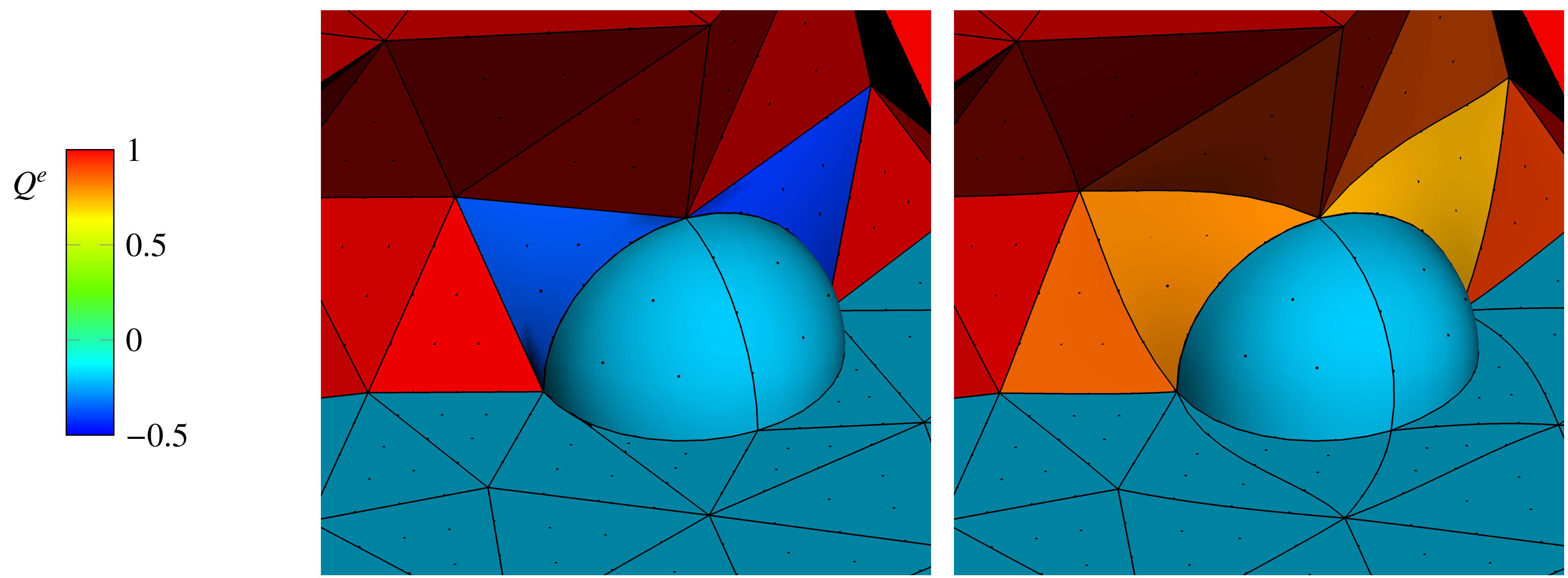




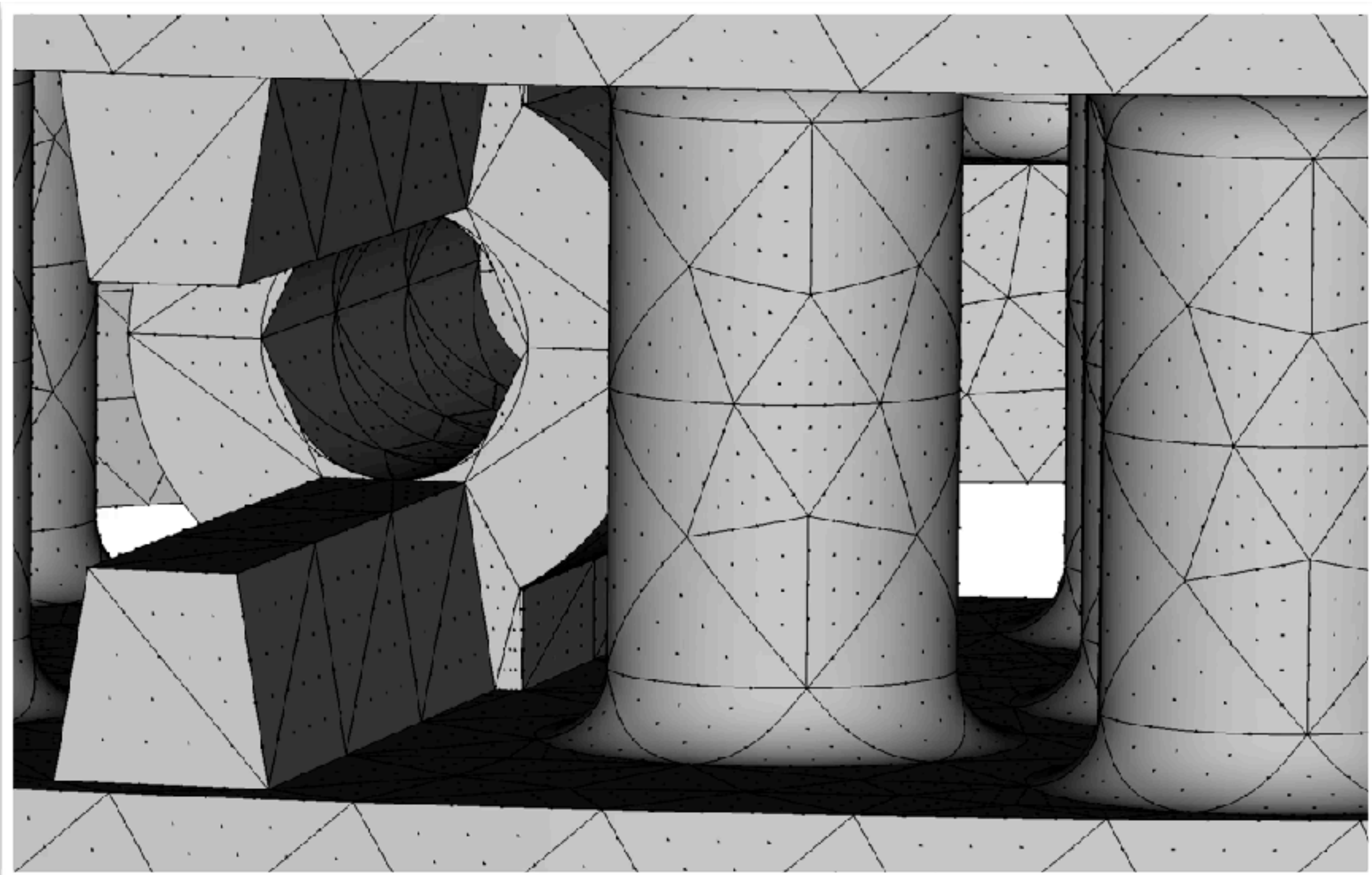
$\approx$ 375,000 DoF

Surface mesh sliding

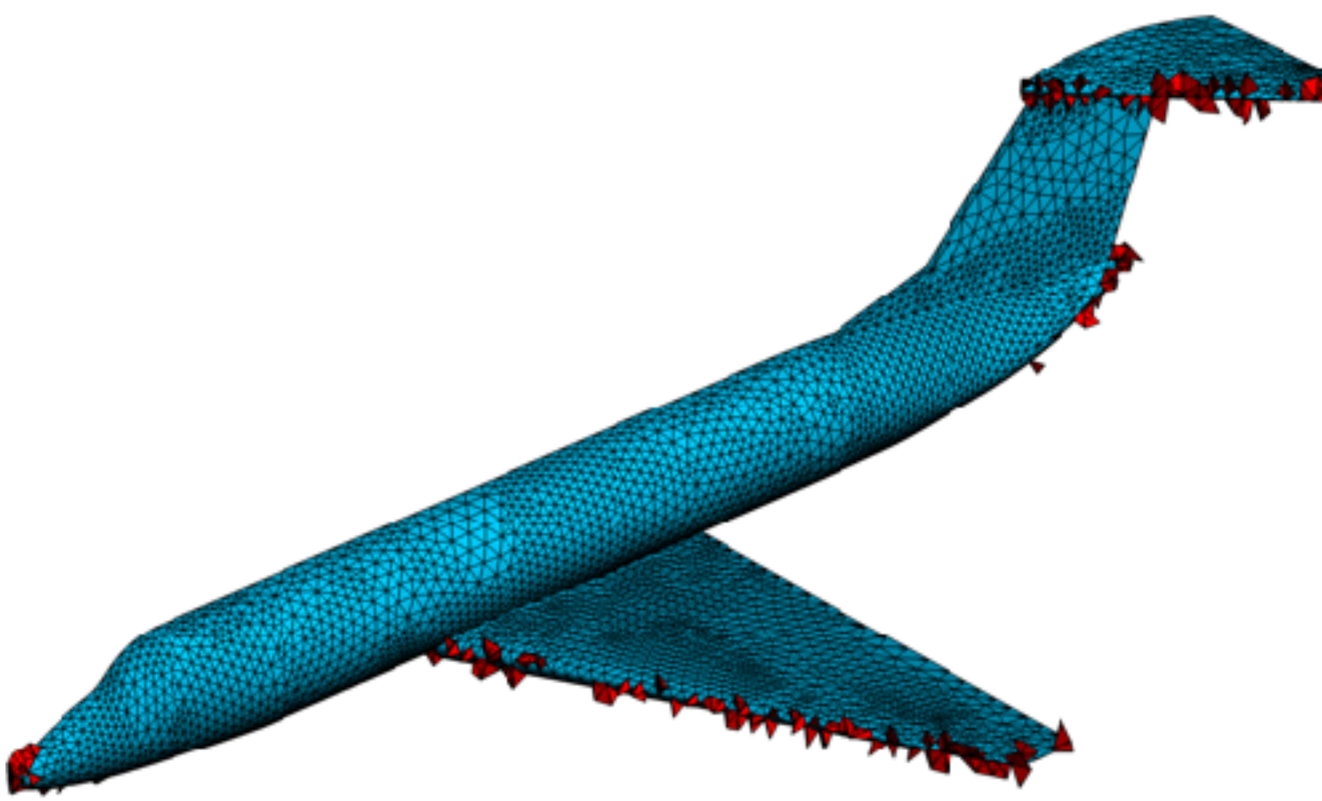
Often surface mesh will never yield valid volume to be generated: solve by sliding elements on the CAD surface



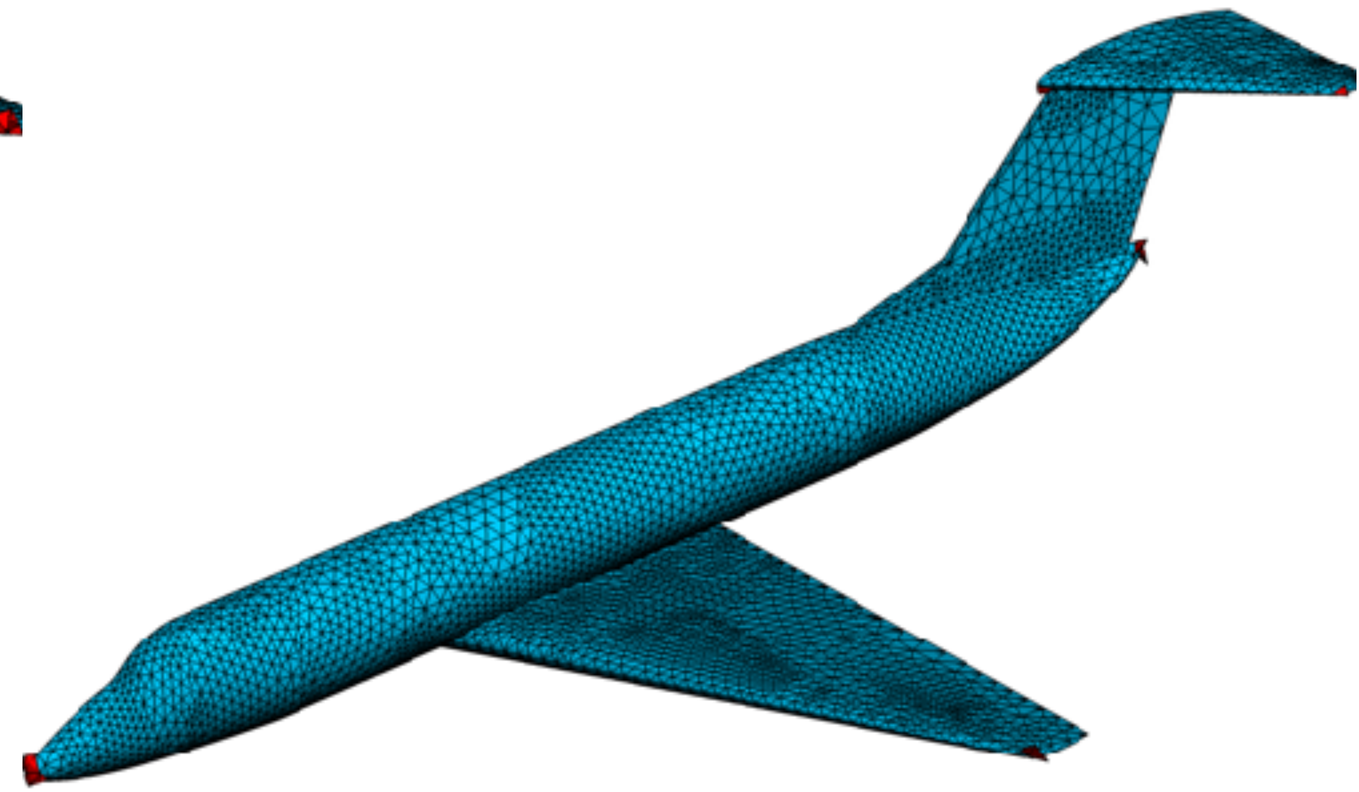
Surface mesh optimisation



Example: jet configuration

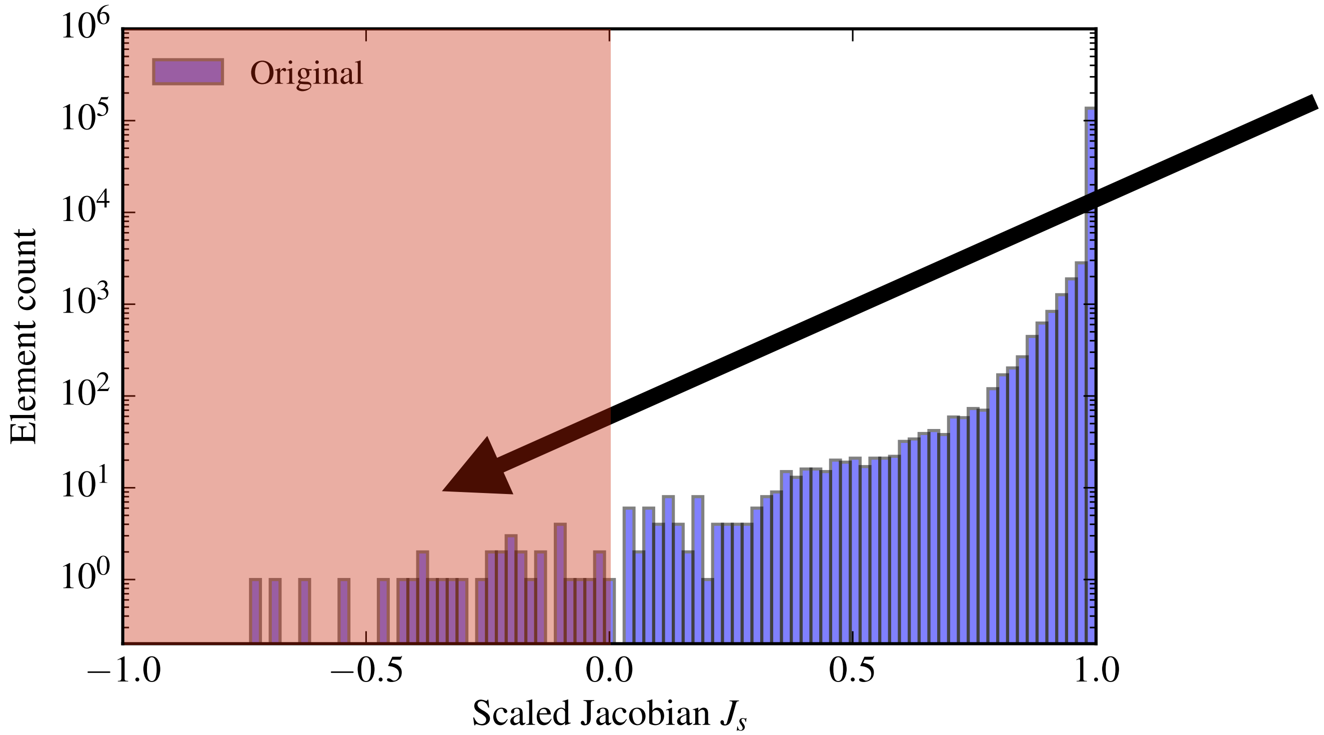


Before optimisation
$J < 0.5$

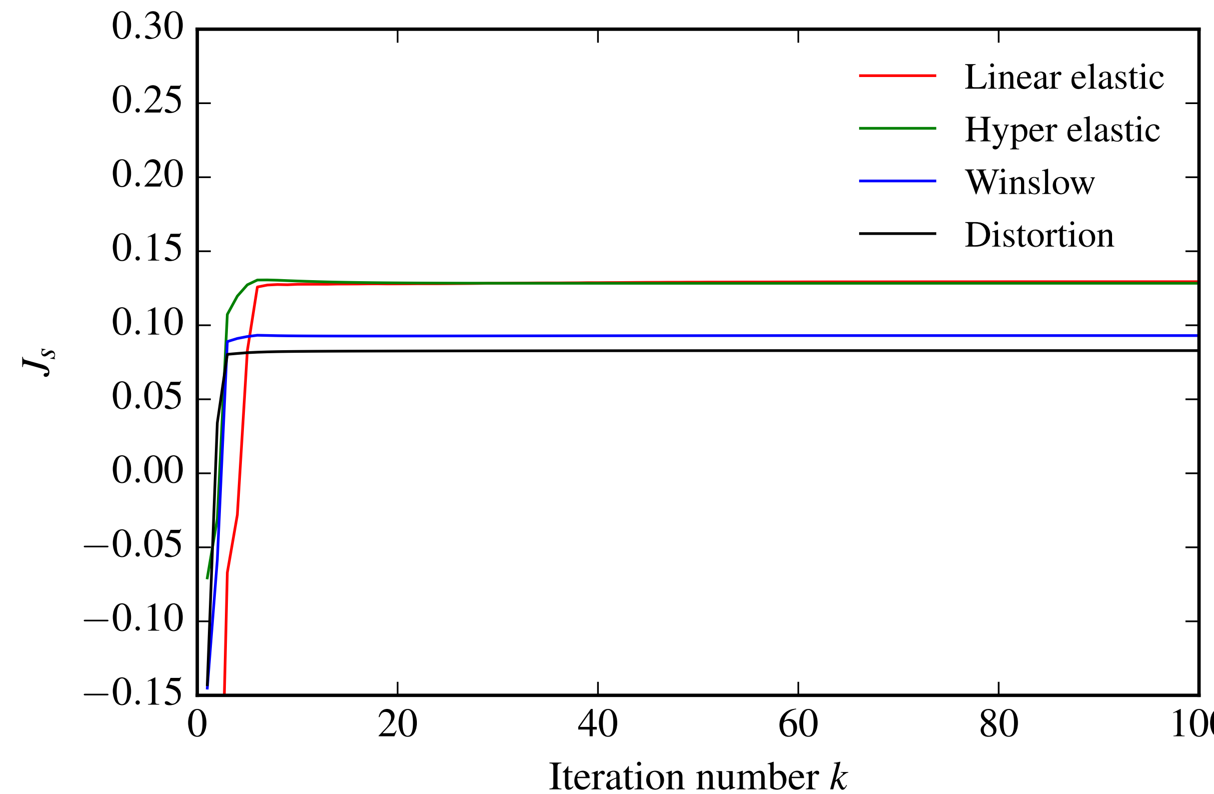


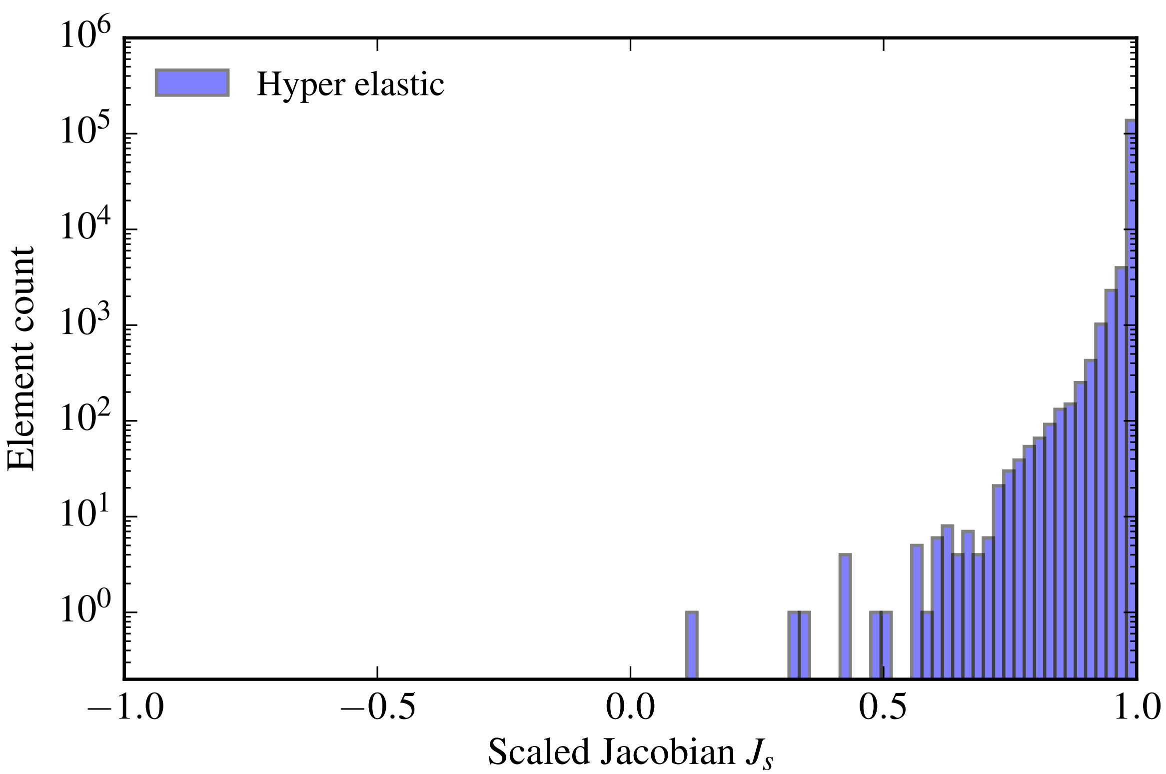
After

Example: jet configuration

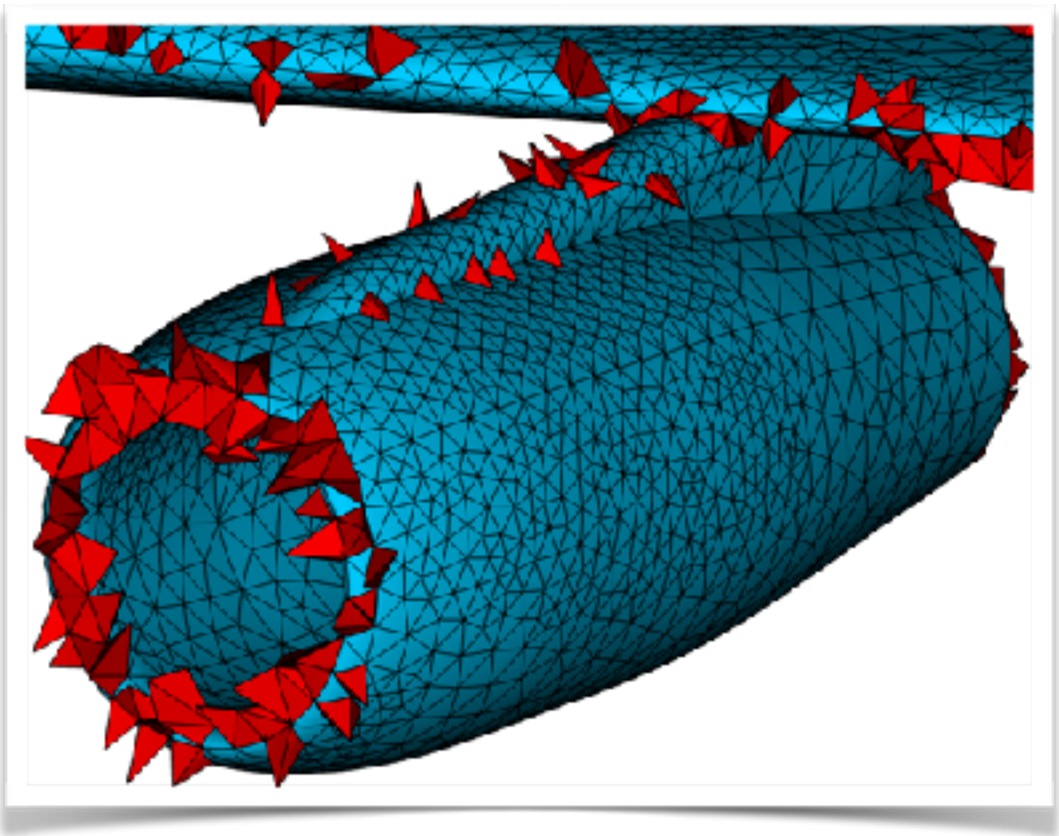


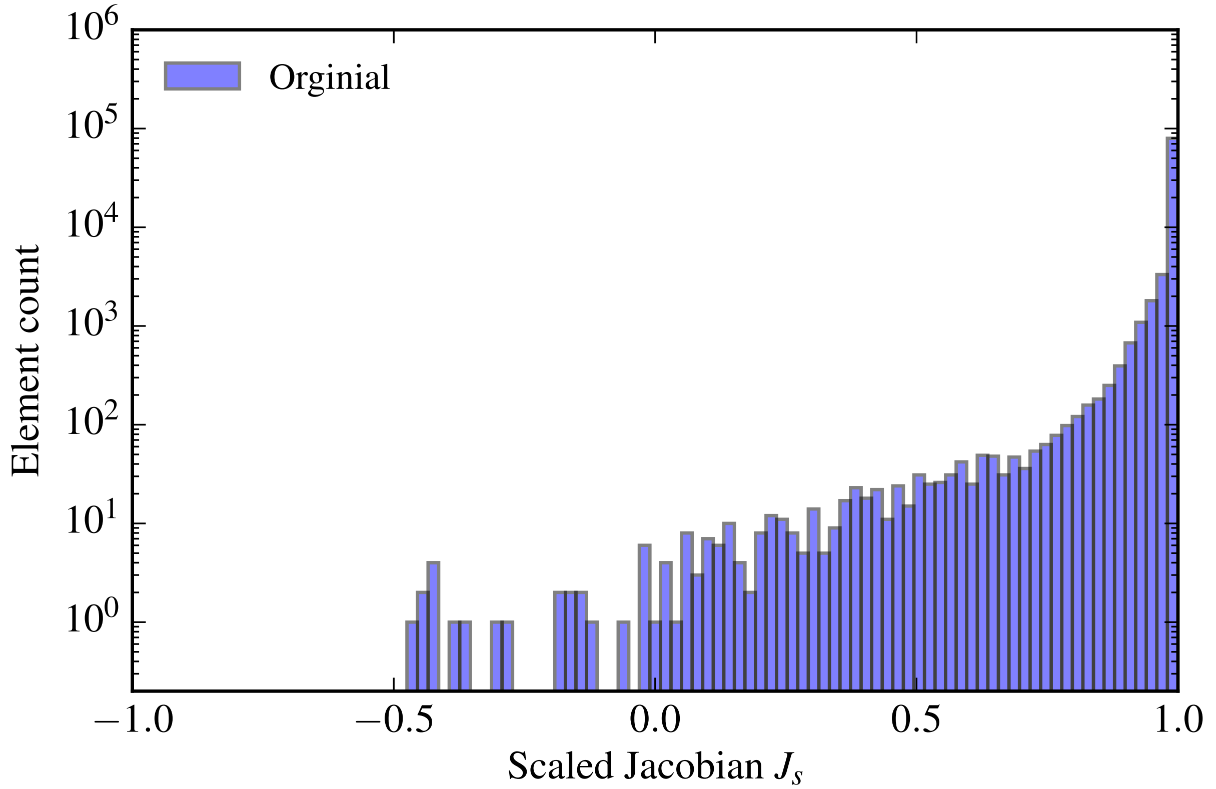
Lots of invalid elements

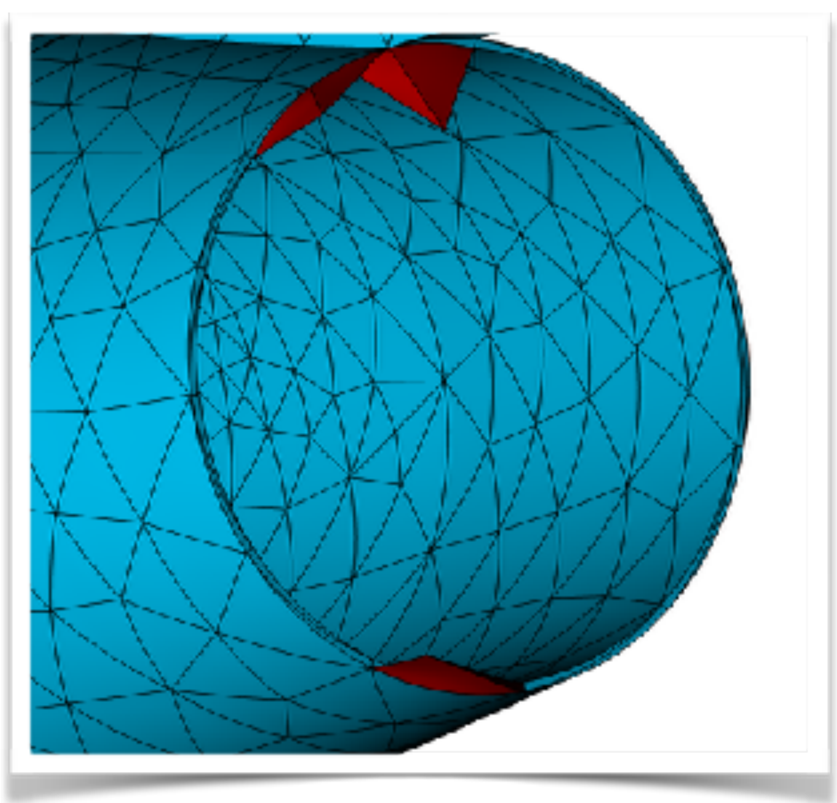


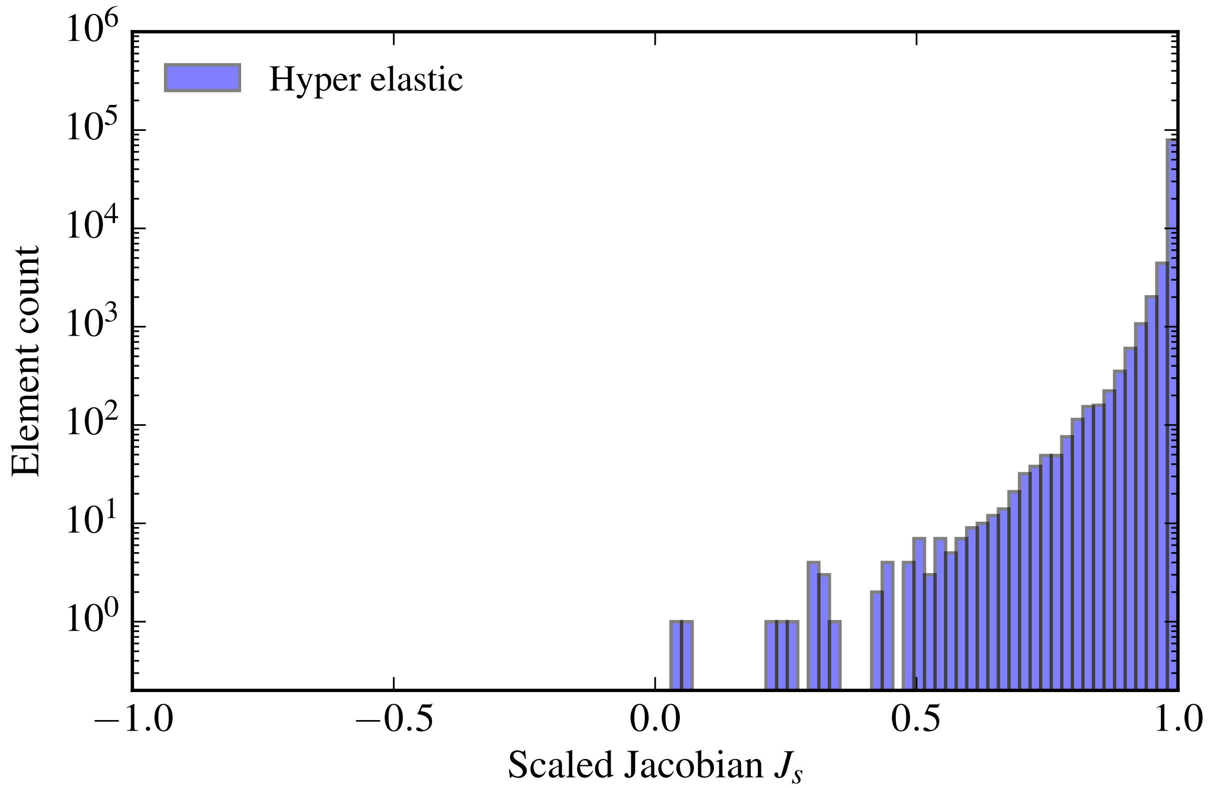


Example: DLR F6 engine

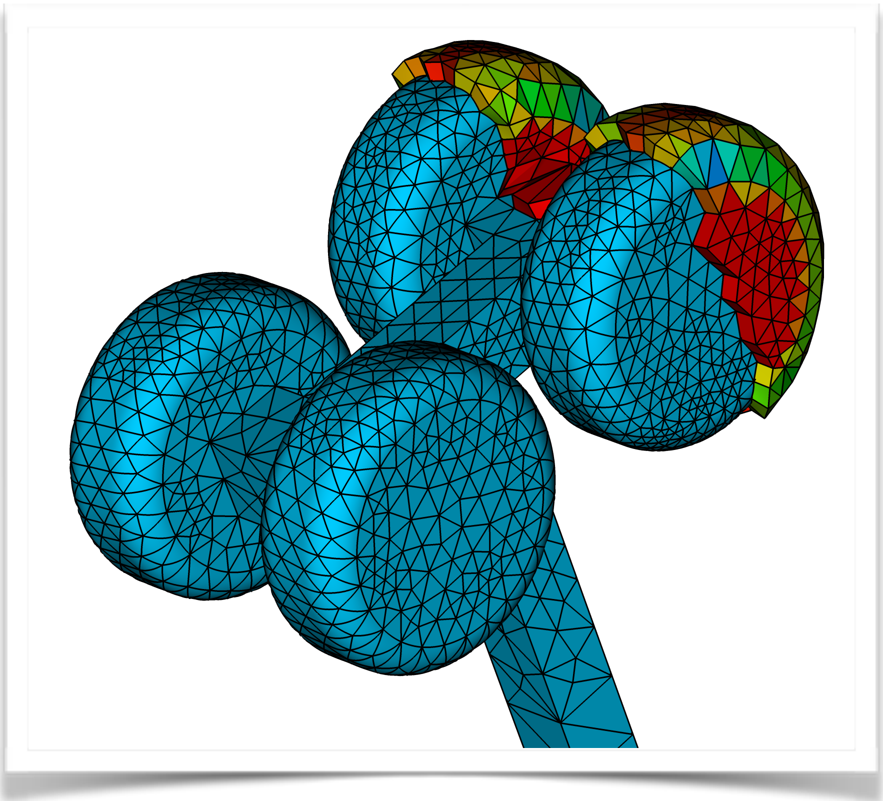


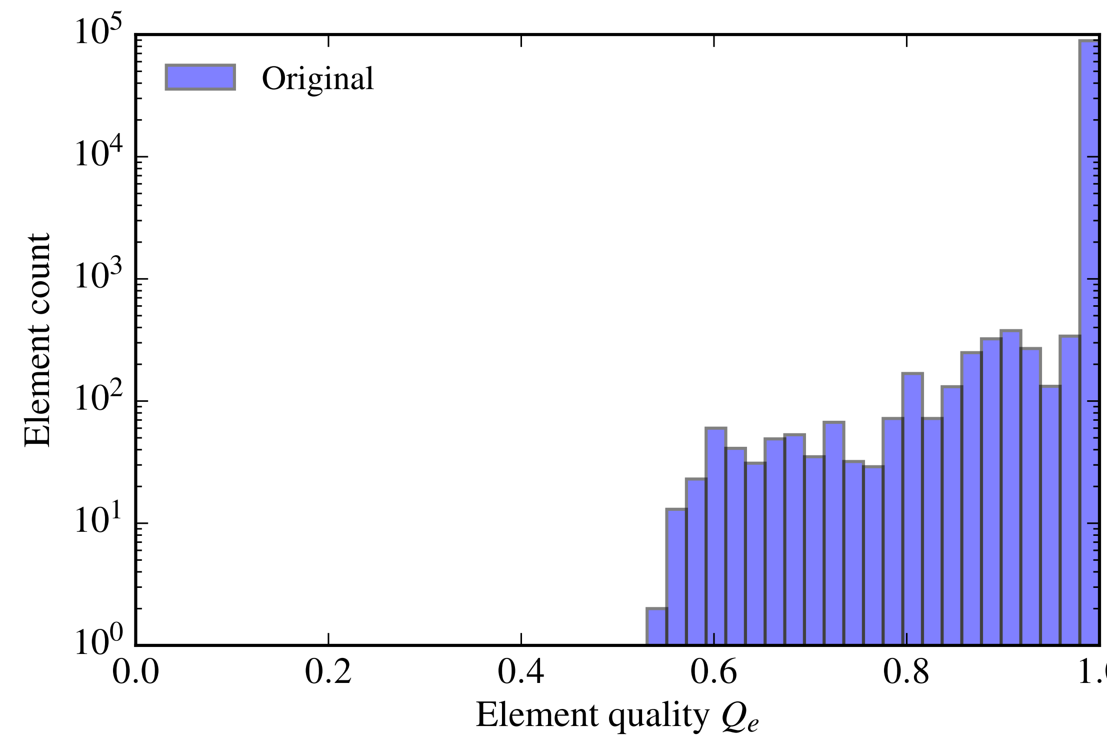


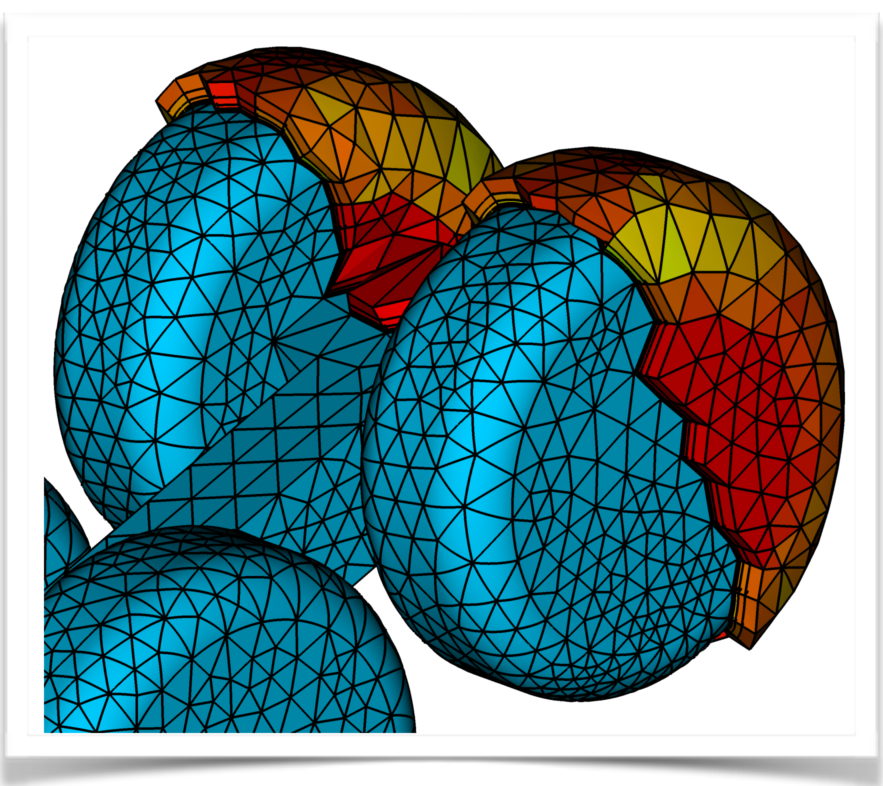


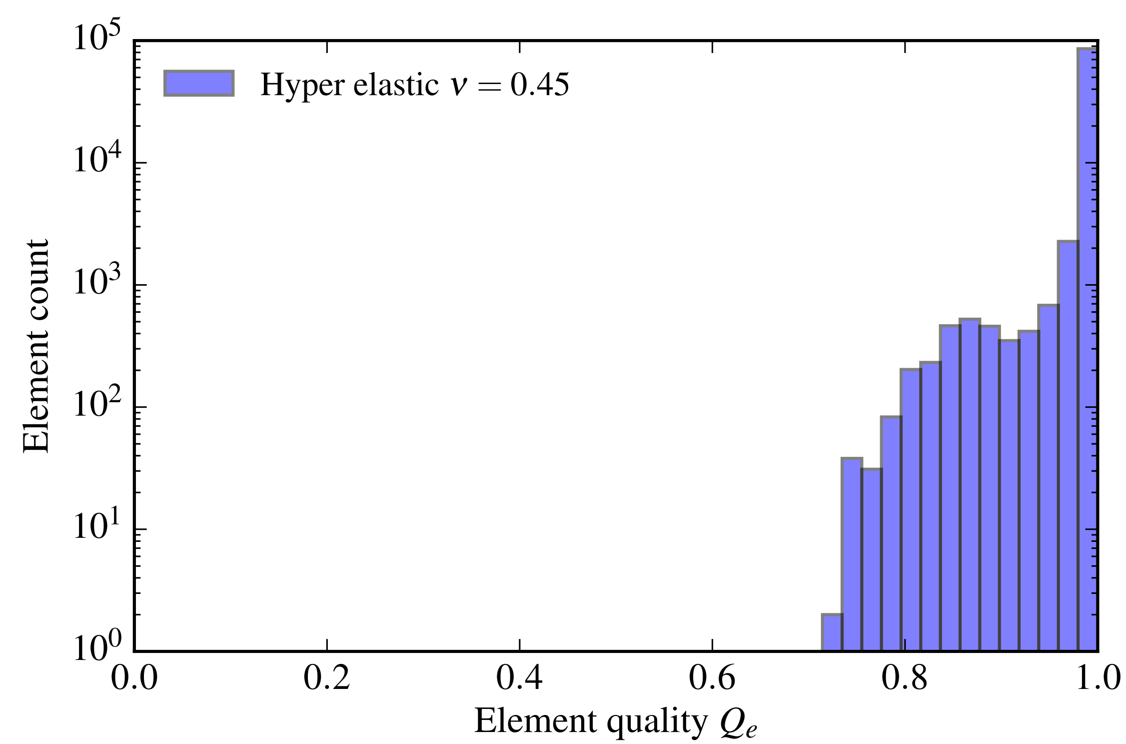


Example: Boeing reduced LG



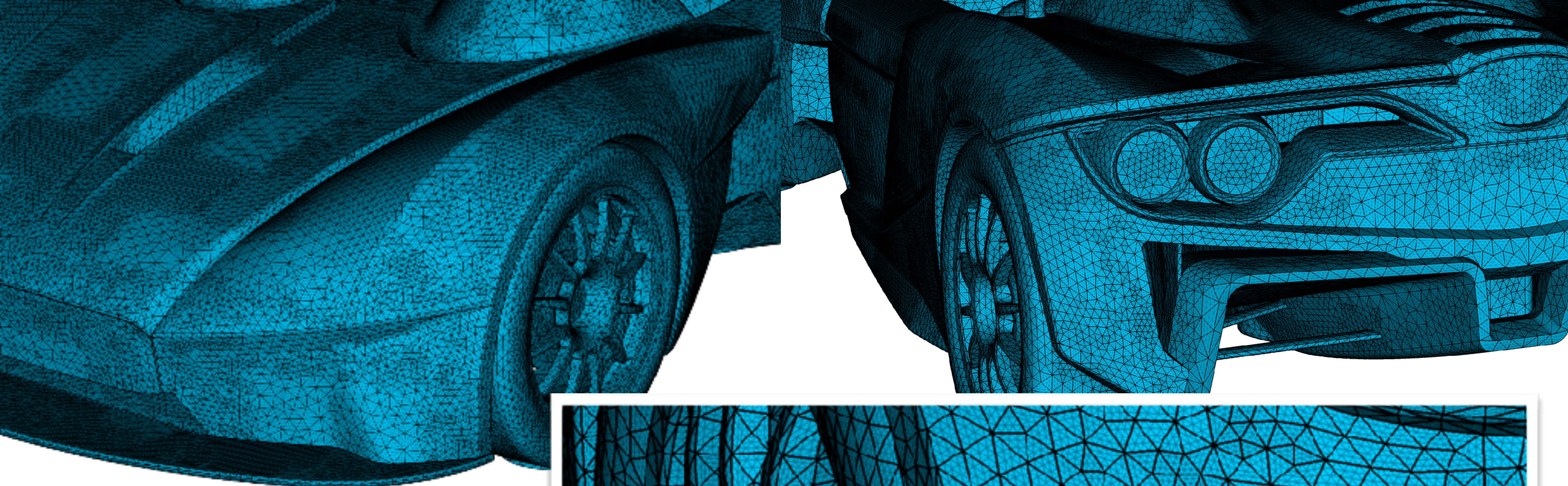






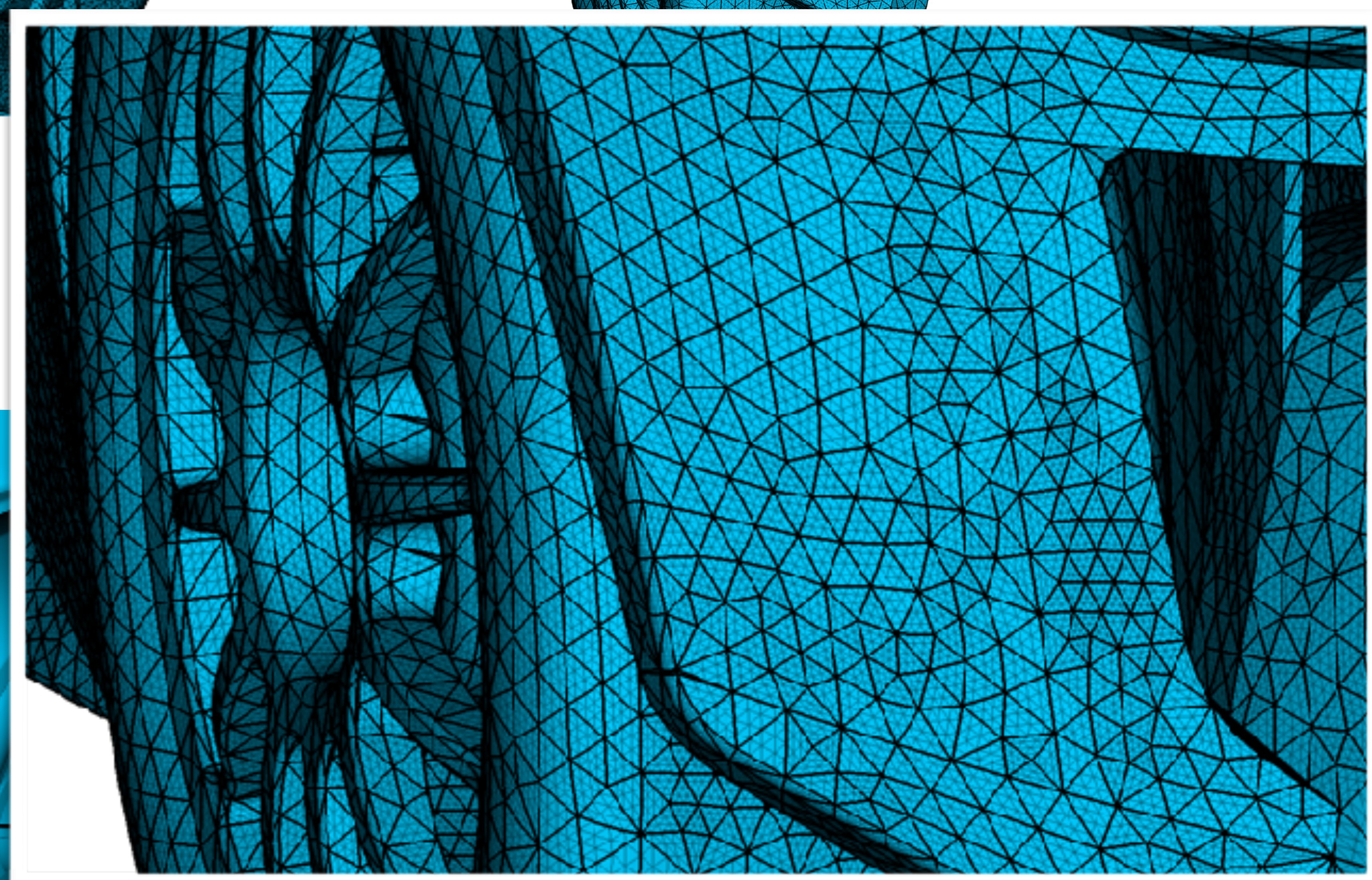
Industrial meshing

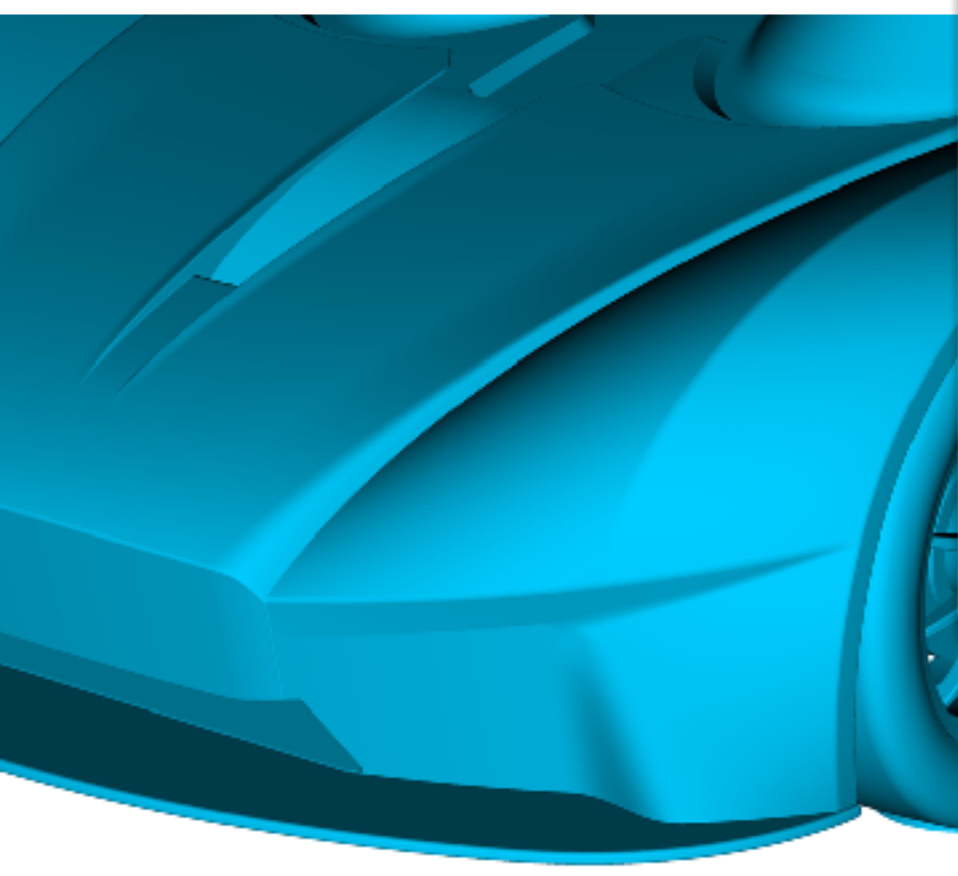
- Lots of improvements for very complex 3D cases where BL generation is a distinct problem
- Improved Star-CCM+ pipeline
- Interaction with other CAD engines (e.g. CFI/CADFix)
- Lots of improvements for really cool geometries that I can't show meshes of



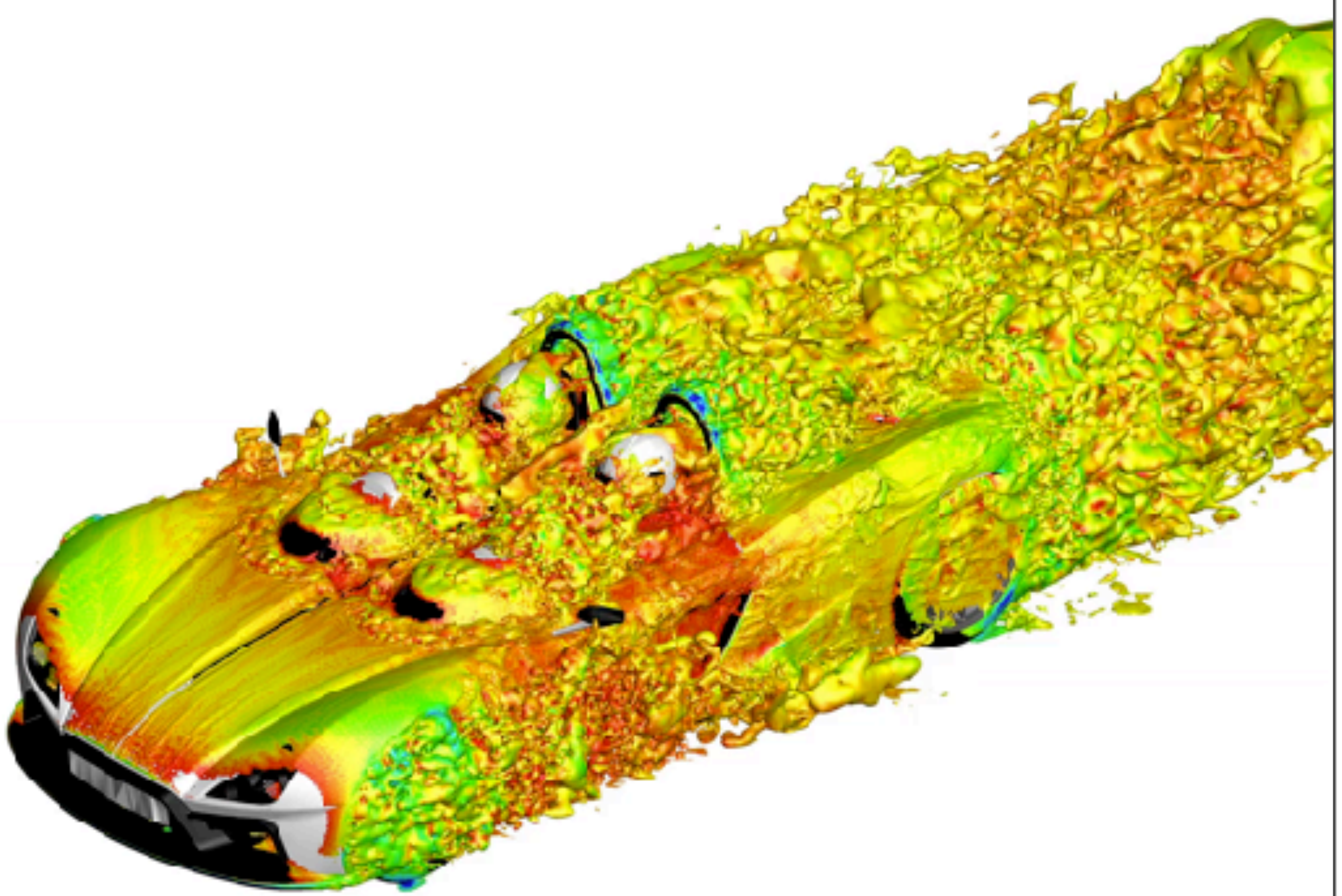
Road car

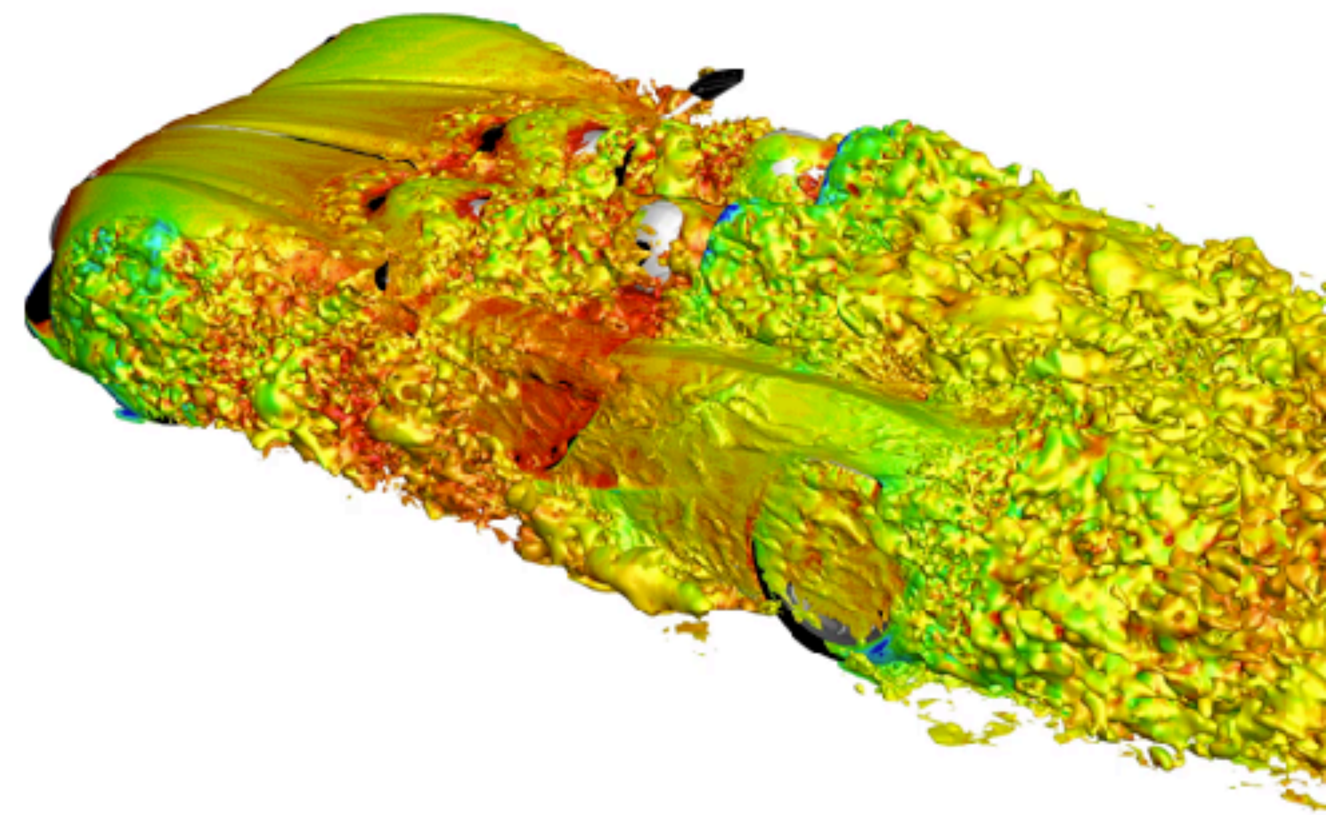
$P = 4$





Simulations





$P = 5$ around 1000 million dof

BL through Star-CCM+ Re = 50k

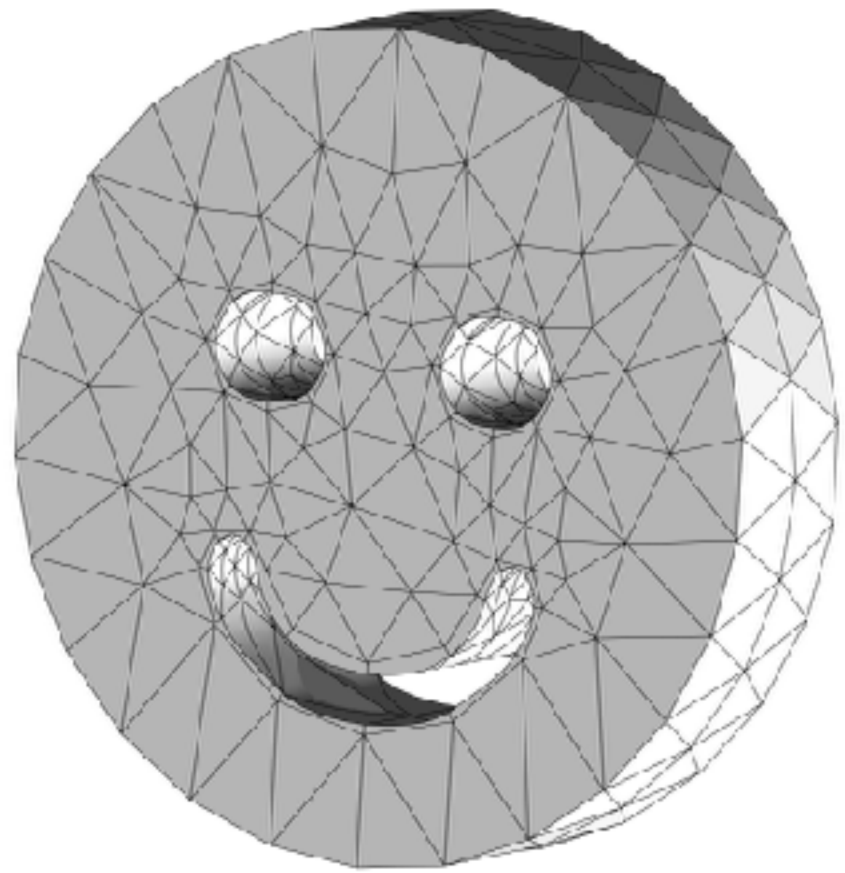
Outlook

- Release of a standalone NekMesh tool for external groups to use
- Further computational enhancements through e.g. GPU acceleration (Jan Eichstadt, Mashy Green)
- Further integration of NekMesh throughout the library, particularly with SpatialDomains
- Aim: on-the-fly mesh movement through variational approach (Julian Marcon)

Conclusions

- NekMesh is really progressing, very few (if any) tools can do what we can
- Really pushing to promote NekMesh as the mesh generator to use with Nektar++
  - If it doesn’t do something that you want it to, tell us and we’ll see if we can do it (no promises)
- 2D mesh generation tutorial now online, so check it out on the ‘open session’ on Friday morning

Thanks for listening!



@davidmoxey

d.moxey@exeter.ac.uk

www.nektar.info