Numerical and computational performance of spectral element methods for prototype fusion problems

David Moxey & Mashy Green

Department of Engineering, King's College London

Bin Liu, Chris Cantwell & Spencer Sherwin

Department of Aeronautics, Imperial College London

UKAEA Workshop, Abingdon, 5th September 2022

Overview & background

Overview

- As part of NEPTUNE, we have been investigating the application of high-order methods within prototype fusion problems.
- The work undertaken thus far focuses on three areas:

High-order mesh generation

- How do we generate meshes suitable for fusion applications?
- Consider surface mesh accuracy, anisotropy and geometric complexity.

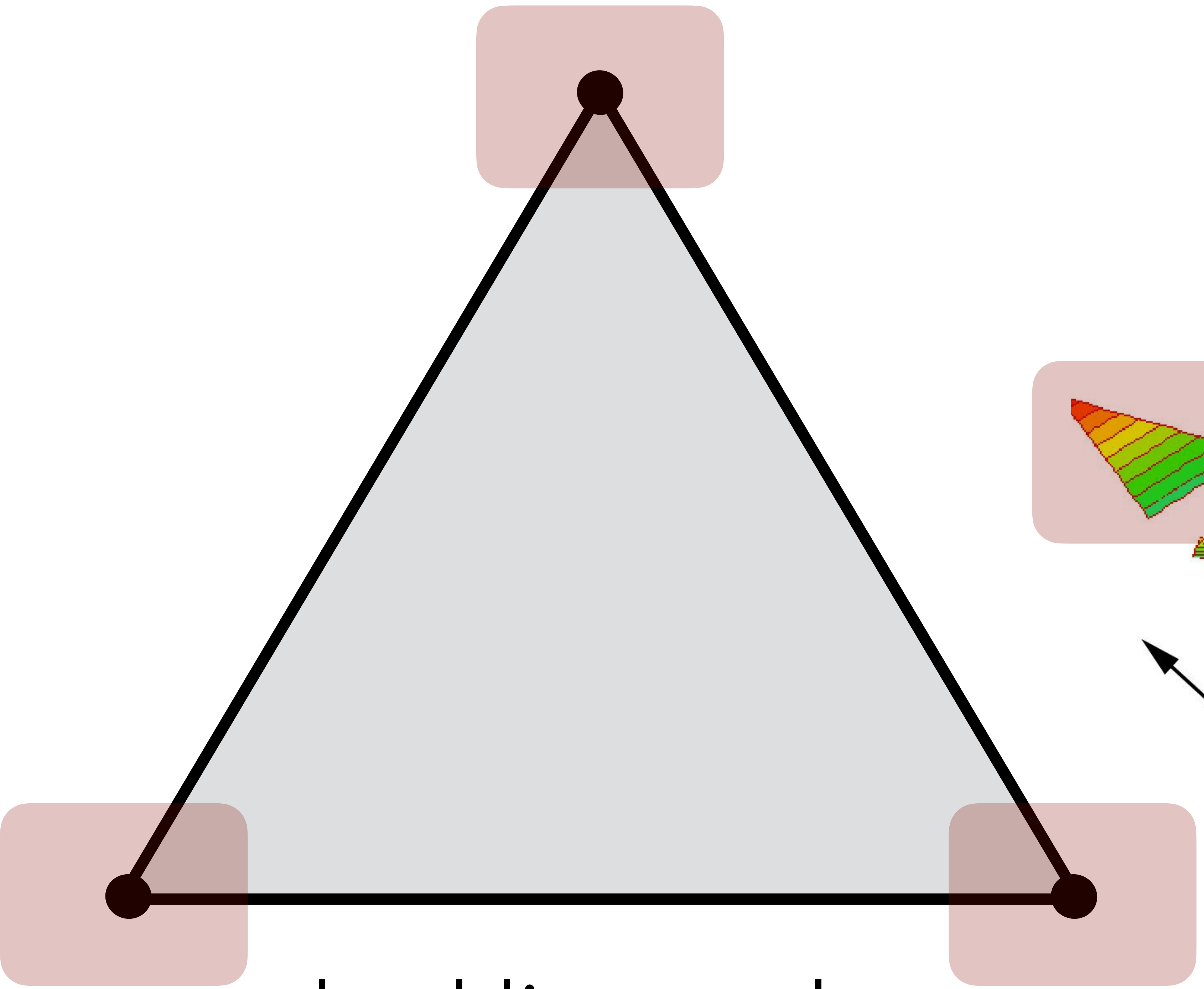
Performance on modern hardware

- High-order methods show great potential for exascale platforms.
- Examining how performant kernels can be designed for **unstructured** elements.

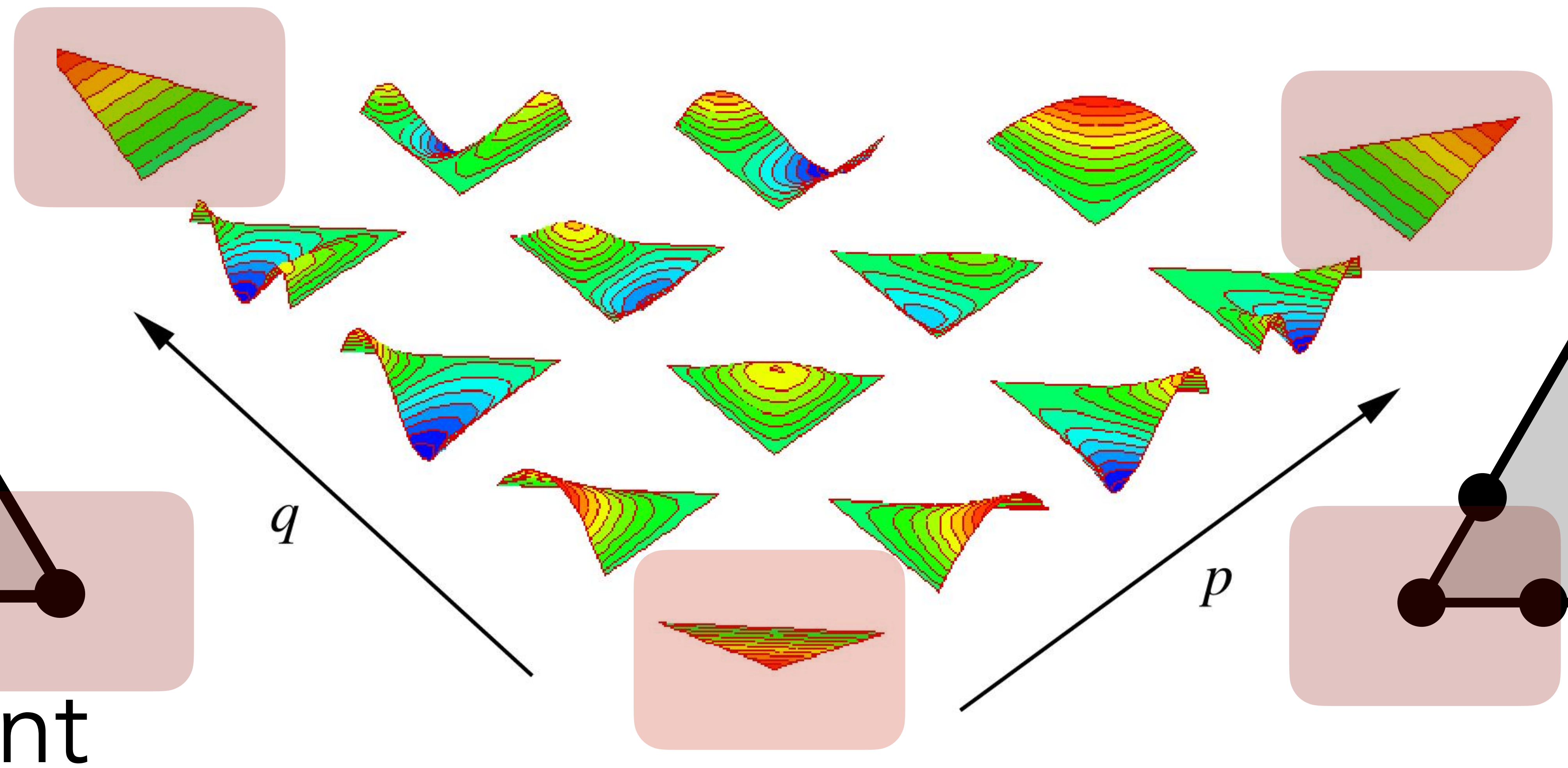
Community engagement with NEPTUNE

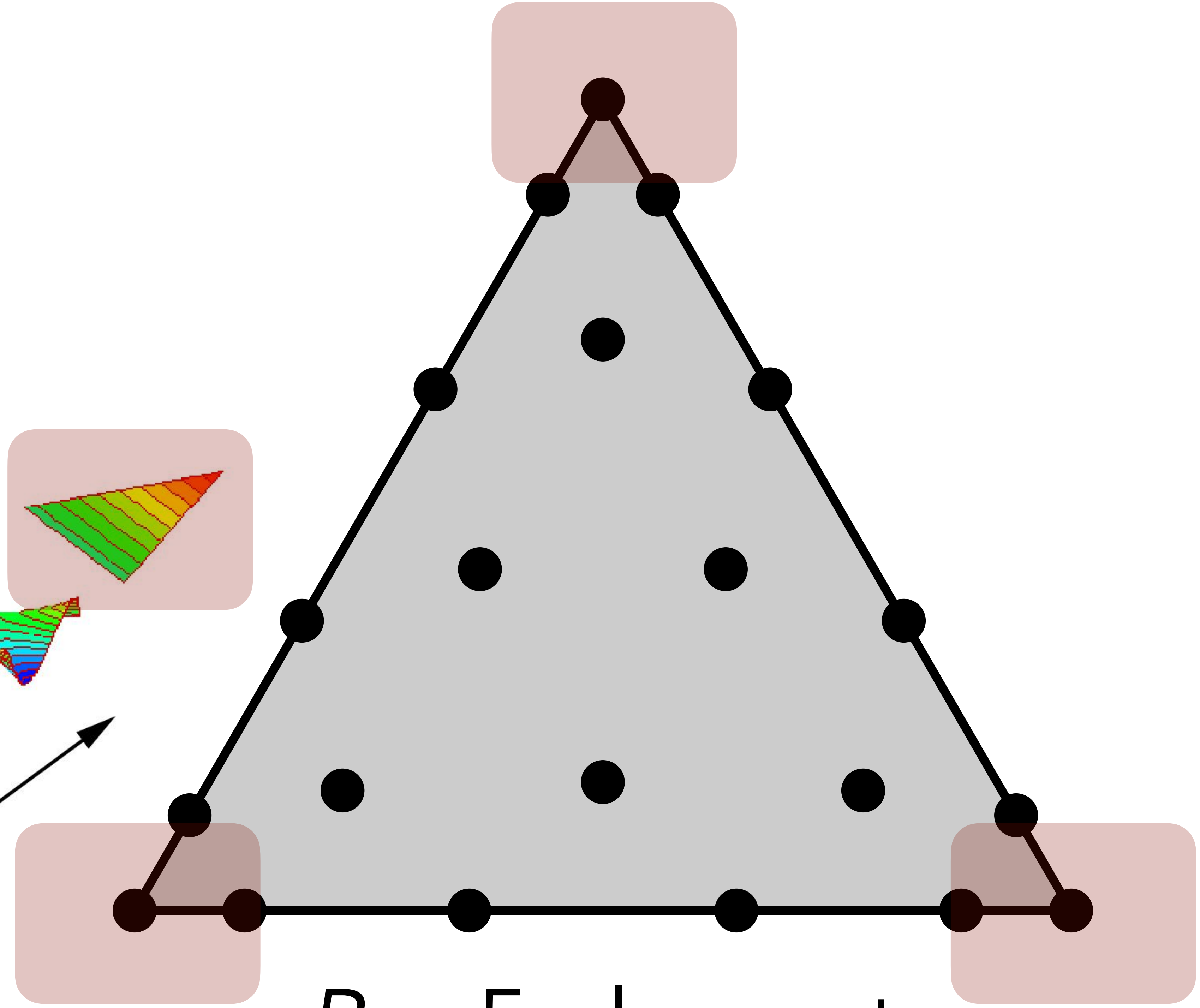
- NEPTUNE is composed of several partners, each developing proxyapps
- Developing coupling approaches, modern CI and coordination

Background: High-order methods



standard linear element
3 degrees of freedom
one per vertex (or cell)

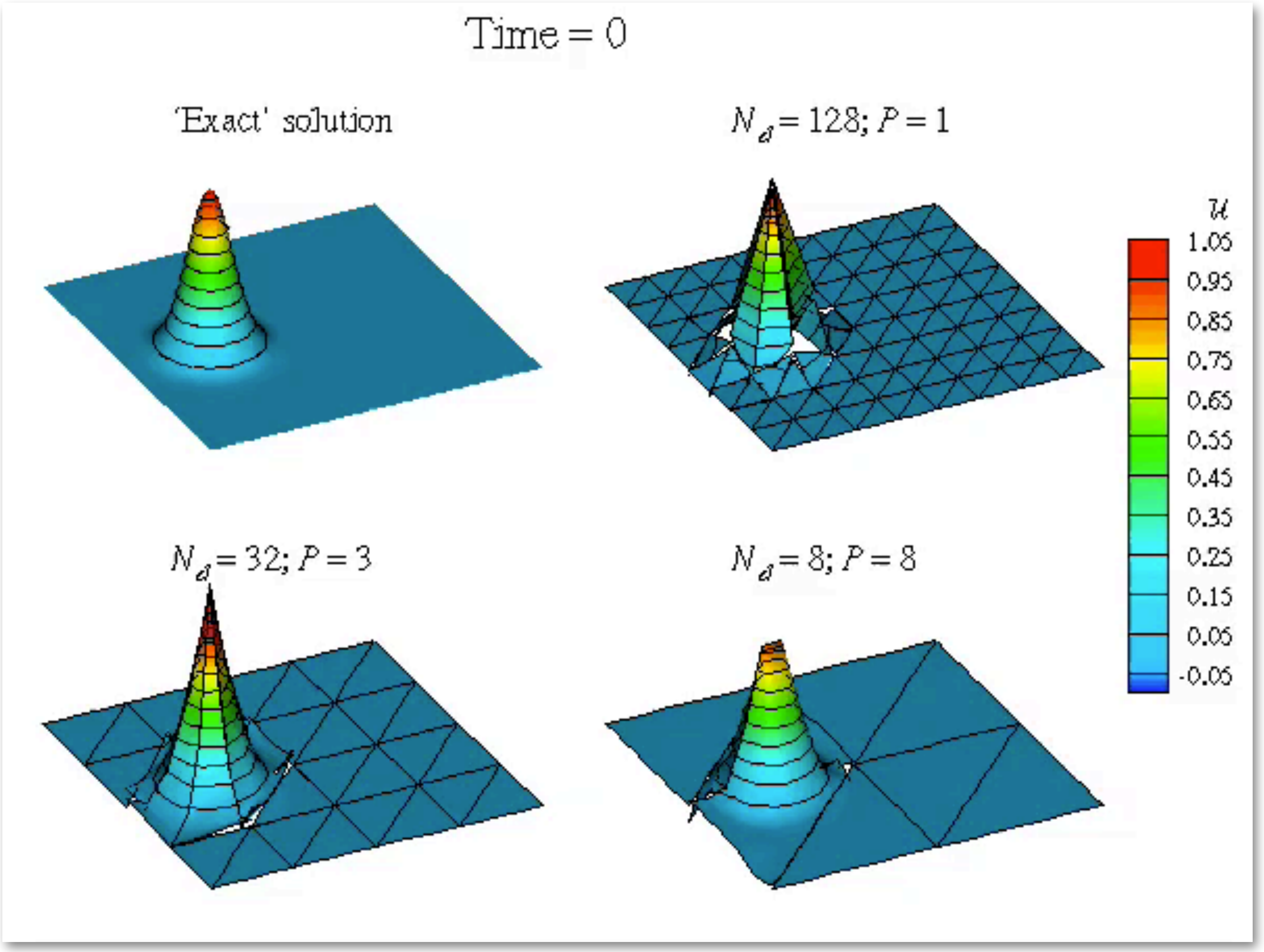


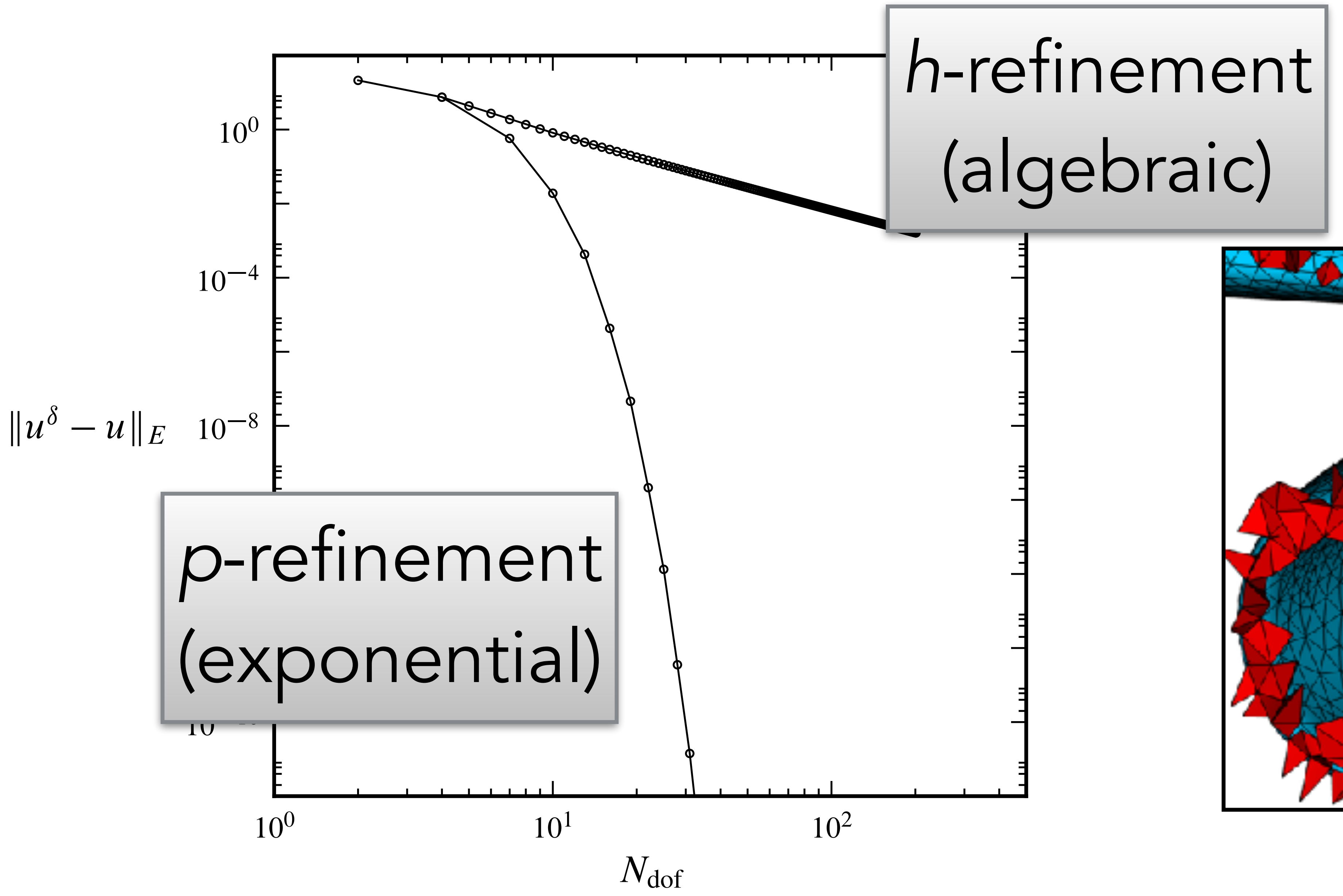


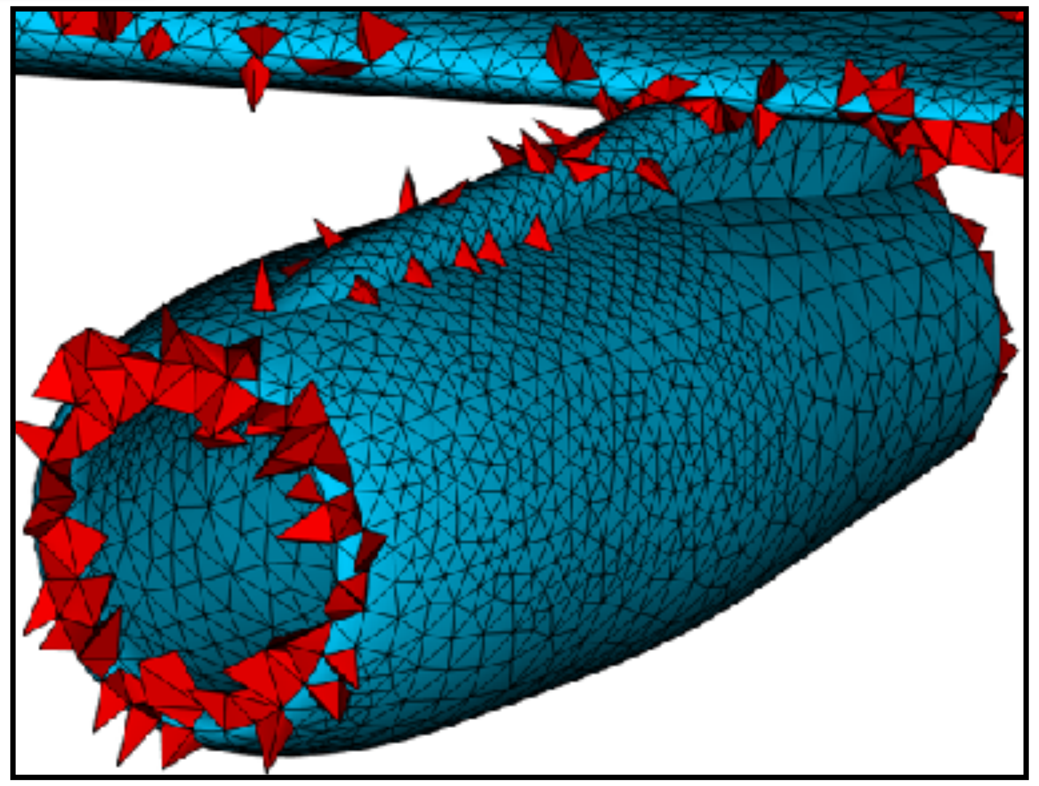
$P = 5$ element
21 degrees of freedom
vertex, edge and interior ‘modes’

Why use a high-order method?

- ✓ error decays *exponentially* (smooth solutions);
- ✓ favorable diffusion & dispersion characteristics;
- ✓ model complex domains
- ✓ computational advantage: reduced memory bandwidth, better use of hardware.

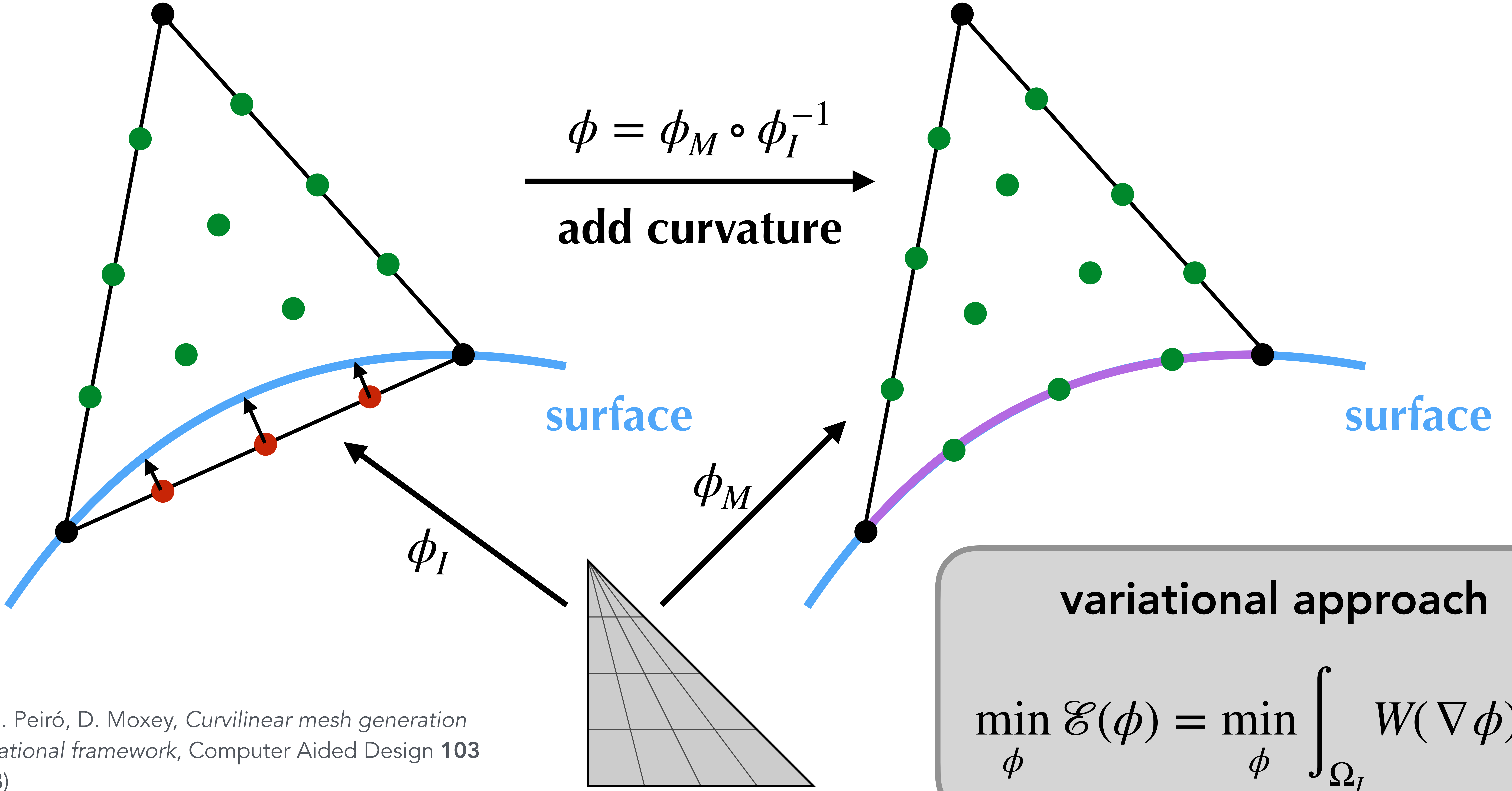






High-order mesh generation

High-order mesh generation



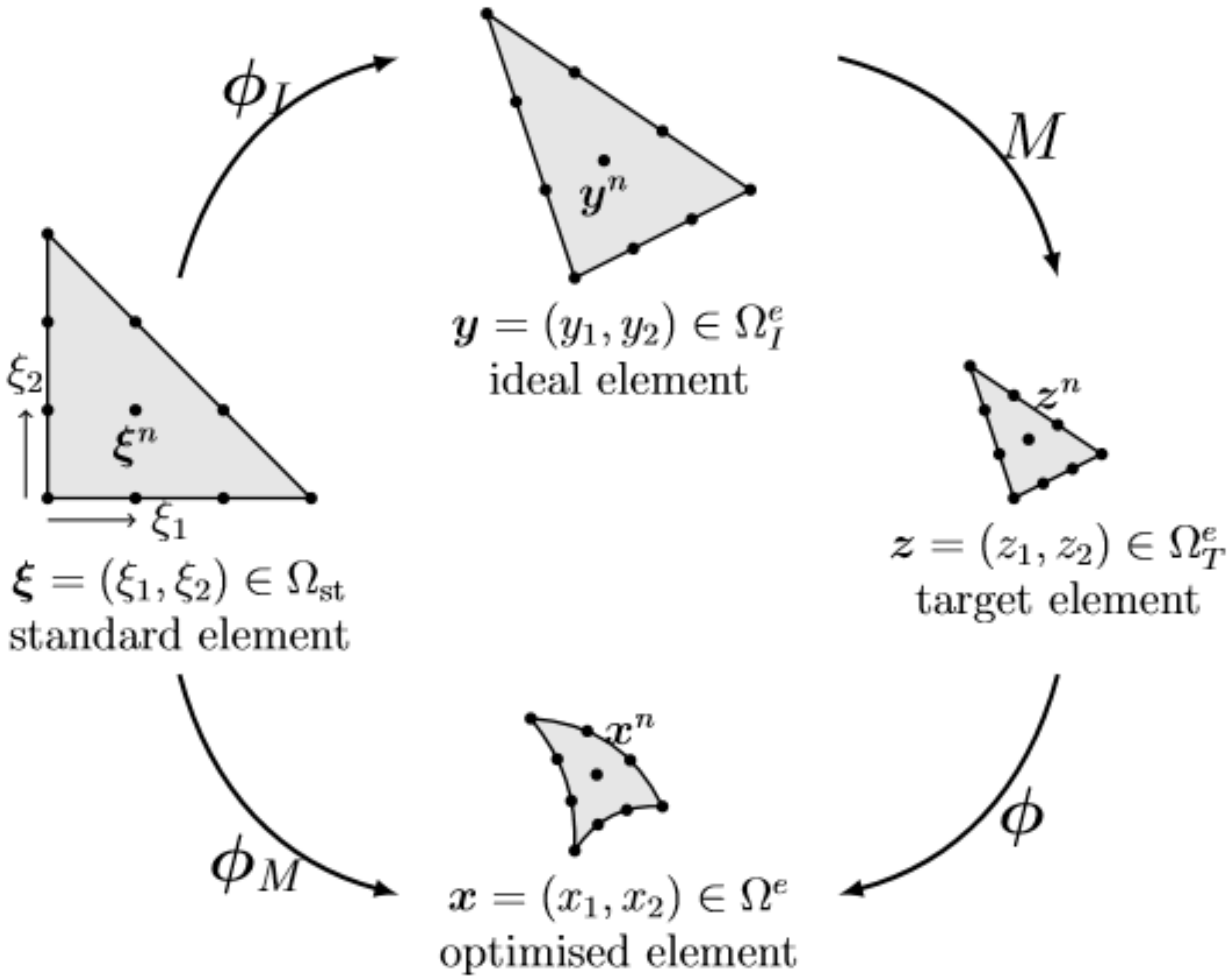
variational approach

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

M. Turner, J. Peiró, D. Moxey, *Curvilinear mesh generation using a variational framework*, Computer Aided Design **103** 73-91 (2018)

Generating anisotropic grids

- Can then modify this to incorporate an additional metric term M.
- Then the mapping becomes $\phi = \phi_M \circ \phi_I^{-1} \circ M^{-1}$.
- Adjustment of M on a per-element basis allows for the generation of anisotropic grids.
- Optimisation performed through a quasi-Newton approach, parallelised with shared memory threading.

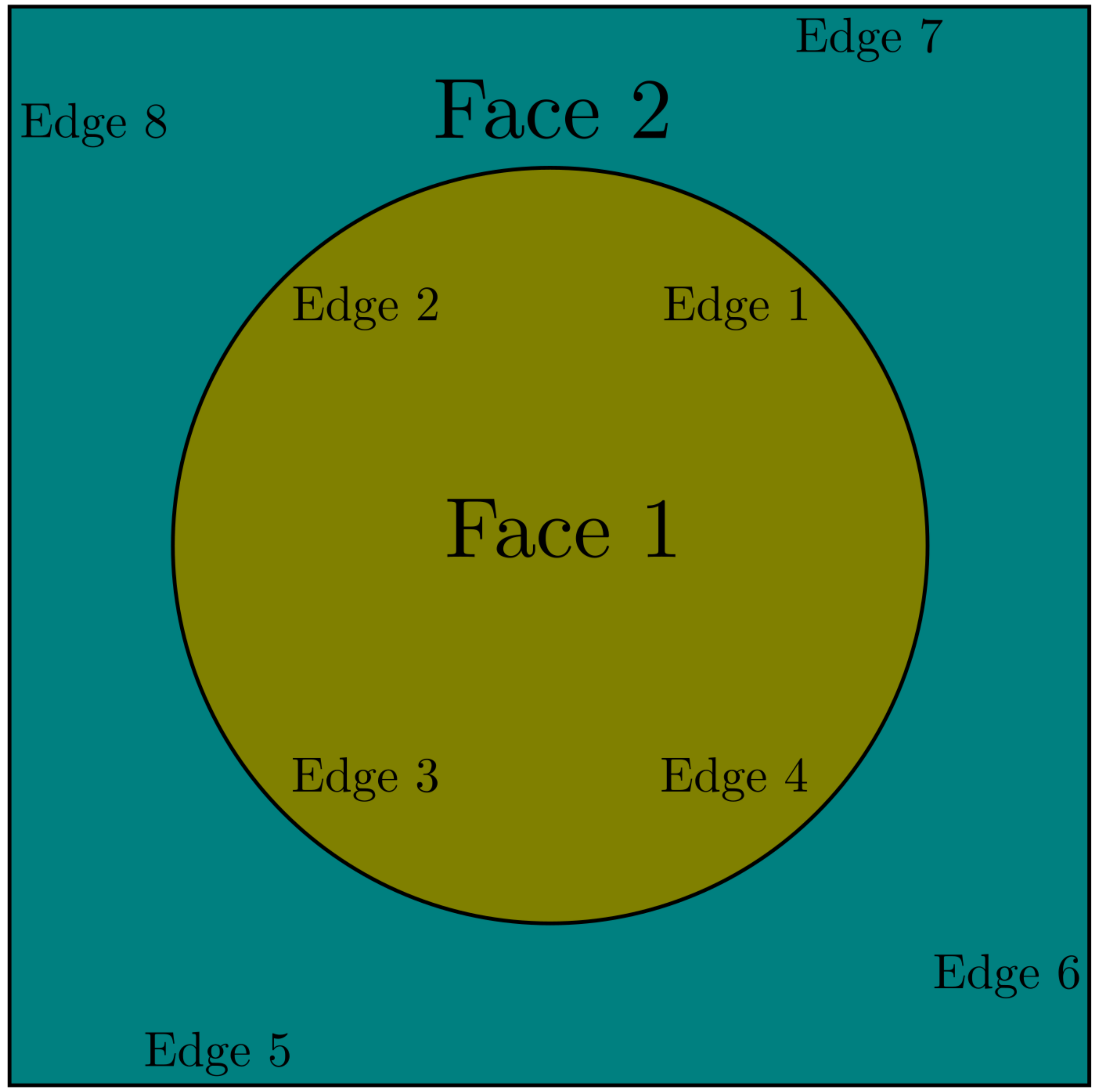


CAD-exact feature representation

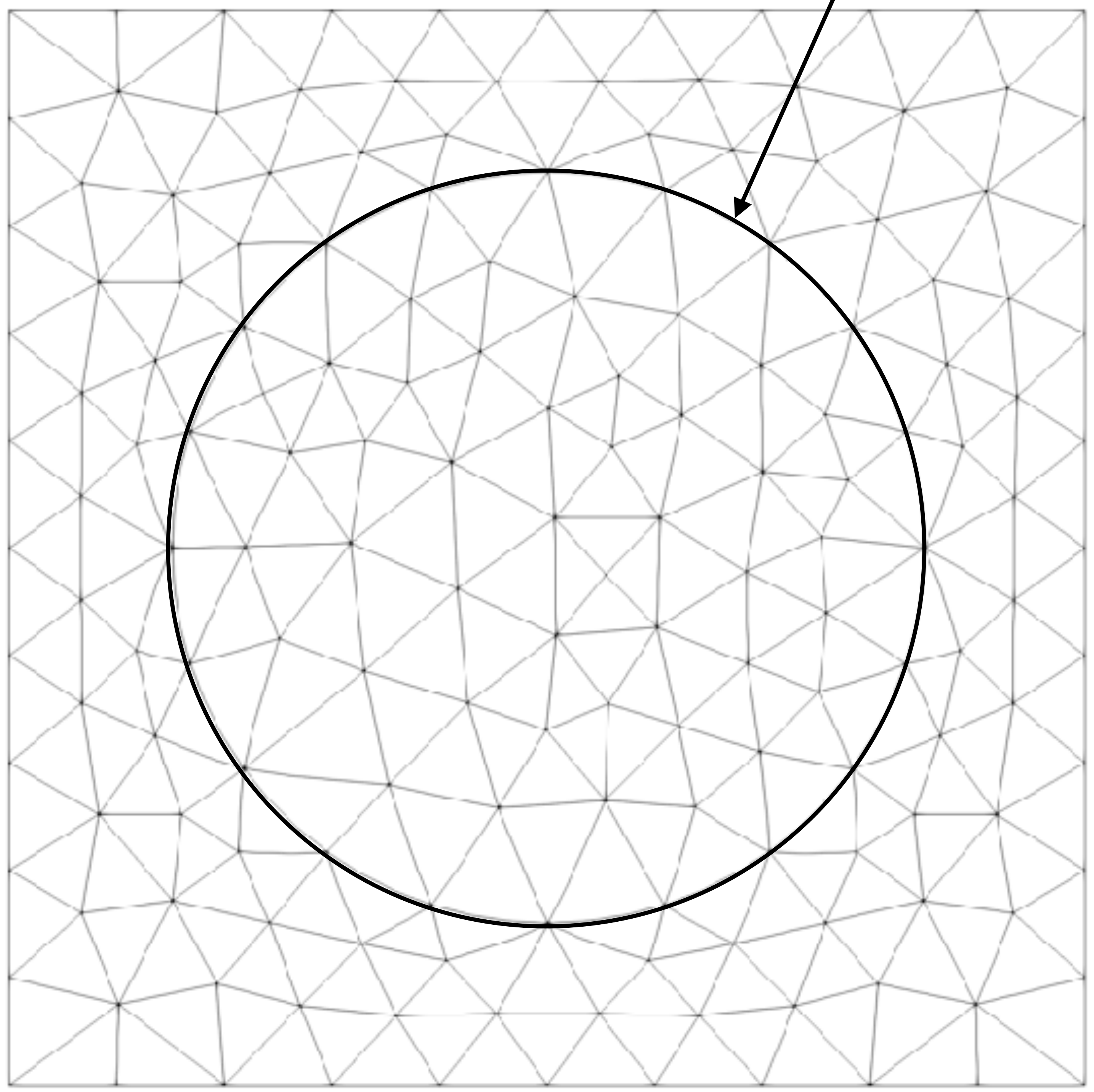
- For fusion applications, want to investigate how to resolve internal features such as the plasma separatrix.
- Also beneficial to generate anisotropic elements within these regions to align with the corresponding physical model.
- Within the mesh generation process, we have developed the use of internal CAD curves which allow for exact representation of these features.
- Furthermore, the optimisation process can be made CAD-aware, so that during optimisation, nodes 'slide' along CAD curves & surfaces.
- Implementation of this within the high-order mesh generation software *NekMesh*, which is part of the Nektar++ spectral/*hp* element framework.

Example: circle in square

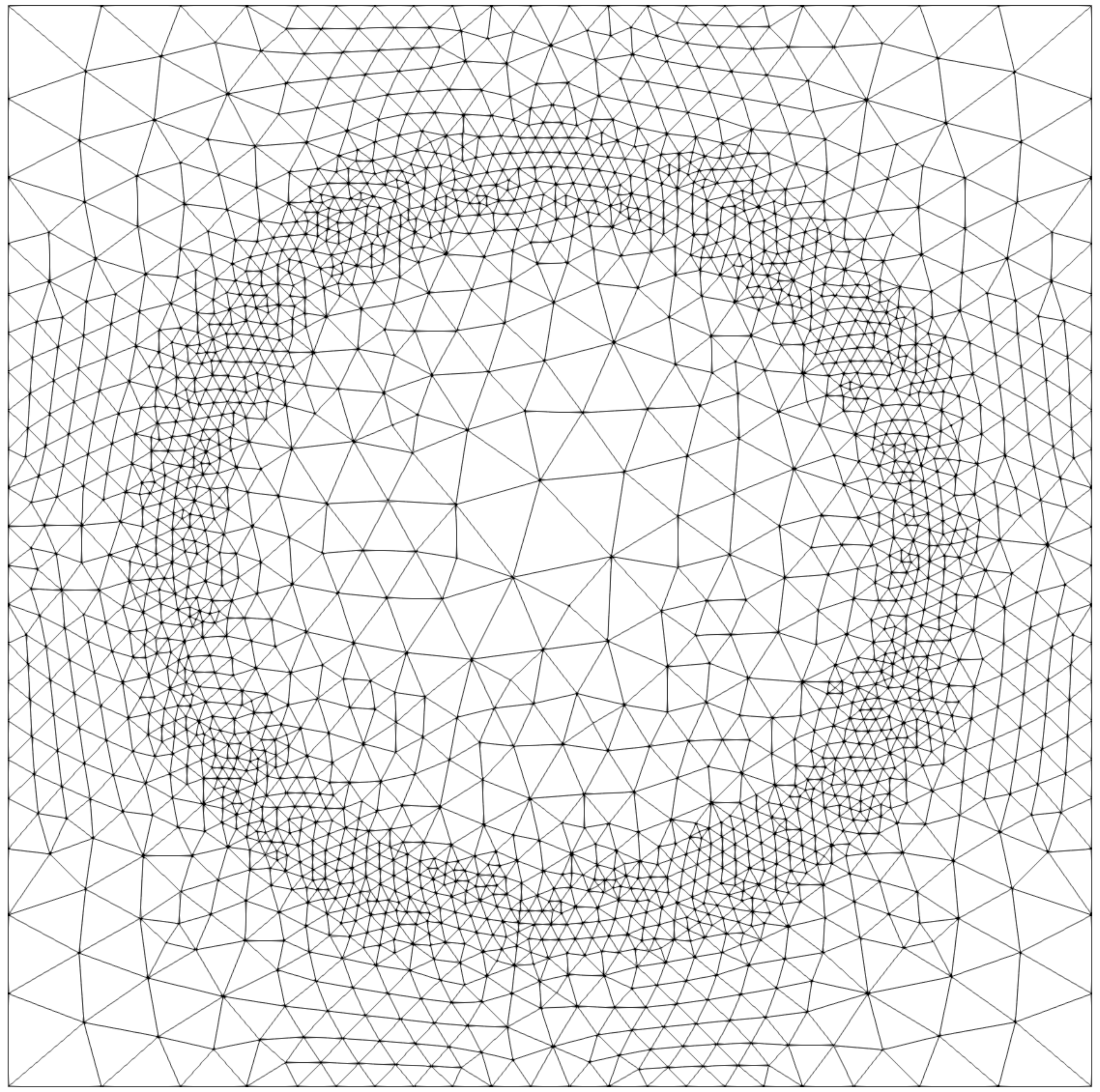
Mesh edges align with CAD curve regions



CAD boundary representation

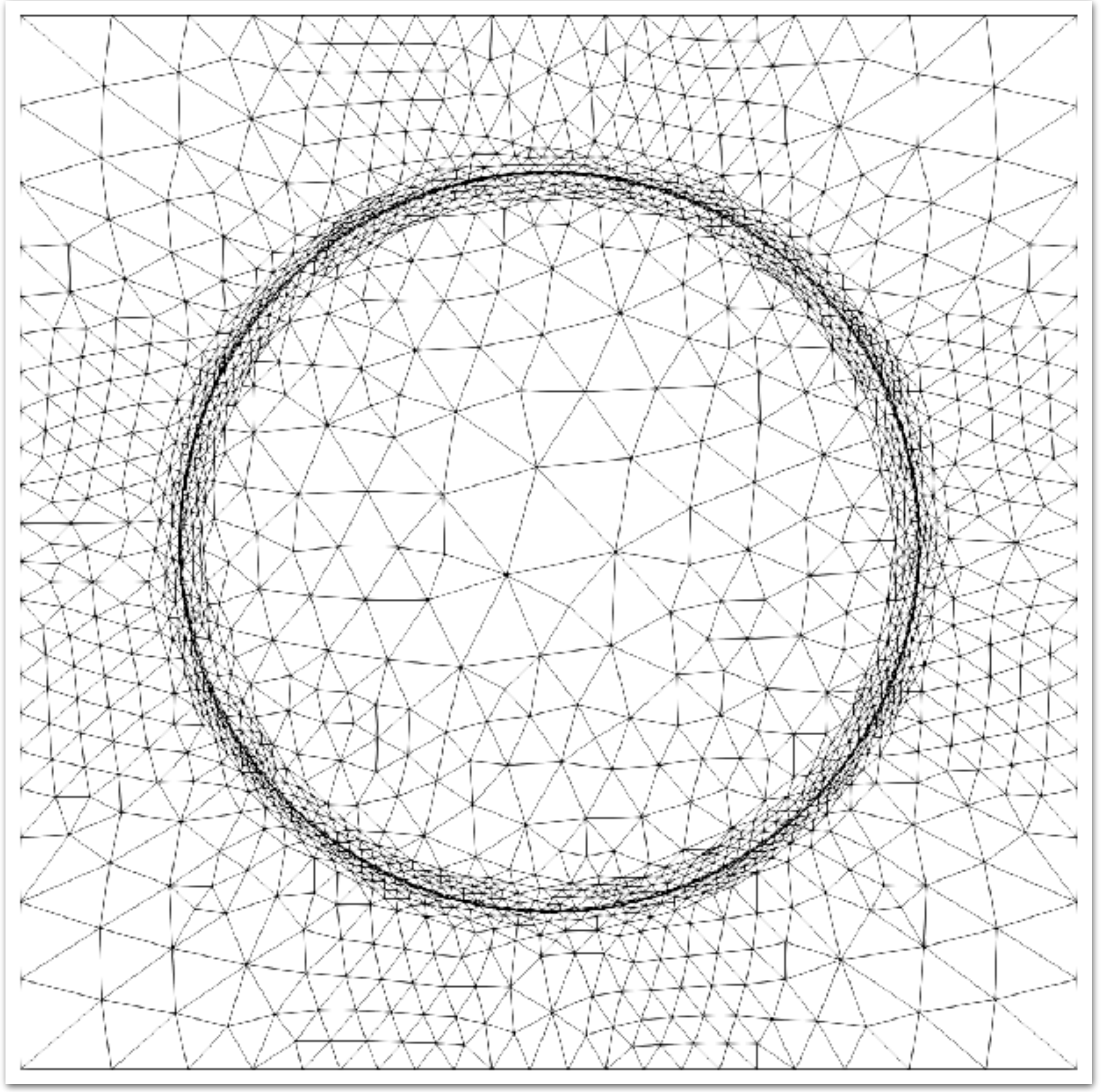


Coarse high-order grid

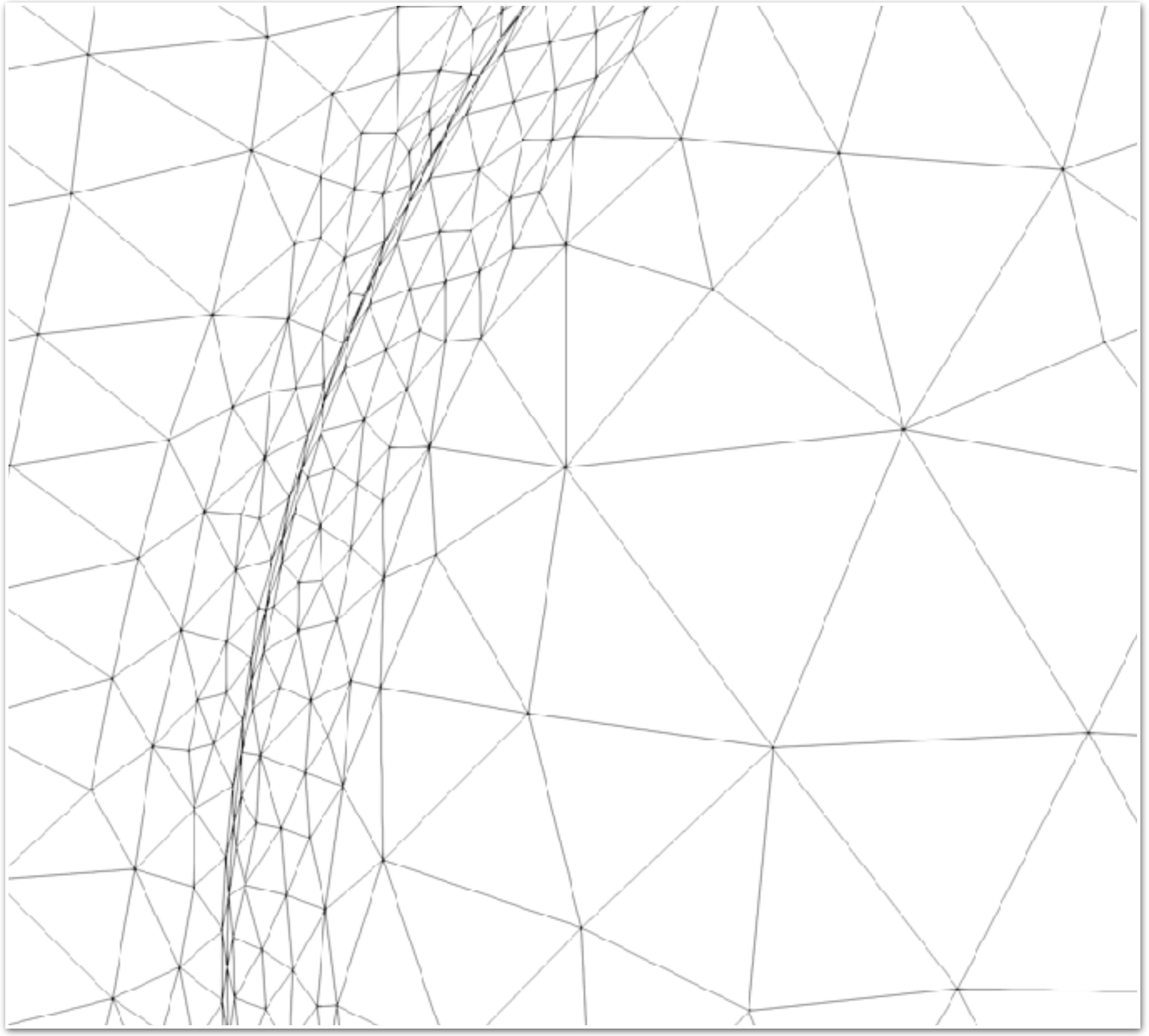


Isotropically-refined high-order grid

Example: circle in square



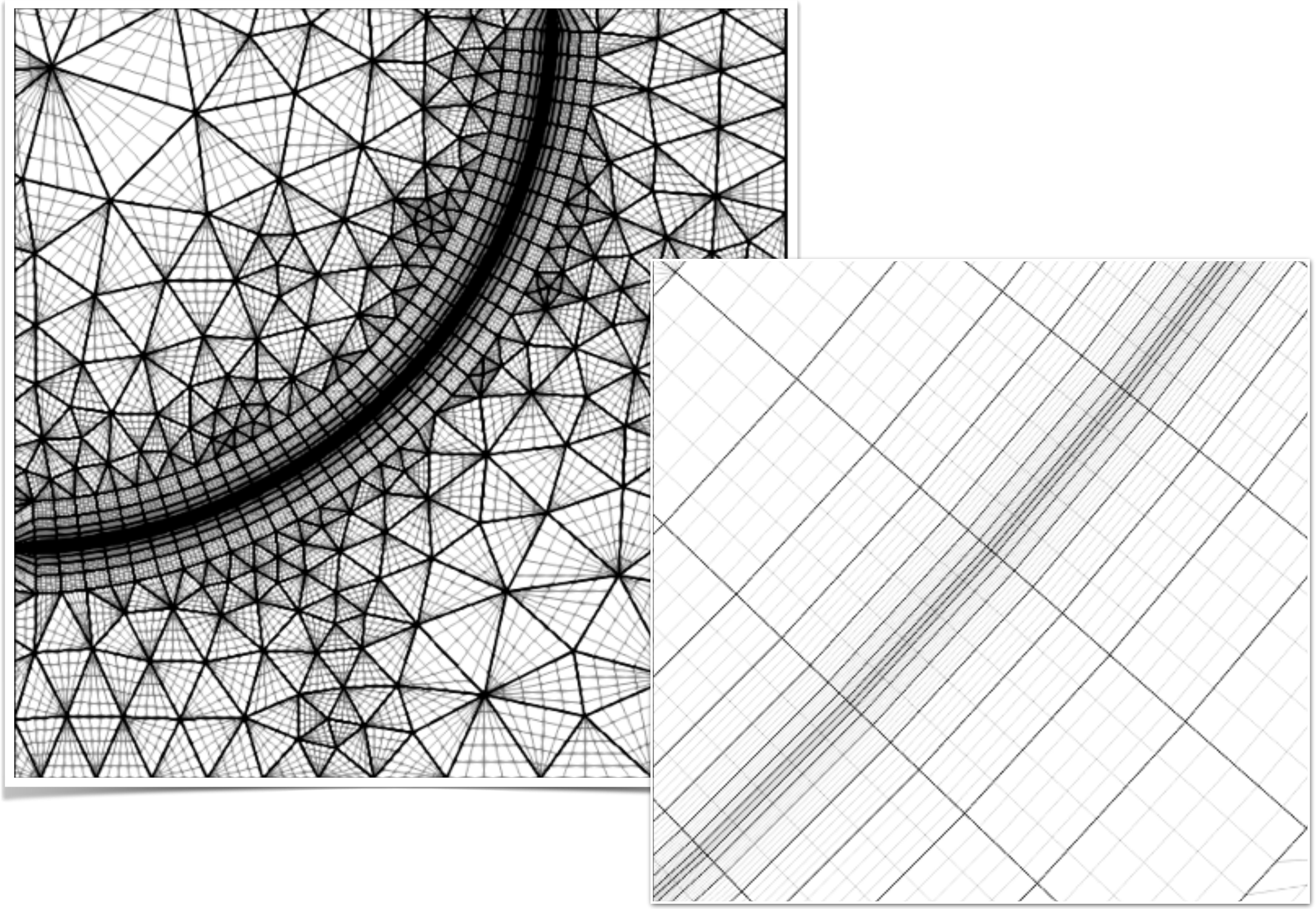
Applying anisotropic refinement



Close-up view of boundary of circle

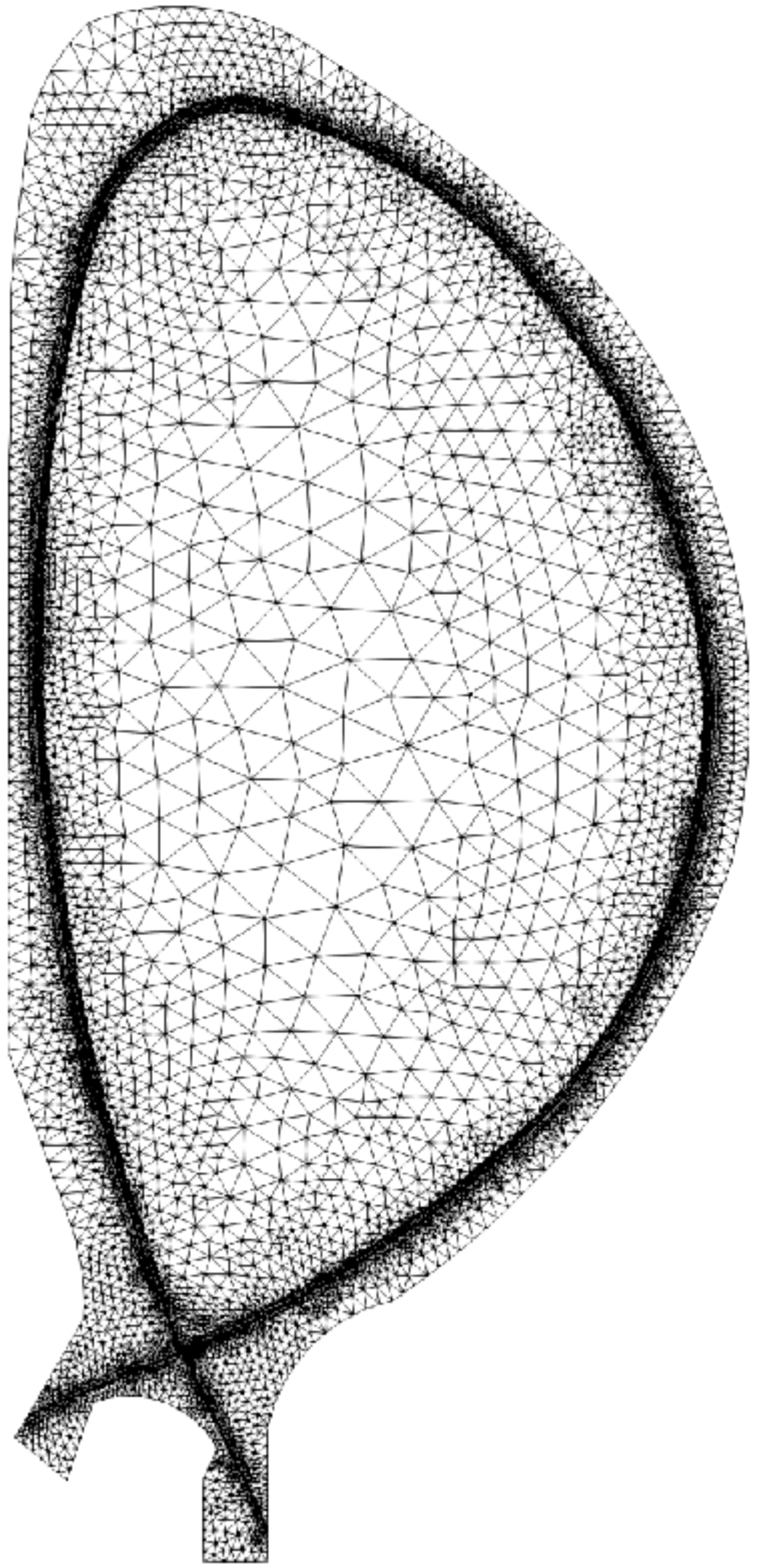
Interior layer refinements

- Have also begin process of generating refinement in a more structured manner.
- This uses boundary layer refinement strategies we typically adopt for external flows but on the internal CAD curve.
- Isoparametric approach allows arbitrary refinement as desired.



Example: sample plasma core separatrix

- Same process applied to a schematic tokamak cross-section.
- Fully valid mesh (no inverted elements) at order $P = 5$.
- Future directions:
  - extension to 3D;
  - improving performance via different optimisation strategies;
  - distributed-type parallelism, GPU offloading.



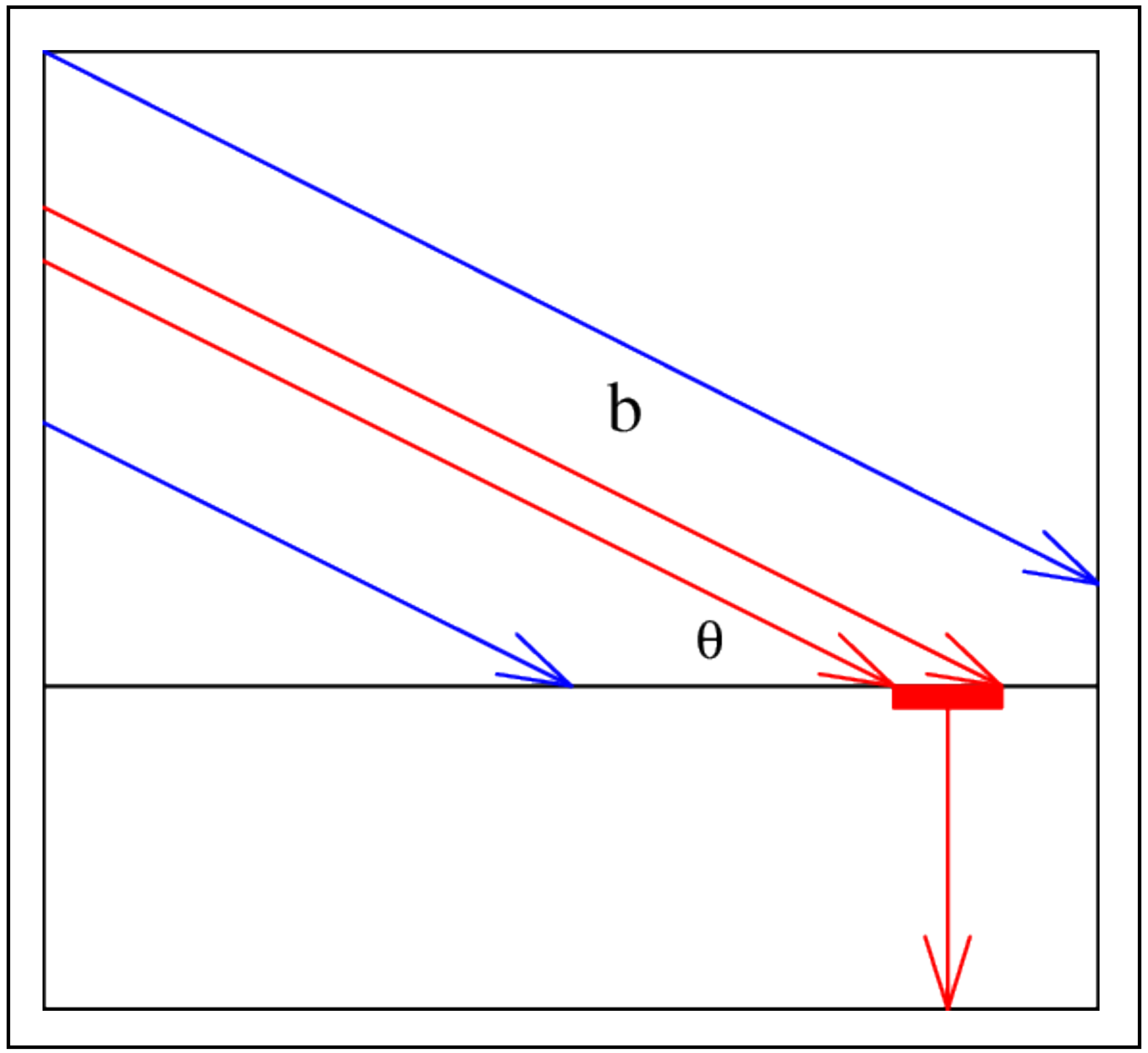
Performance

Test cases within NEPTUNE

- Focuses around setup of a solver for a highly anisotropic heat transport case based on high-order elements.

$$\frac{3}{2}N\frac{\partial T}{\partial t} = \nabla \cdot \left(\kappa_{\|}\mathbf{b}[\mathbf{b}\cdot\nabla T] + \kappa_{\perp}(\nabla T - [\mathbf{b}\cdot\nabla T])\right)$$

- Two aspects being considered:
  - **Physics:** for this model problem, how do high-order elements perform for high levels of anisotropy?
  - **Software:** how do we implement efficient realisations suitable for exascale platforms?

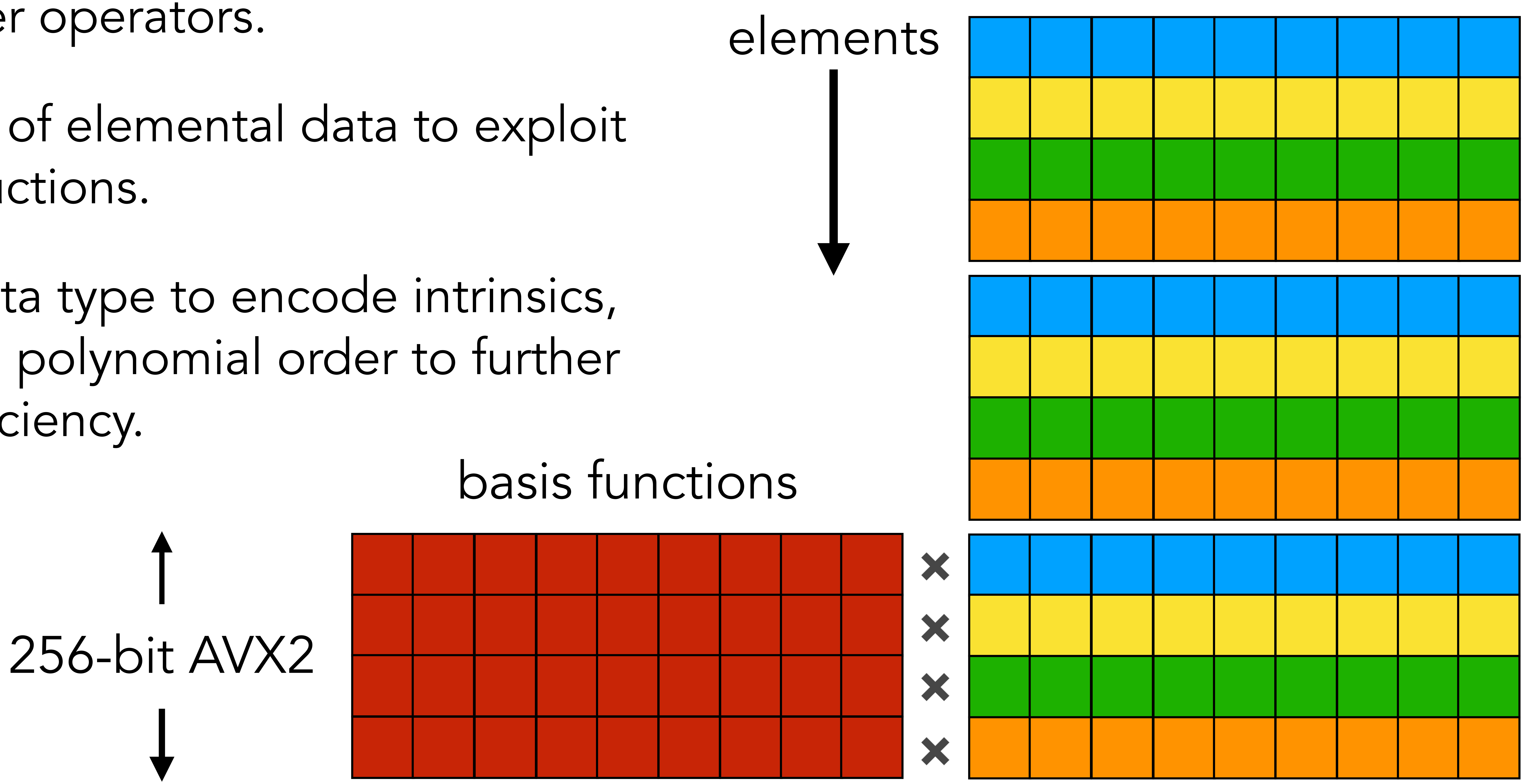


Performance & matrix-free operator evaluation

- A key ability of high-order methods is their **arithmetic intensity:** the ratio of floating-point operations (FLOPS) per byte of data transferred over memory.
- As P increases, so too does the amount of FLOPS needed for evaluation.
    - This sounds bad, but we get a corresponding increase in accuracy!
- Significantly reduce the impact of memory bandwidth limits found on modern hardware architectures & exascale platforms.
- Solution of the anisotropic heat transport involves evaluation of operators $\mathscr{L}(u) = \nabla[D \cdot \nabla u]$, where D is a spatially-constant $d \times d$ tensor.
- Second track of work focuses on performant-kernels for this operator, particularly focused on **unstructured elements** due to complexity of geometry found in realistic problems.

Matrix-free operations & data layout

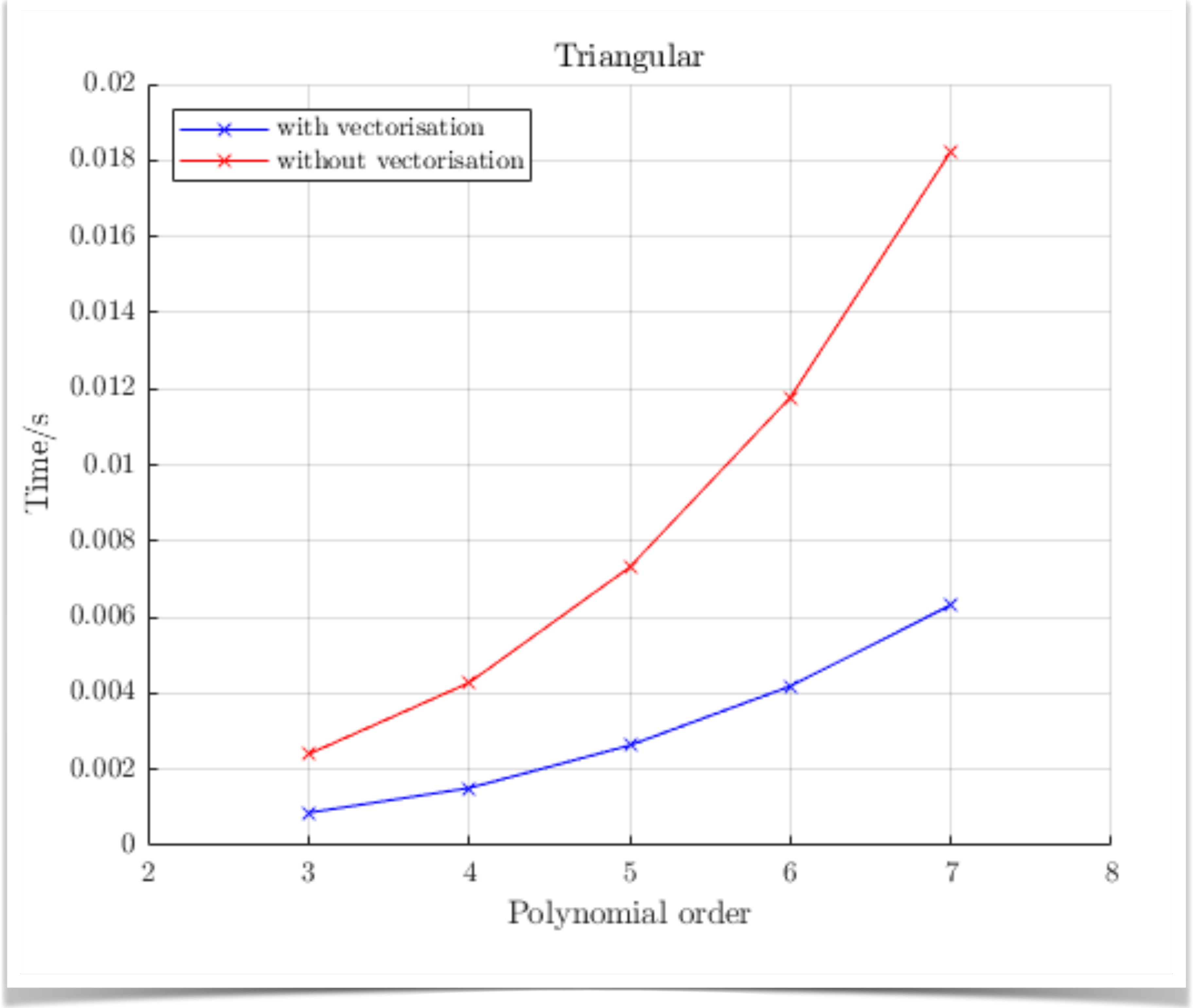
- Focused on **matrix-free** implementation of high-order operators.
- Interleaving of elemental data to exploit vector instructions.
- Use C++ data type to encode intrinsics, template on polynomial order to further improve efficiency.

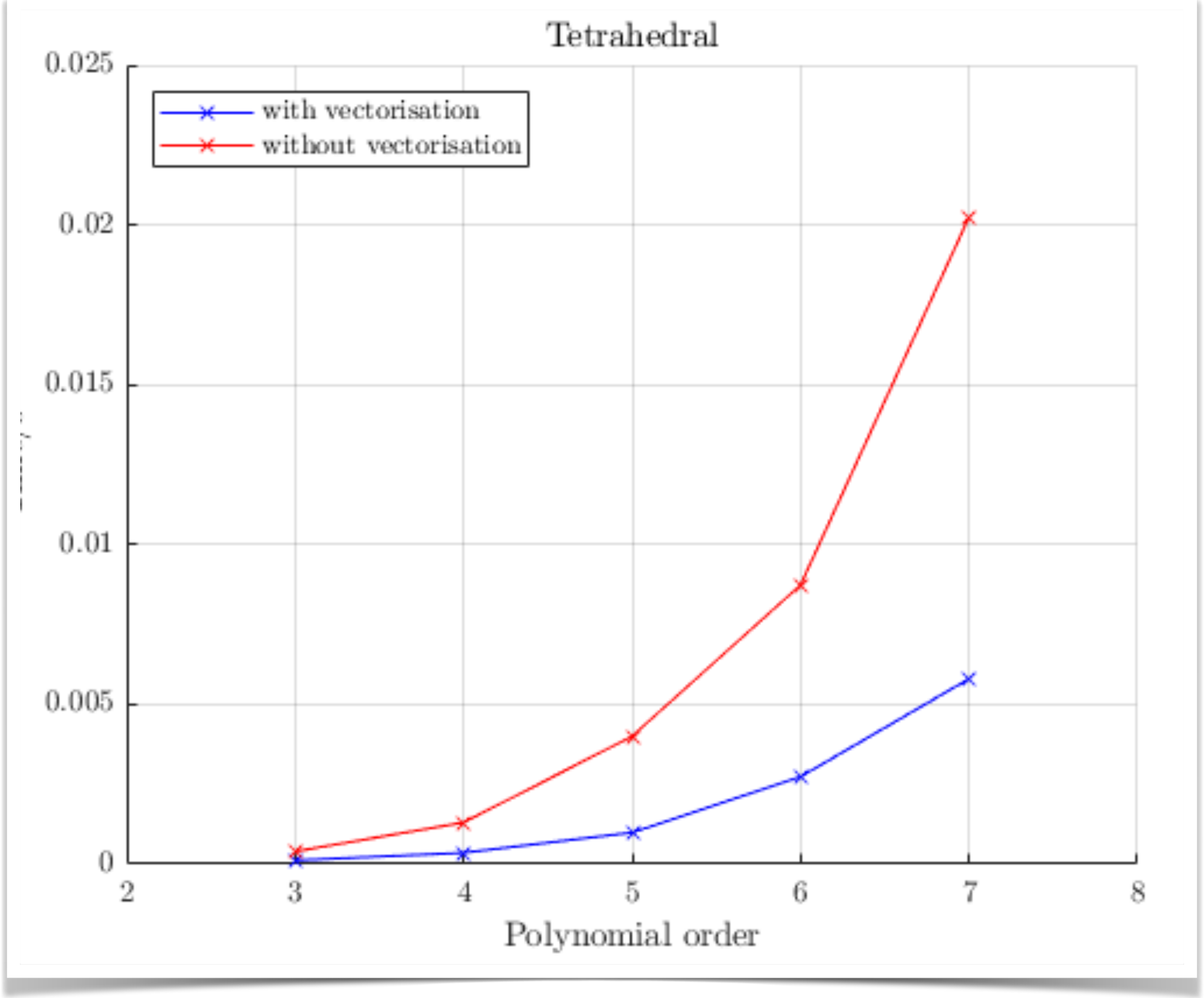


Software implementation

$$\mathscr{L}(u) = \nabla[D \cdot \nabla u]$$

d x *d* spatially-constant diffusion tensor





Good speedups observed over non-vectorised versions of this operator for both 2D & 3D elements

Summary

Nektar++

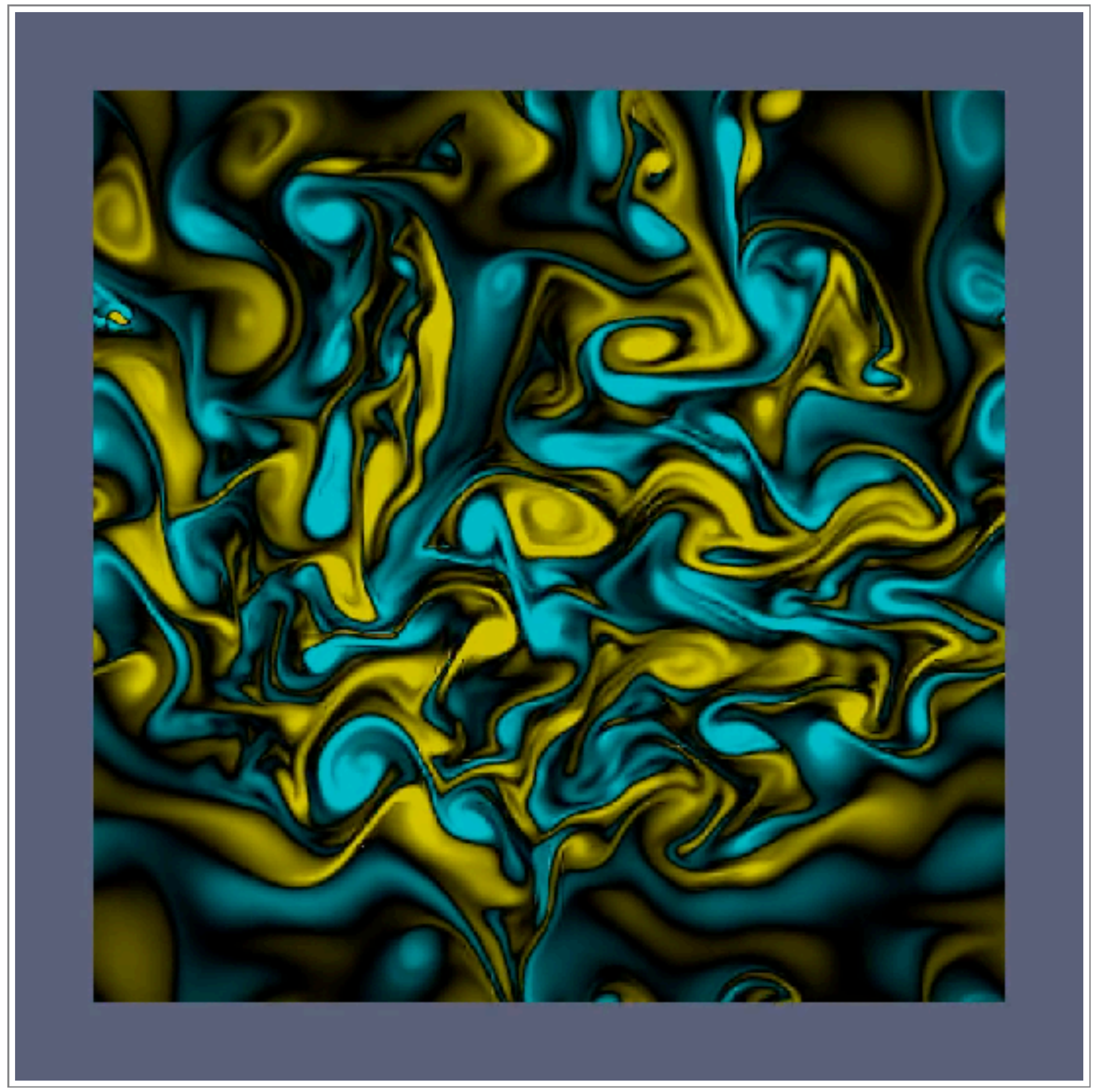
spectral/hp element framework

- Nektar++ is an **open source framework** for high-order methods.
- Although fluid dynamics is a key application area, we try to make it easier to use these methods in other areas.
- C++ API, with ambitions to bridge current and future hardware diversity (e.g. many-core processors, GPUs).
- Modern development practices with continuous integration, git, etc.

Summary & future work

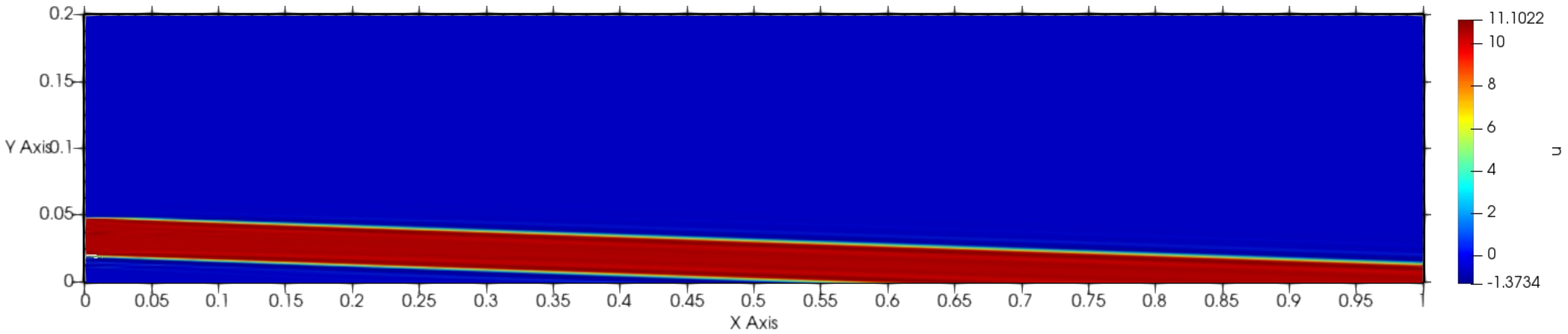
- Good progress in achieving significant developments that target features of interest to the NEPTUNE community.
- Overall summary of our project & future directions:
    - In **mesh generation**, clear requirement to develop 3D generation strategies extending some of the methods we have implemented here.
    - In **performance**, we developed x86 implementations using AVX2/AVX-512, ARM SVE and GPU versions, looking to properly integrate this within the solver framework.
    - In **community-building**, ensuring our proxyapps are well-documented and can be extended for future needs.

Development of proxyapps



Hasegawa-Wakatani drift-wave system

https://github.com/ExCALIBUR-NEPTUNE/nektar-driftwave



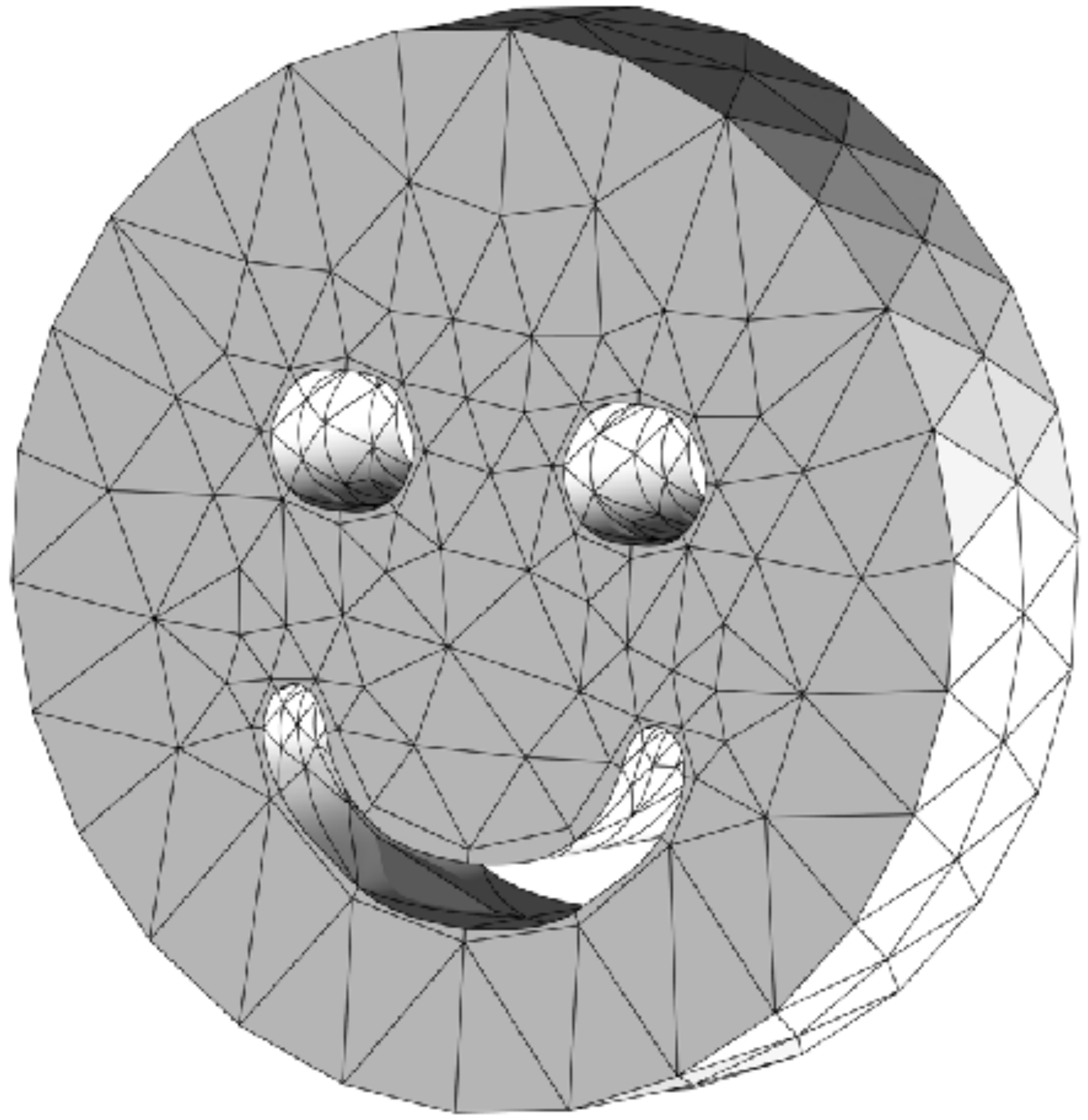
Anisotropic heat transport

https://github.com/ExCALIBUR-NEPTUNE/nektar-diffusion

SOL-solver (single species 1D solver)

https://github.com/ExCALIBUR-NEPTUNE/nektar-sol-1d

Thanks for listening! Questions?



www.nektar.info

gitlab.nektar.info

https://davidmoxey.uk/

d.moxey@kcl.ac.uk

@davidmoxey