ExCALIBUR 10

2D MESHING FOR TOKAMAK FUSION SIMULATION

Ed Threlfall

Zoom, 28th January 2021

UK Atomic Energy Authority

The need for accurate meshing

- Finite-difference approaches traditionally use rectilinear grid of sample points …
- Differences typically second-order accurate, error e.g. $\sim h^2$ for grid spacing h.
- Naïve versions have problems with non-grid-aligned boundaries (though more advanced geometrical schemes are possible).

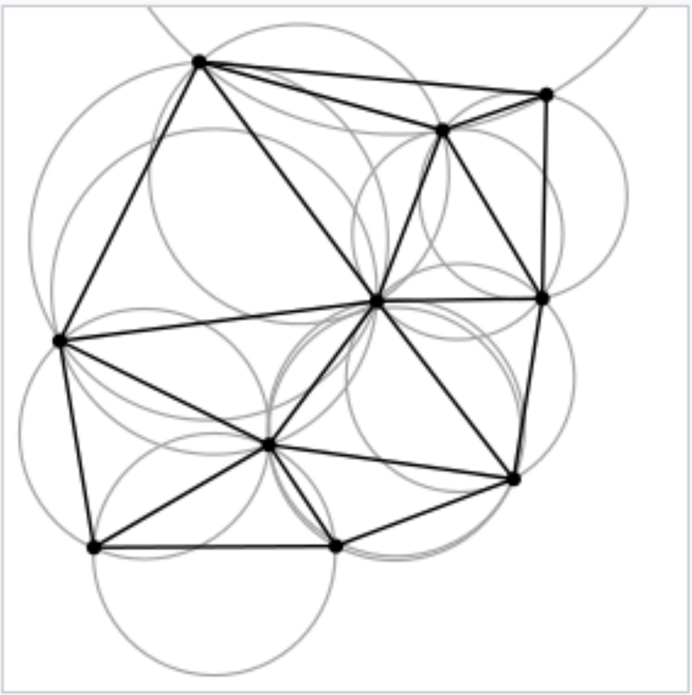
- Finite-element approaches allow domain decomposition into elements of arbitrary shape … element boundaries can conform to constraining geometry.
- Increasing number of intra-element degrees of freedom (p) gives exponential convergence, error $\sim h^p$ (decrease h or increase p for higher accuracy).
- Many subtleties, e.g. high order permits use of distorted elements …

- Notes on finite-element method: see the NEPTUNE docs repository: *https://github.com/ExCALIBUR-NEPTUNE/Documents/tree/main/tex/note_on_finite_elements*.

- Additionally, current supercomputer architectures well-suited to the mathematical structure of higher-order codes e.g. large number of operations per unit of element data (called *arithmetic intensity*, as in Exeter's call).

Meshing considerations

- Geometrical constraints lead one to consider unstructured meshing; a strategy to optimize the mesh is to use the Delaunay property in 2D or 3D (it maximizes the minimum angle) for triangles / tetrahedra.
- Unstructured mesh generators typically refine a triangulation based on certain prerequisites (primarily the local element size e.g. max area in 2D) - h-refinement.
- Generally, all elements in simulation being of similar size makes numerics easier …
- \`Small' elements worsen the condition number for a time-implicit method (preconditioners help), or give small timestep for a time-explicit method (constrained by stability: Courant limit) .
- Heuristically, \`smallness' measured by the maximum size inscribed circle of an element (so squashed elements count as small).

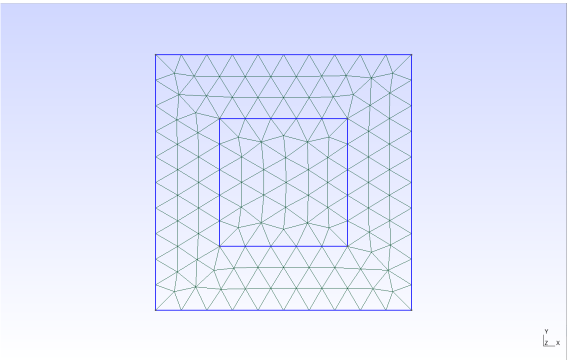
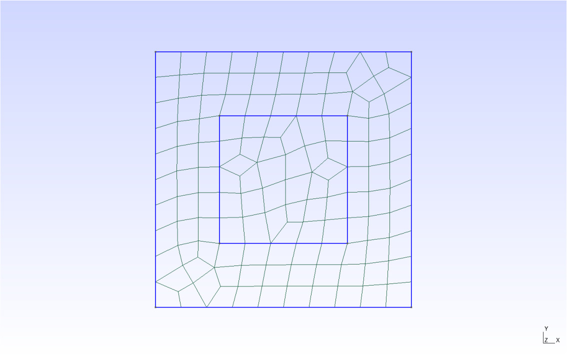


A Delaunay triangulation in the plane with circumcircles shown

From Wikipedia *Delaunay triangulation*.

Some community meshing tools

- Gmsh: CAD modelling with GUI, 2D triangulations, 3D tetrahedralizations. Supports higher-order (i.e. curved) meshing. Able to convert 2D triangle mesh to irregular quad mesh.

- Triangle: 2D triangulations, straight-sided only. Can be used as library, or via CLI with command-line arguments to control mesh quality. Older-style procedural C code but nicely commented. Python wrapper available.
- Tetgen: 3D tetrahedralizations, straight-edged only. I've found it harder to follow than Triangle. Can be used as library or via CLI.
- Nekmesh: geometry engine included in Nektar++. Uses Triangle and Tetgen to do discretizations. Has optimizations to generate and optimize curved meshes, boundary layer refinement …
- … others exist e.g. fTetWild tetrahedral mesher…

Constraints for tokamak meshes (1)

- Typical geometry is a complicated `first wall' made out of various materials … this needs to be meshed accurately e.g. normals to sub-degree fidelity, ultimately.
- This is a complex 3D system (Be tiles, ducts for neutral beam injection, sensing antennas, tritium breeders …), which we do not target at this proxyapp stage.

- There are also magnetic flux surfaces contained by the first wall. (Magnetic field lines spiral around these …)

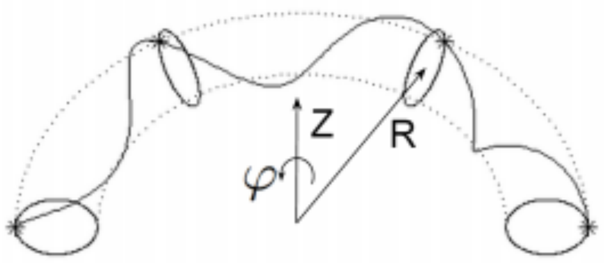


Fig. 6: Magnetic field line on a closed magnetic flux surface.

From Zhang et al. *Mesh Generation for Confined Plasma Simulation*, 2015.

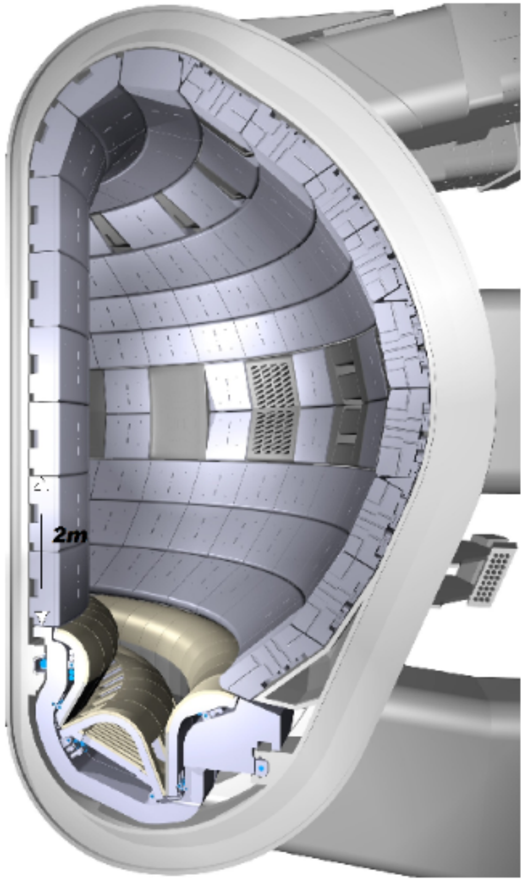


Figure 1.1: The complexity of the ITER first wall

From *ExCALIBUR NEPTUNE M2.1.1 Report: Options for Geometry Representation*.

Constraints for tokamak meshes (2)

- Focus on 2D poloidal cross section of tokamak.

- The region within the first wall contains a central region containing closed, nested magnetic flux surfaces (plasma core).
- There is a last closed flux surface, called separatrix. Outside are non-closed magnetic flux surfaces, in a region called the scrape-off layer. This contains plasma and neutrals that interact with the first wall.

- Aside (3D only): additional constraint from need to mesh acute incidence of magnetic flux surfaces on first wall; designed this way to spread power deposition over a wider area to avoid damage (2 degrees is mentioned in M2.1.1 Report, figure illustrating problem uses 9.5 degrees).

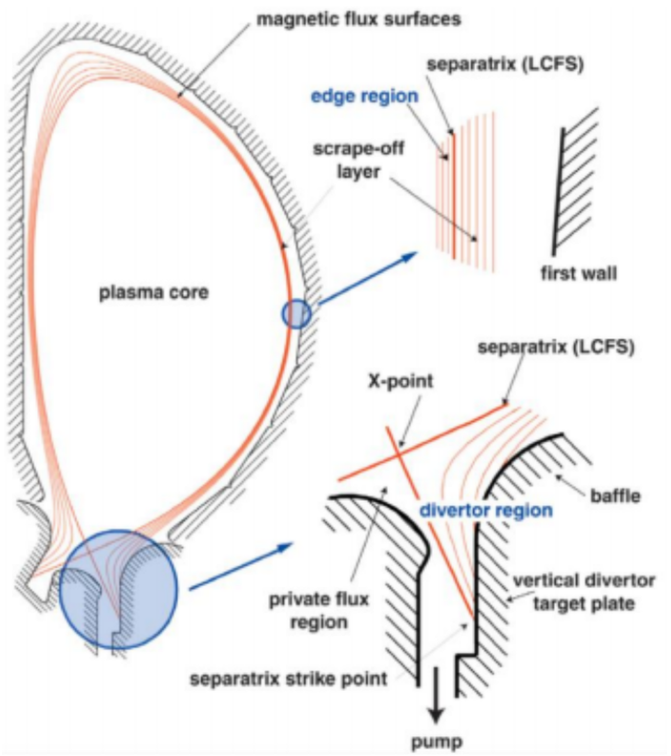


Figure 1: Schematic diagram of a generic tokamak "poloidal cross section" showing the areas of plasma and first wall that will be targeted by project NEPTUNE (shaded circles). Attribution: G. Federici et al. [CC BY 3.0 (https://creativecommons.org/licenses/by/3.0)].

From *ExCALIBUR Fusion Modelling System Science Plan.*

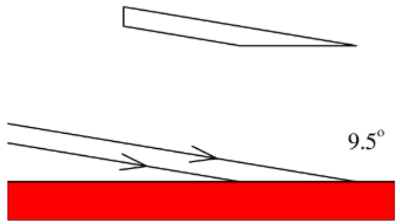
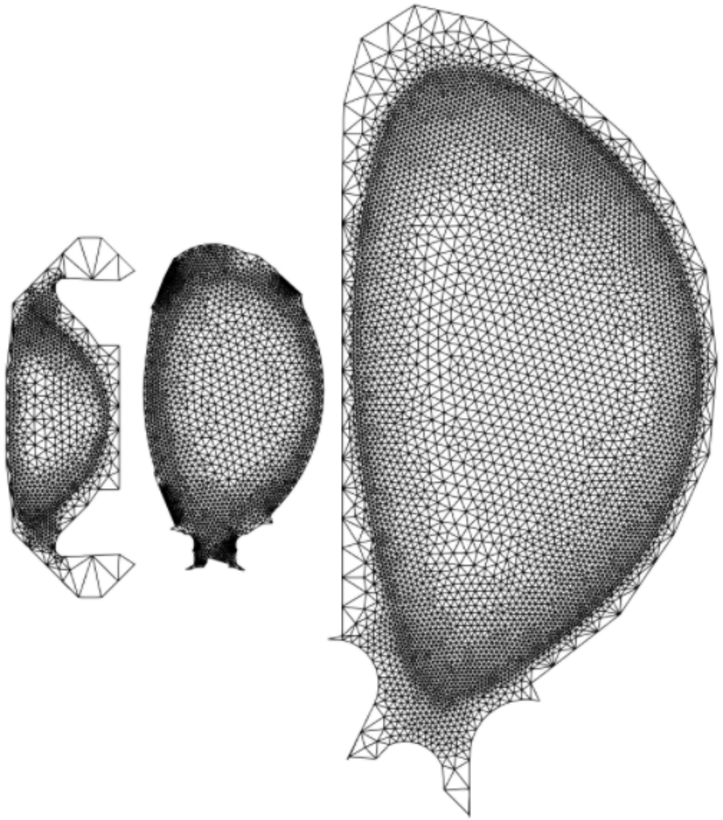


Figure 2.8: Undesirable element shape if meshing conforms to both surface and fieldline.

From *ExCALIBUR NEPTUNE M2.1.1 Report: Options for Geometry Representation.*

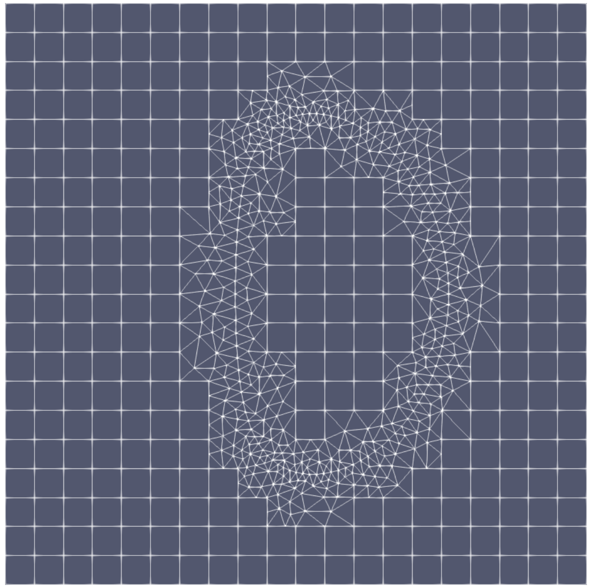
Tokamak cross-section meshes (1)

- Cross-sections of MAST-Upgrade, JET, ITER meshed in Gmsh by Stan Pamela (UKAEA).
- Also provides Python script for converting EQDSK file to Gmsh .geo file.
- Adjustable parameters: location of core-edge boundary; resolution of SOL, separatrix, core plasma.

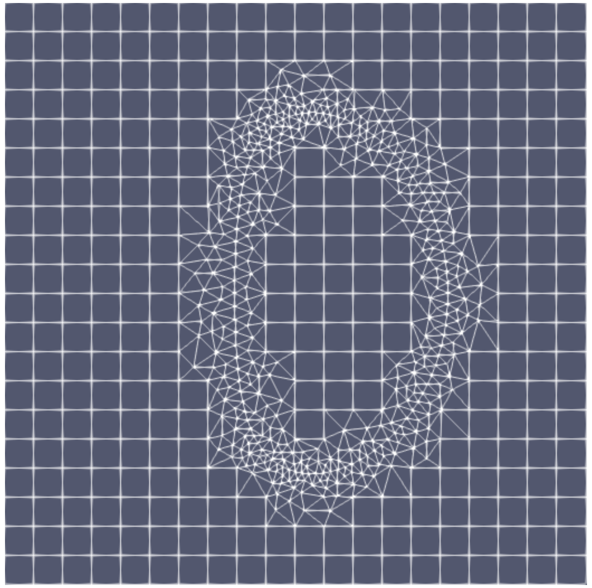


Tokamak cross-section meshes (2)

- One technique to speed up codes is to replace triangles with quads.
- Using grid-aligned quads makes certain FEM implementations very sparse.
- Simple examples: set up regular quad grid, remove quads near constraining geometry, use boundary of extant quads as additional inputs for Triangle. (The D-shapes were generated from circles using the Joukowsky transform – toy model.)



triangle_shewchuk pa0.1q31.5 reactor.poly



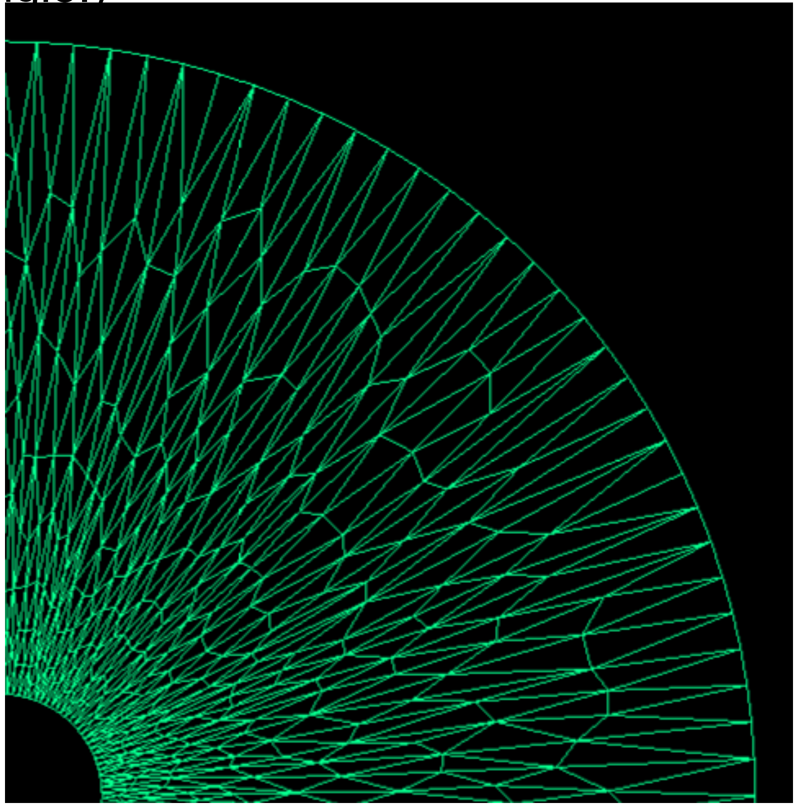
triangle_shewchuk pa0.1q31.5Y reactor.poly

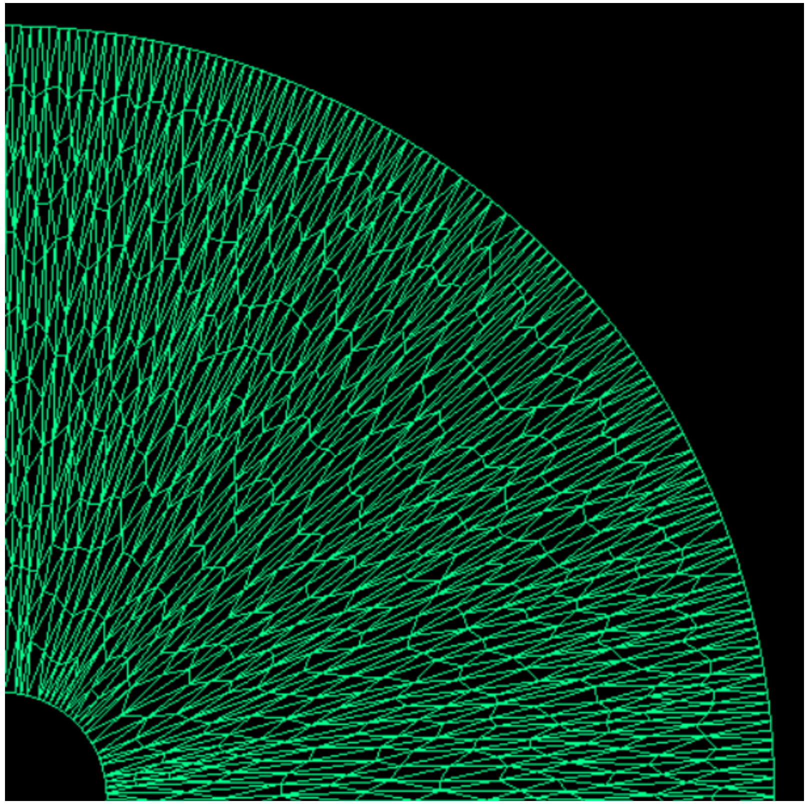
2D anisotropic meshes (1)

- T/NA078 Deliverable 2.1 is for a 2D solver for anisotropic heat transport; spatial anisotropy can be very large (10^5 $-$!!).
- Isotropic Delaunay meshes are unlikely to serve us well in this case, due to need to resolve directions with greater diffusivity more finely.
- While it's trivial to generate globally anisotropic meshes (just squash!), we will need anisotropy to correspond to magnetic flux surfaces …
- One way to do this is to recast the 2D coordinates in the isotropic gauge: $ds^2 = f(x_1, x_2)\left(dx_1^2 + dx_2^2\right)$ and do Delaunay triangulation in those coordinates … the result has a position-dependent refinement level, but this can be removed by altering the area constraint of the triangulator.

2D anisotropic meshes (2)

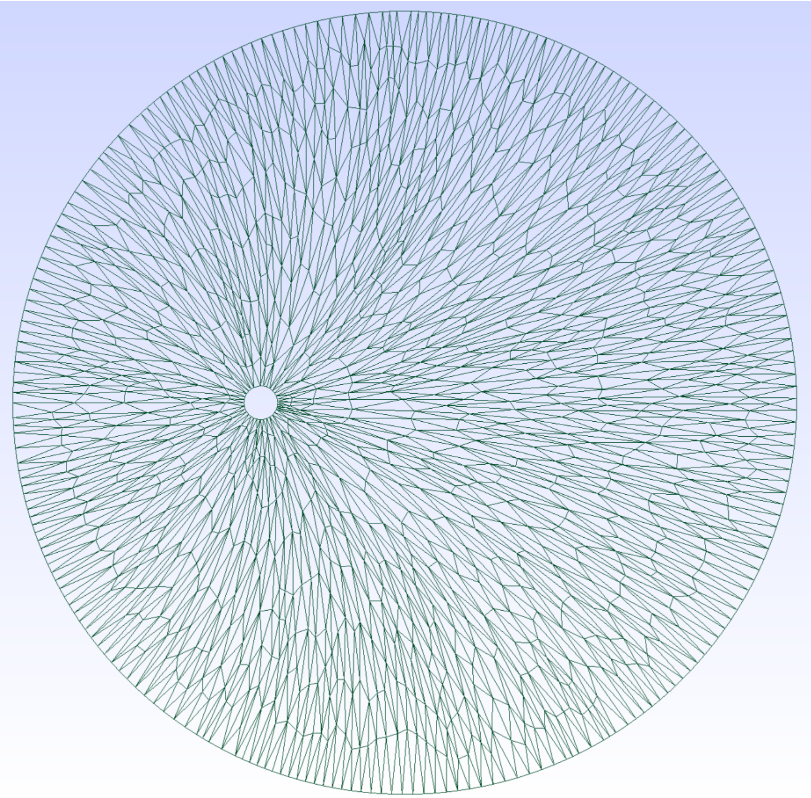
- Example: 2D polar coordinates, $ds^2 = dr^2 + r^2 d\varphi^2$
- Isotropic gauge: $ds^2 = e^{2u}(du^2 + d\varphi^2)$
- Then modify coordinates to $ds^2 = e^{2u}(du^2 + \alpha^2 d\varphi^2)$ to introduce anisotropy …
- … then hack triangulator to make density uniform. (This is a few lines of code in void testtriangle(…) of Triangle.)

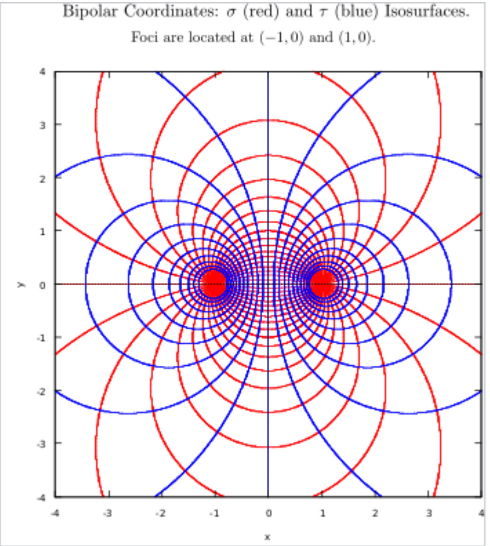




2D anisotropic meshes (3)

- Example is part of the plane in bipolar coordinates …
- This approach has potential problem at X-point … and also is quite specific to 2D (in which infinitely many sets of locally-orthogonal coordinates exist – related by holomorphic maps).





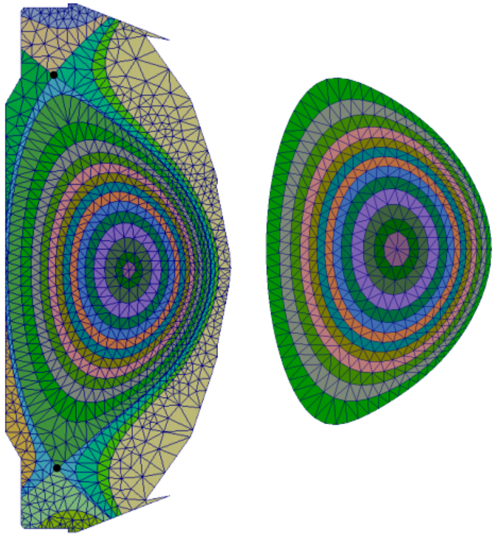


Fig. 12: Example mesh with two X-points (2,878 elements) and no X-point (1,030 elements).

Alternative …
From Zhang et al. *Mesh Generation for Confined Plasma Simulation*, 2015.

Summary

- Use of higher-order methods necessitates geometry-conforming meshing.

- Several community meshing tools are available.

- Tokamaks (can sometimes) simplify to 2D poloidal plane sections; geometry includes first wall and probably some magnetic flux surfaces.

- 2D anisotropic meshes can be generated for arbitrary 2D coordinatizations of the plane using a simple technique.