High-order spectral/hp methods for aerodynamic applications

D. Moxey, M. Hazan, S. Sherwin, J. Peiro

Department of Aeronautics,
Imperial College London

ECCOMAS 2012,
12th September 2012

Introduction

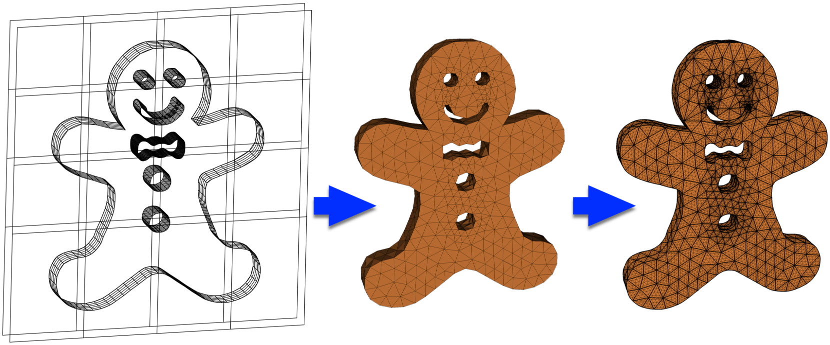
In this talk, we introduce two techniques designed to aid the implementation of high-order methods for industrial application:

- **High-order mesh generation:** generating curvilinear meshes for complex geometries, and a boundary layer refinement technique.
- ***h*-to-*p* efficiency:** a hybrid approach to operator evaluation to increase computational efficiency.

High-order mesh generation (*p*-mesh)

Steps:

1. CAD boundary representation (B-Rep).
2. Mesh of linear elements.
3. Mesh of high-order elements.



B-Rep Linear mesh High-order mesh

Producing meshes for high-Re simulations

Viscous flows $\rightarrow$ boundary layer around walls.

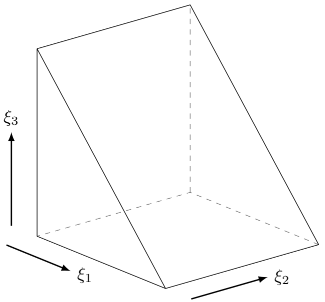
- From the surface triangulation, we generate a prismatic boundary layer (better mesh quality).
- Rest of the volume is constructed using unstructured tetrahedra.

For high Reynolds number simulations:

- Require an extremely thin boundary layer.
- Must not contain invalid elements.

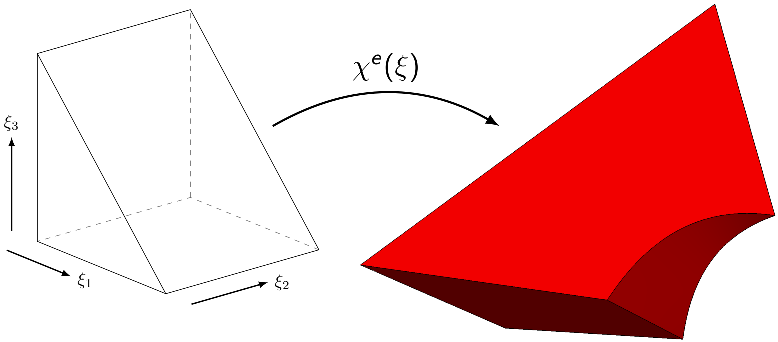
We introduce a method to refine a mesh with existing prismatic boundary layer to produce a valid prismatic or tetrahedralised mesh.

Boundary layer refinement



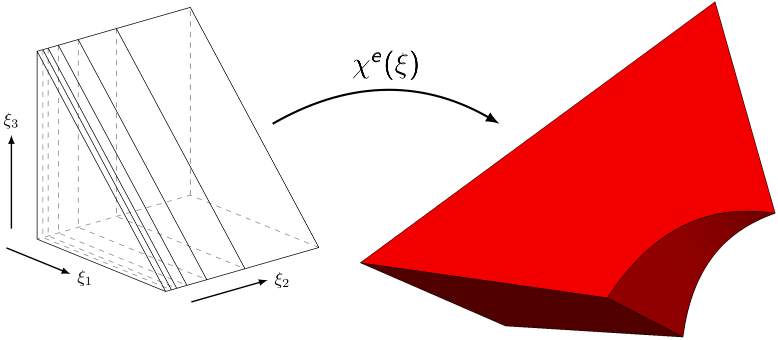
Begin with reference element Ω_{st}.

Boundary layer refinement



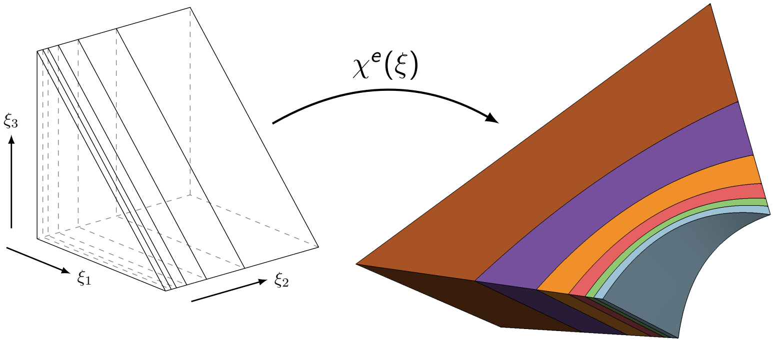
Apply mapping $\chi^e : \Omega_{st} \to \Omega^e$ to obtain Ω^e.

Boundary layer refinement



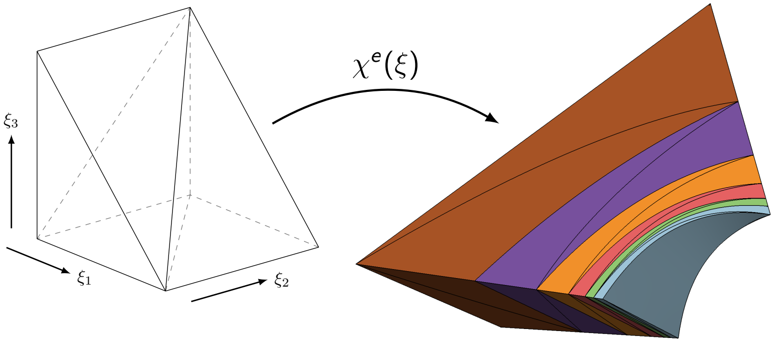
Split Ω_{st} into subelements according to a spacing function $f(\xi_2)$.

Boundary layer refinement



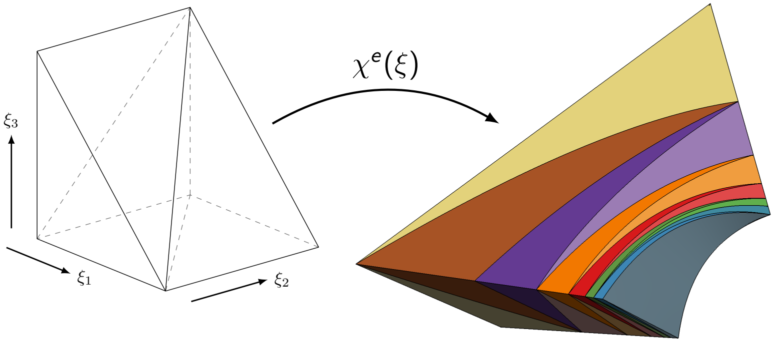
Apply χ^e to split Ω^e into curved subelements.

Boundary layer refinement



To get tetrahedral mesh, split each subelement of Ω_{st} into tetrahedra and apply χ^e.

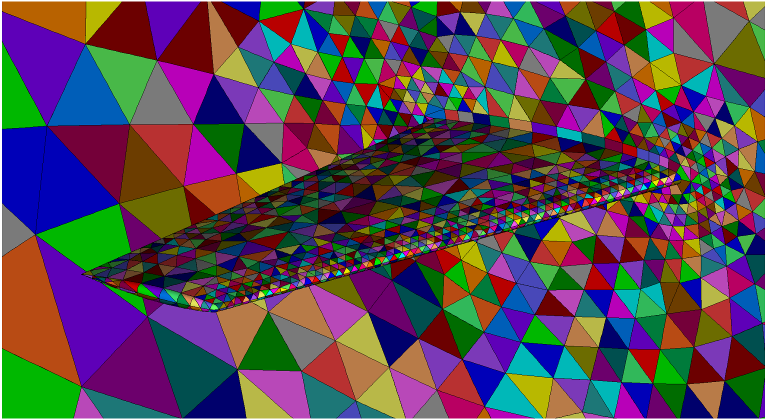
Boundary layer refinement



To get tetrahedral mesh, split each subelement of Ω_{st} into tetrahedra and apply χ^e.

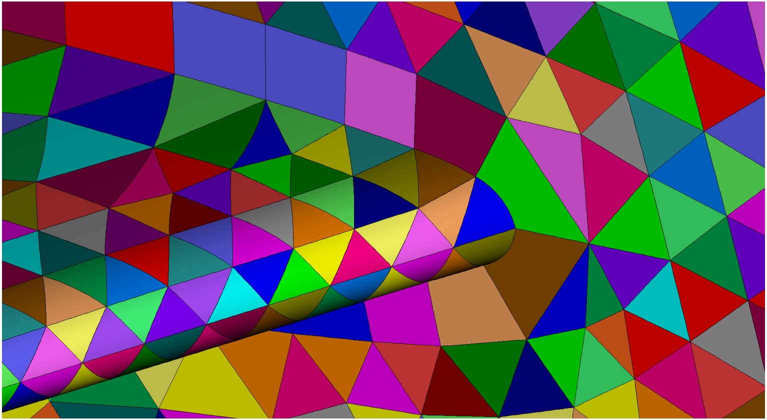
Example: ONERA M6 Wing

Boundary layer thickness: $y^+ \approx 10$ for $\mathrm{Re} = 10^5$.



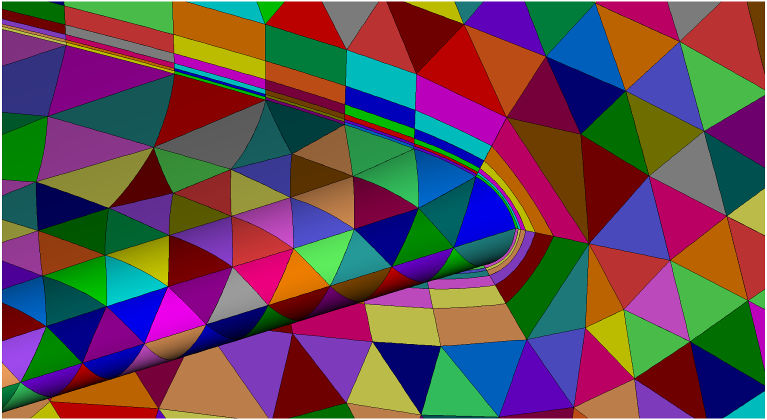
Example: ONERA M6 Wing

Boundary layer thickness: $y^+ \approx 10$ for $\mathrm{Re} = 10^5$.



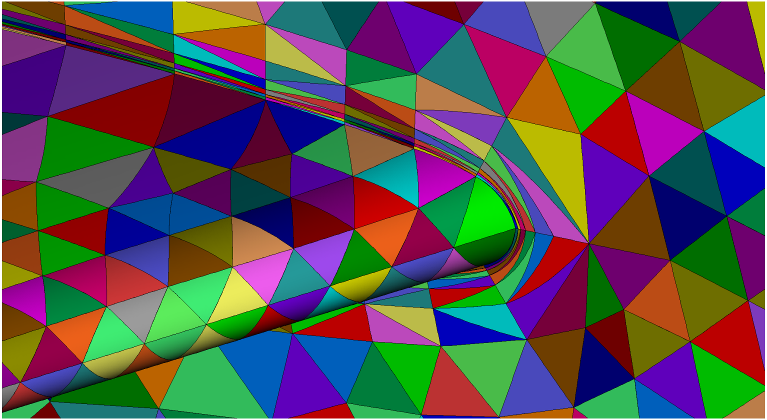
Example: ONERA M6 Wing

Boundary layer thickness: $y^{+} \approx 10$ for $\mathrm{Re} = 10^5$.



Example: ONERA M6 Wing

Boundary layer thickness: $y^+ \approx 10$ for $\mathrm{Re} = 10^5$.



Boundary layer refinement

Features:

- Easy to implement: mapping χ^e is used for spectral element method.
- Produces valid meshes so long as $\det(\chi^e) > 0$.
- Any spacing function $f(\xi_2)$ can be used to determine sizes of subelements.

Boundary layer refinement

Features:

- Easy to implement: mapping χ^e is used for spectral element method.
- Produces valid meshes so long as $\det(\chi^e) > 0$.
- Any spacing function $f(\xi_2)$ can be used to determine sizes of subelements.

Limitation:

- Assumes an existing valid coarse prismatic mesh.

Operator evaluation strategies

Consider the matrix resulting from the discretisation of an operator on a mesh of elements.

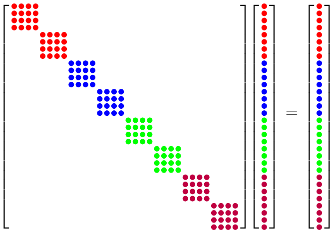
- Assume C^0 continuity of a function approximated on the mesh.
- This can imposed through establishing equality of coefficients representing connected vertices/edges/faces.

Operator evaluation strategies

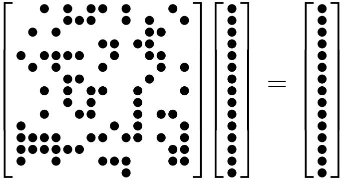
Consider the matrix resulting from the discretisation of an operator on a mesh of elements.

- Assume C^0 continuity of a function approximated on the mesh.
- This can imposed through establishing equality of coefficients representing connected vertices/edges/faces.

There are two well-known operator strategies which can be employed:



Local assembly



Global assembly

Operator evaluation strategies

Which is better? Depends on:

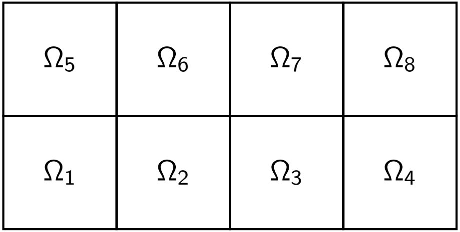
- Polynomial order;
- Architecture (CPU/GPU);
- Dimensionality of mesh.

Generally:

- low order ($P = 1, 2$) $\rightarrow$ global assembly
- high-order $\rightarrow$ local assembly (or sum-factorisation...)

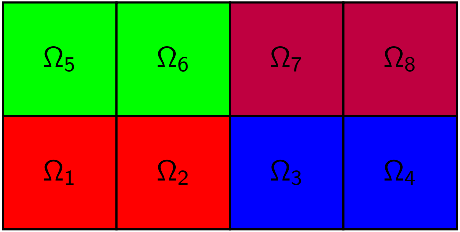
Question: Is there a more optimal strategy which combines a mixture of the two approaches?

Hybrid assembly



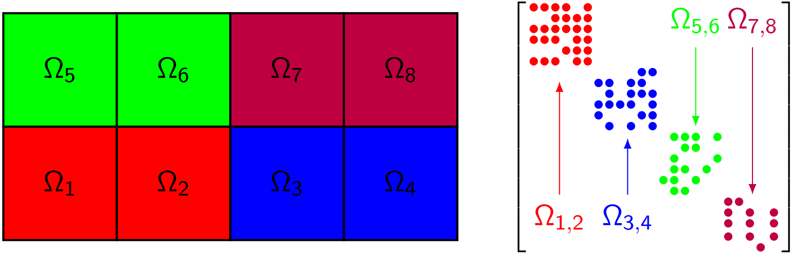
Start with a mesh, for example eight quadrilaterals.

Hybrid assembly



Choose a partition of the mesh.

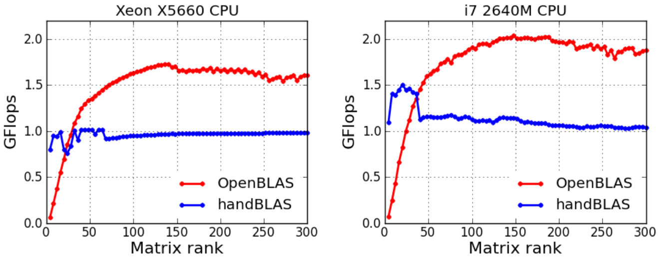
Hybrid assembly



Perform a ‘global’ assembly for each partition to create a *patch*, and constrct a ‘hybrid‘ block matrix.

Why do a hybrid approach?

With hybrid assembly, we can target and optimise for platform specific quantities; e.g. processor cache size.

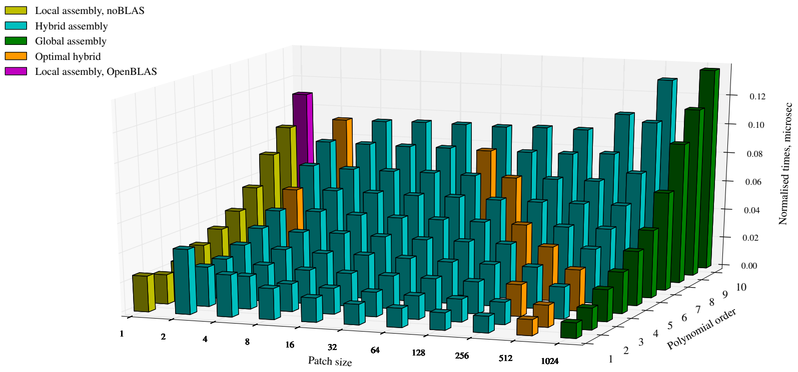


For a given polynomial order we target patch and matrix sizes which optimise CPU throughput using either OpenBLAS or a hand-coded routine.

Some benchmarks

Timings for matrix-vector multiplcation representing an operator on a 1024 triangle mesh were performed.

The following figure shows the optimal techniques for a variety of patch sizes and polynomial orders.



h-to-p efficiency

Generally these timings suggest that:

- Low-order $\rightarrow$ global assembly (1 patch);
- High-order $\rightarrow$ local assembly (N_{el} patches);
- For middle-sized polynomial orders there is some more optimal choice of patch size.

h-to-*p* efficiency

Generally these timings suggest that:

- Low-order $\to$ global assembly (1 patch);
- High-order $\to$ local assembly (N_{el} patches);
- For middle-sized polynomial orders there is some more optimal choice of patch size.

Choosing these parameters manually is difficult. *libFEMpp*:

- will be a platform- and code-independent library;
- combine system configuration and profiling data to produce optimal patch sizes for a given input mesh;
- is currently under development.

Conclusions

Mesh generation:

- Generation of arbitrary high-order meshes is a viable goal for complex geometries.
- Use of refinement technique aides in the generation of boundary layer meshes for aeronautically realistic Reynolds numbers.

h-**to**-*p* **efficiency:**

- Careful choice of operator evaluation strategies is essential for high-order elements.
- Hybrid assembly demonstrates an approach whereby this process can be automated and tuned for individual architectures.