Spectral/hp element modelling for high fidelity simulation

Spencer Sherwin

Imperial College London

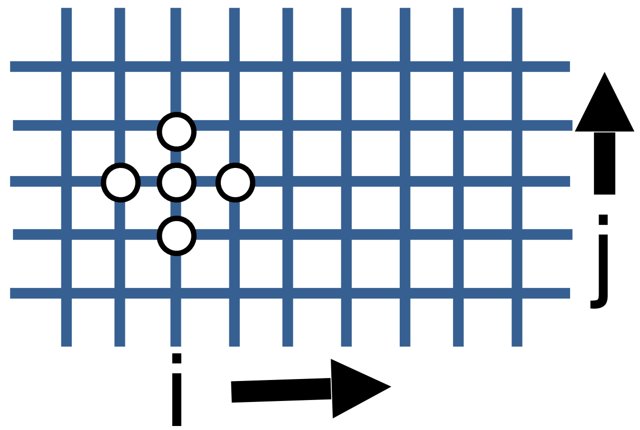
David Moxey

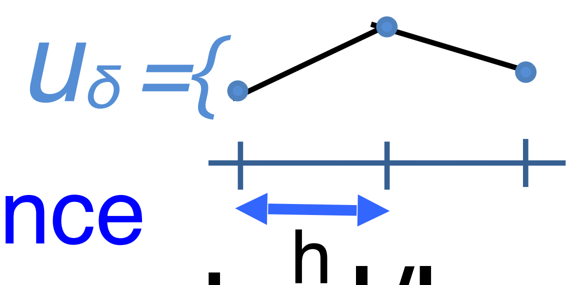
University of Exeter

Outline

- Spectral/hp element Modelling
  - Overview of Nektar++
- Challenges for modelling
  - Stabilisation and robustness
  - Performance
  - Mesh generation
- Applications
  - Hasegawa-Wakatani equations
  - Incompressible Flow Modelling

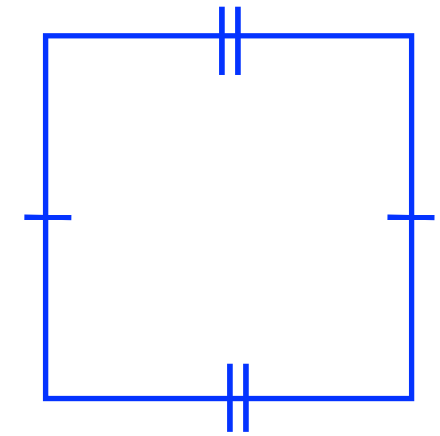
Spectral/hp methods

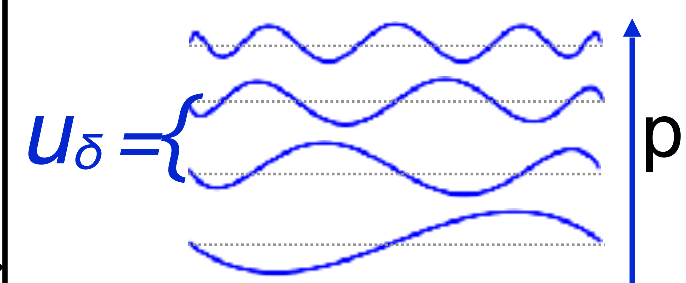


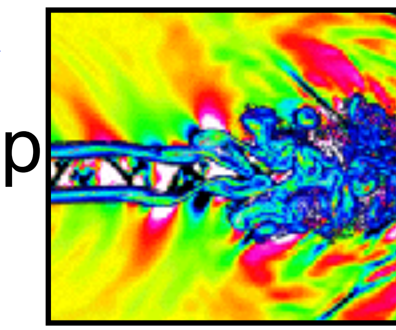


finite difference

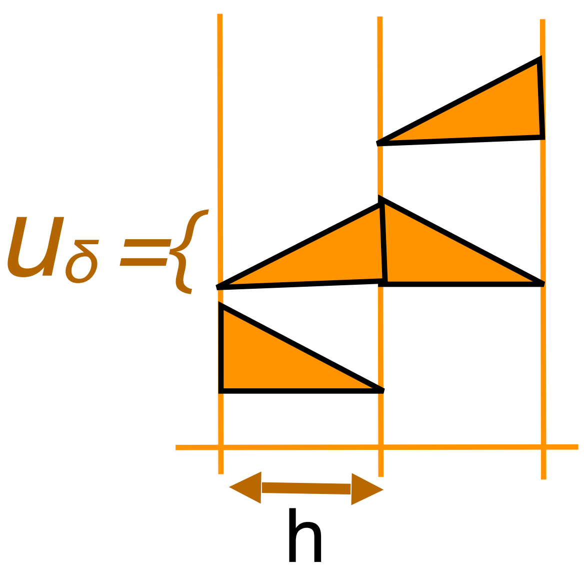
spectral methods

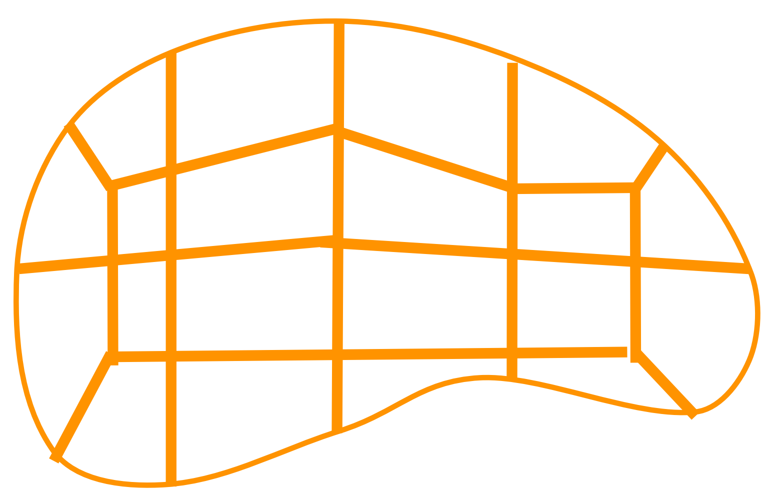


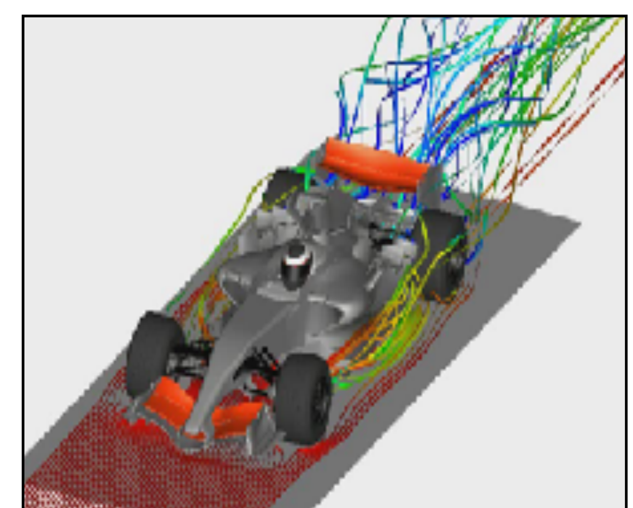




finite element/finite volume



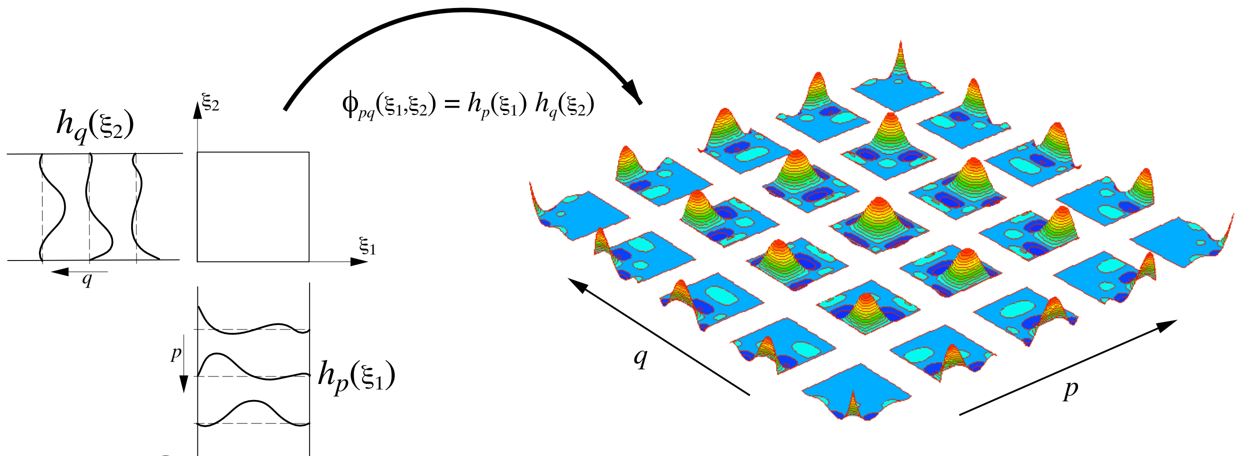




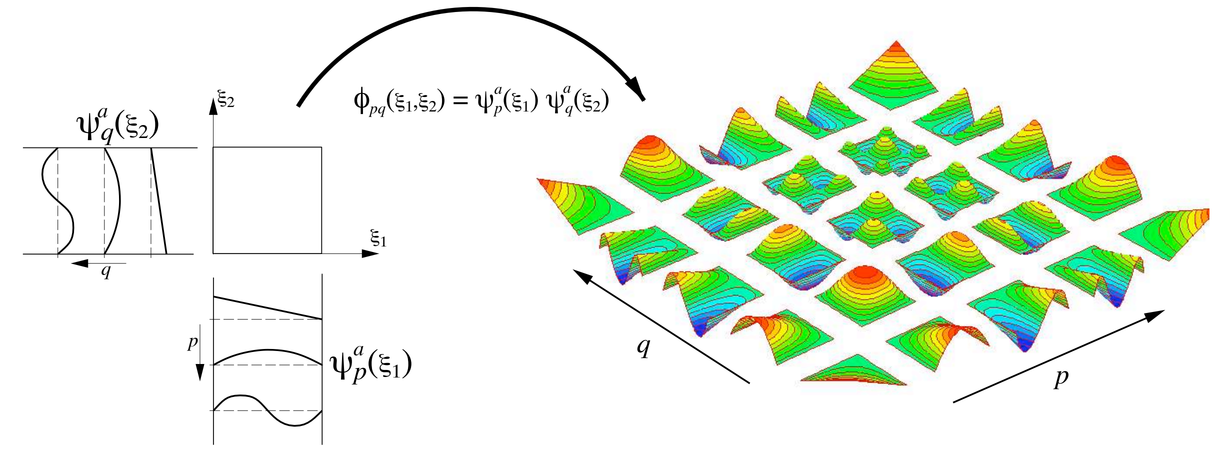
Imperial College London

Spectral/hp elements

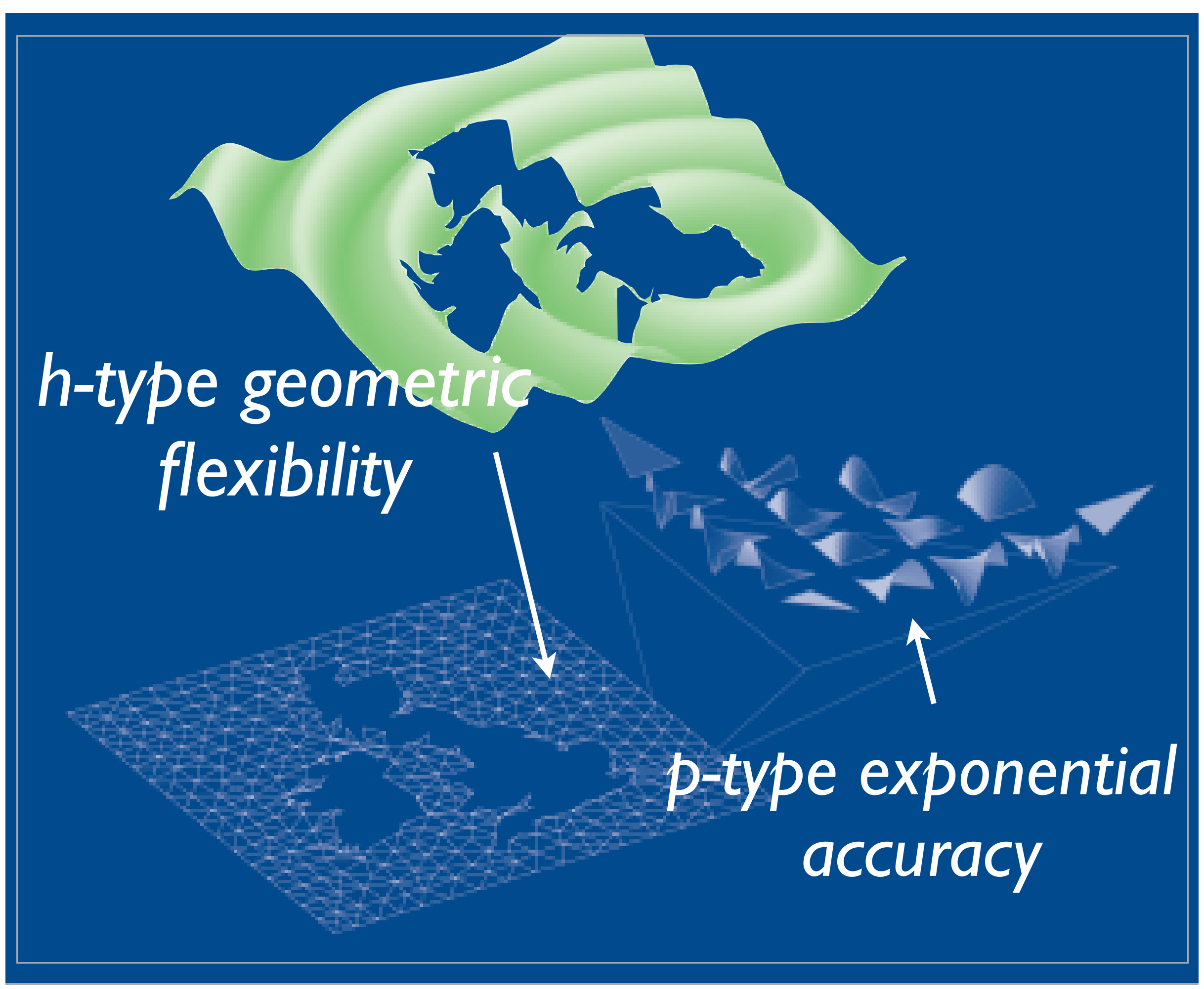
UNIVERSITY OF EXETER



Spectral element method - nodal



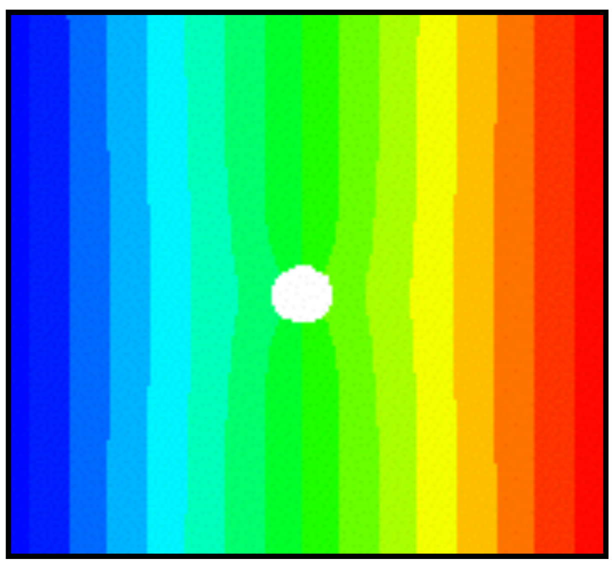
hp finite element - hierarchical



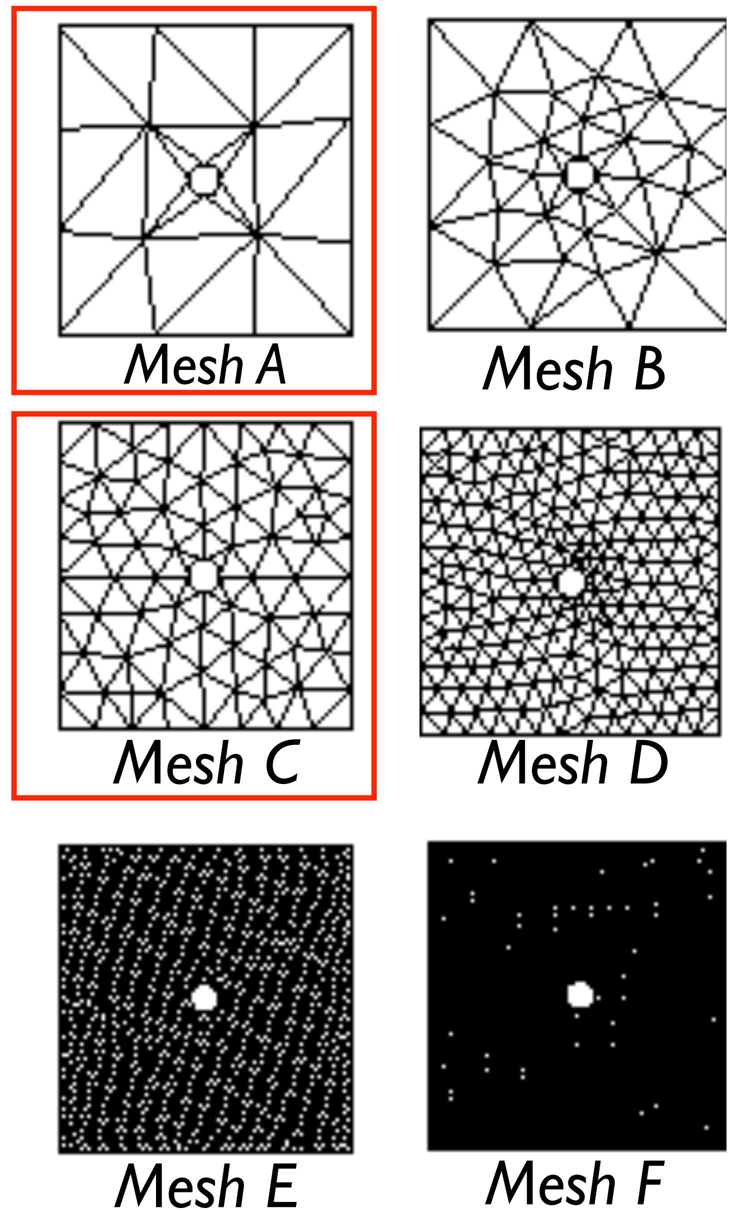
- Continuous Galerkin
- Discontinuous Galerkin
- Flux Reconstruction

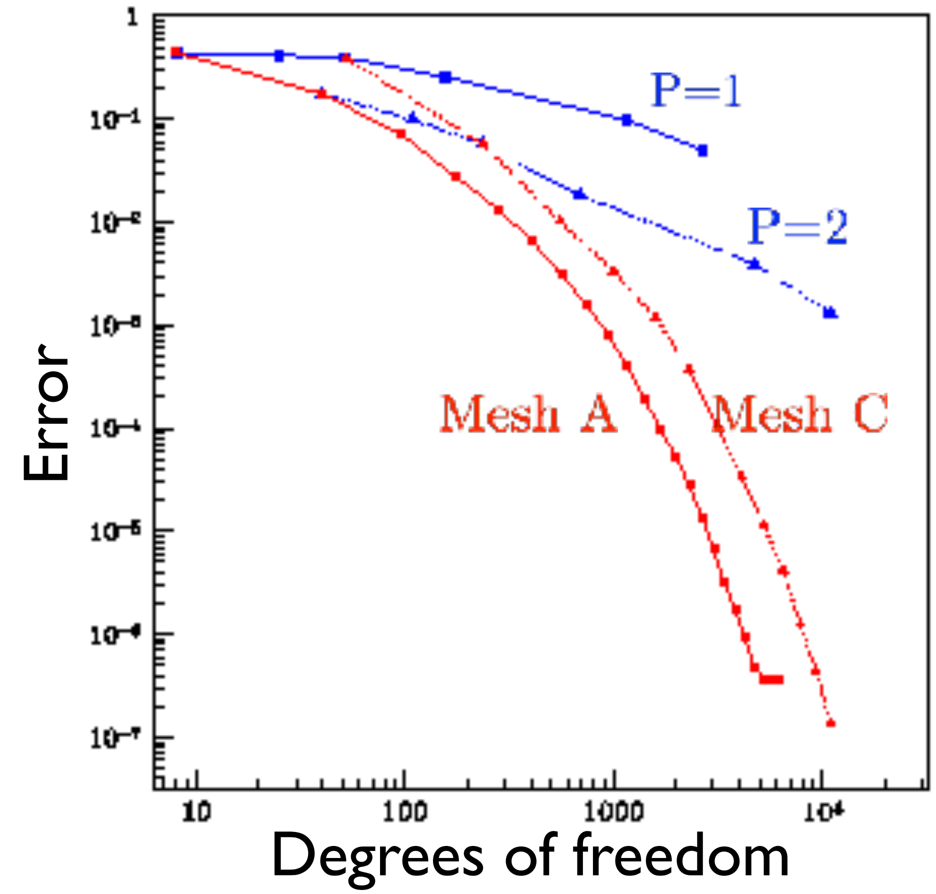
Depends on how we bolt elements together!

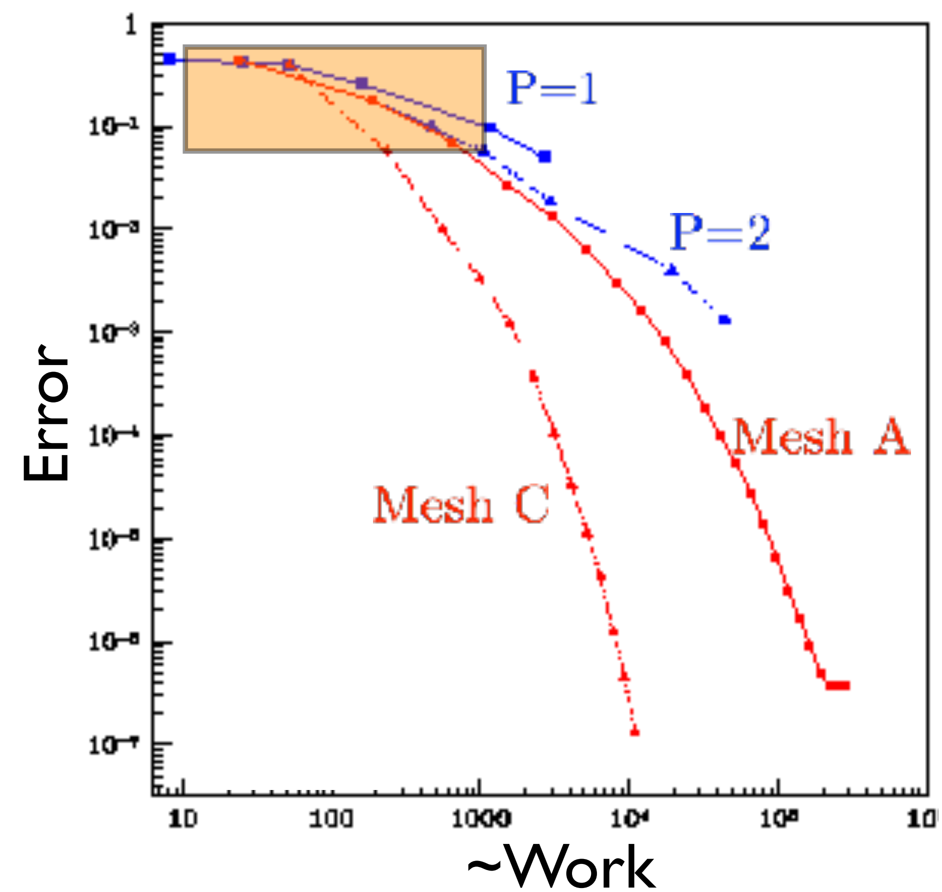
Computational cost ~ P^4 Computational error ~ h^P



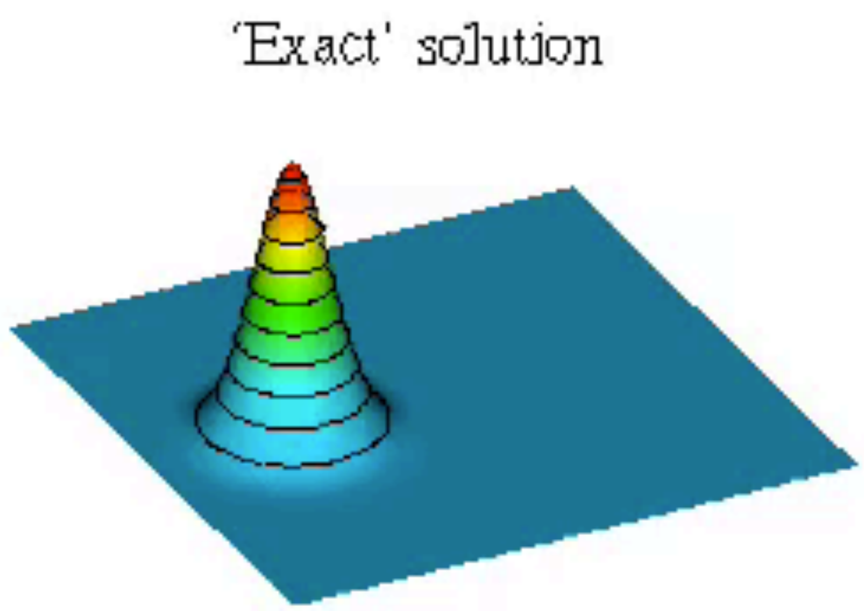
$$\phi = U[r + a^2/r]\cos\theta$$





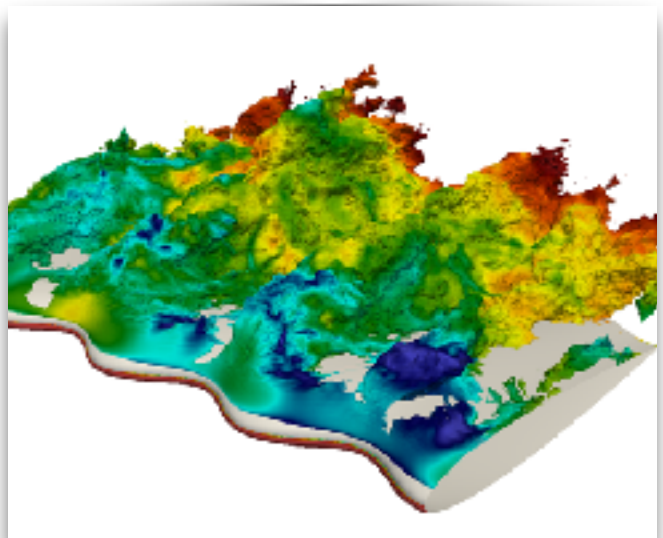


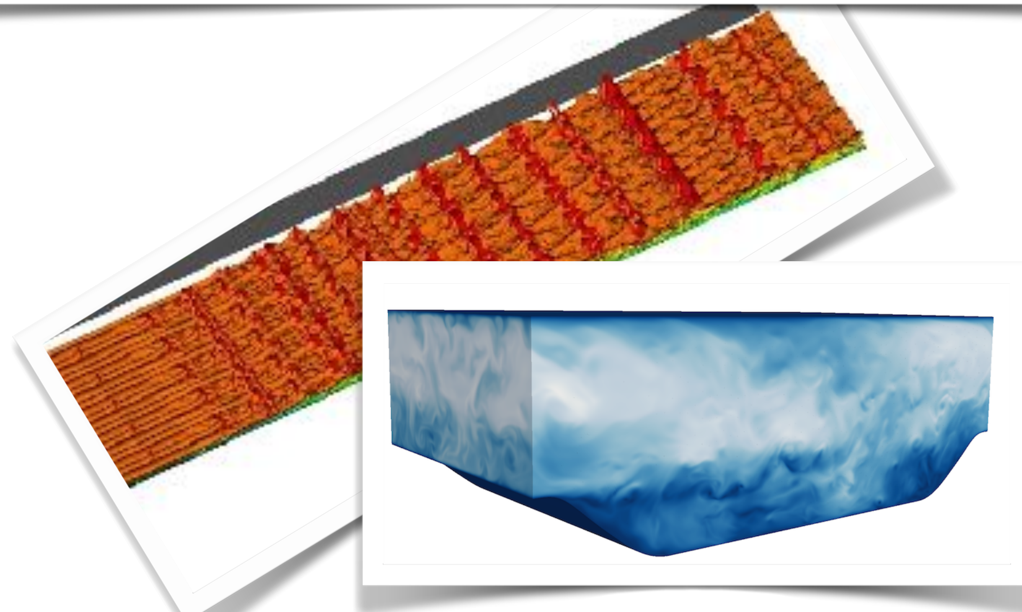
P accuracy

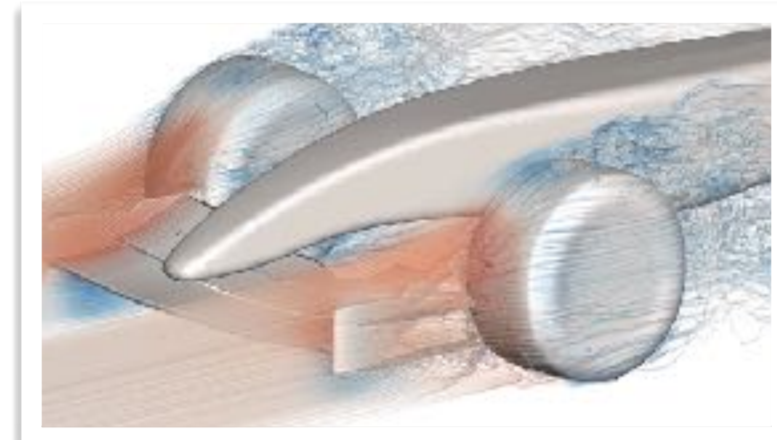


Bolis, Cantwell, Kirby, Sherwin Int J Num Meth. fluid, 2014

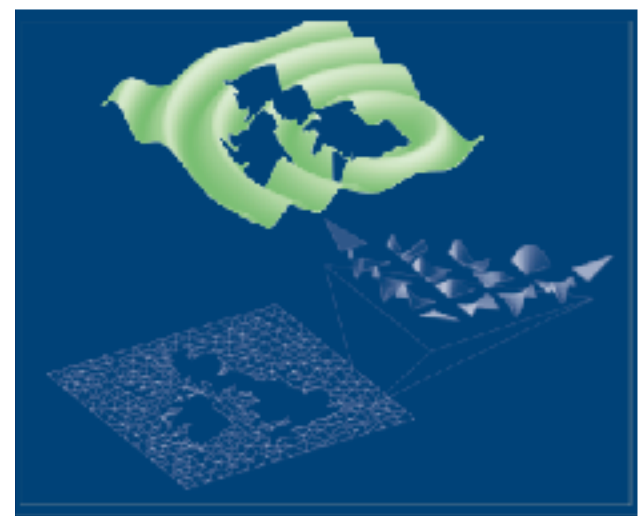
Nektar++: Spectral/hp element modelling

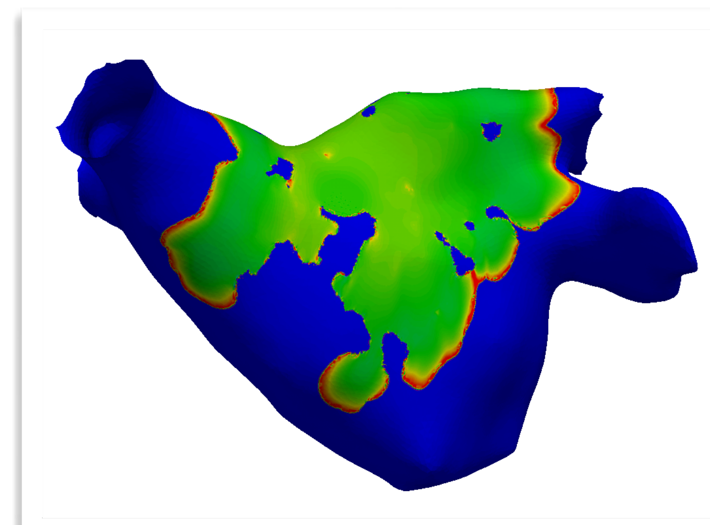


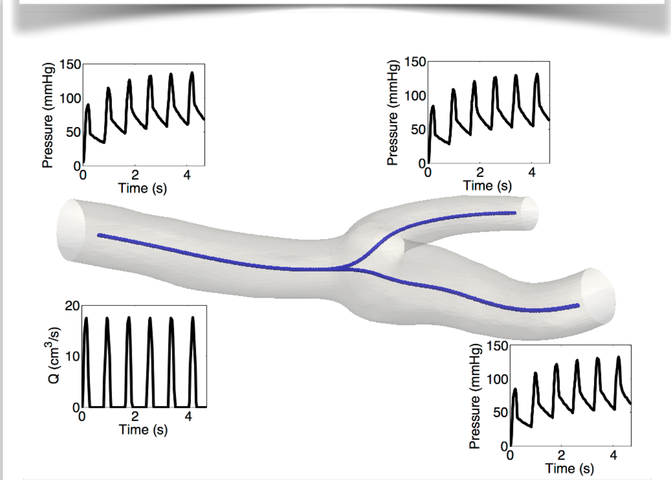


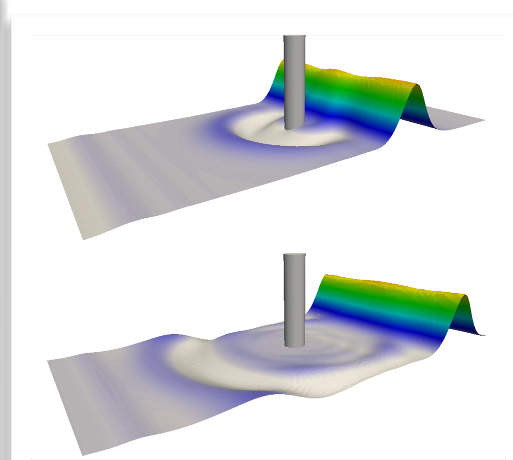


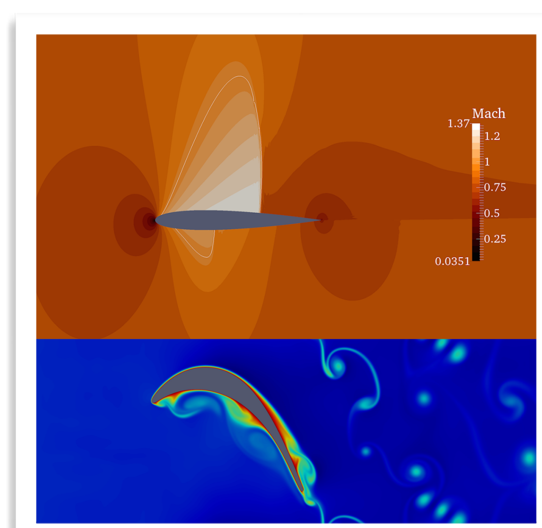
www.nektar.info

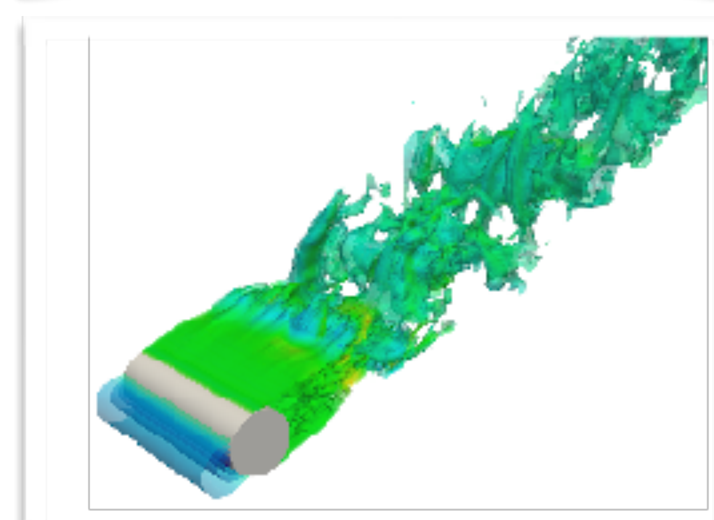




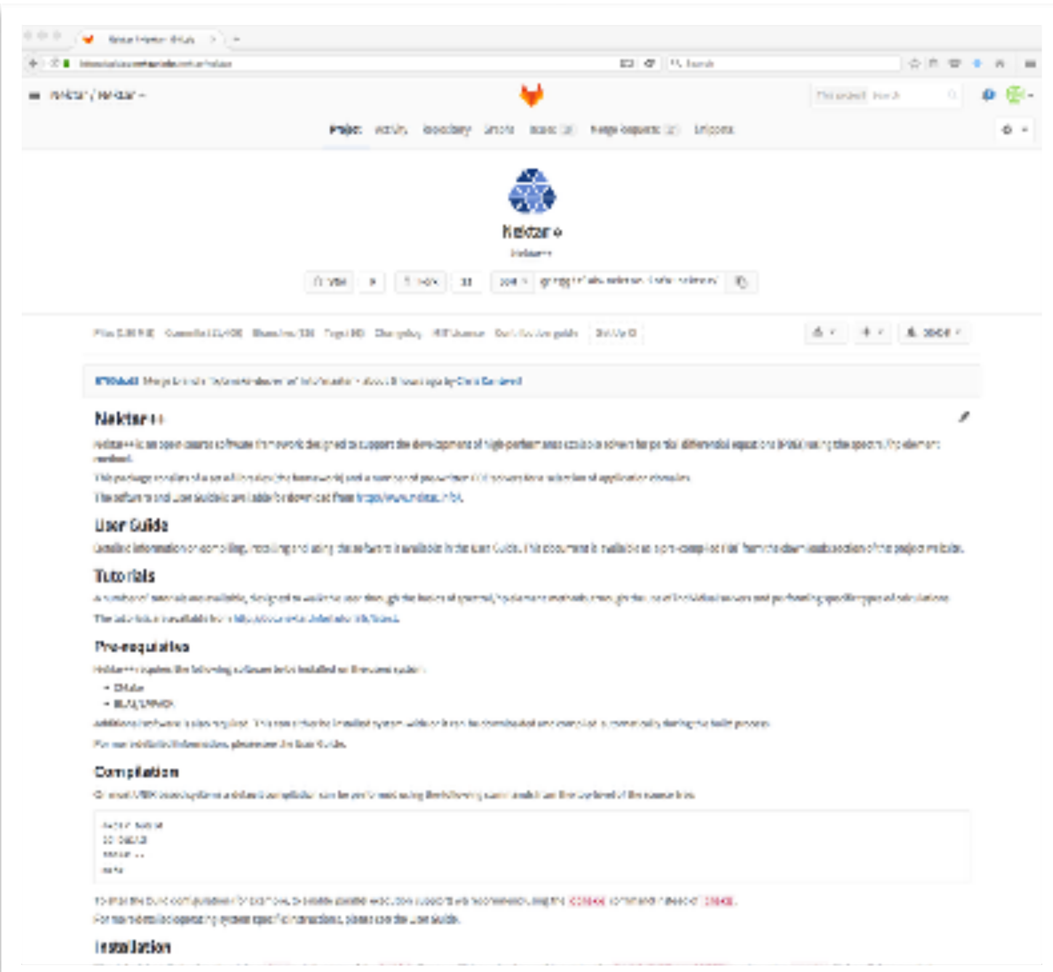




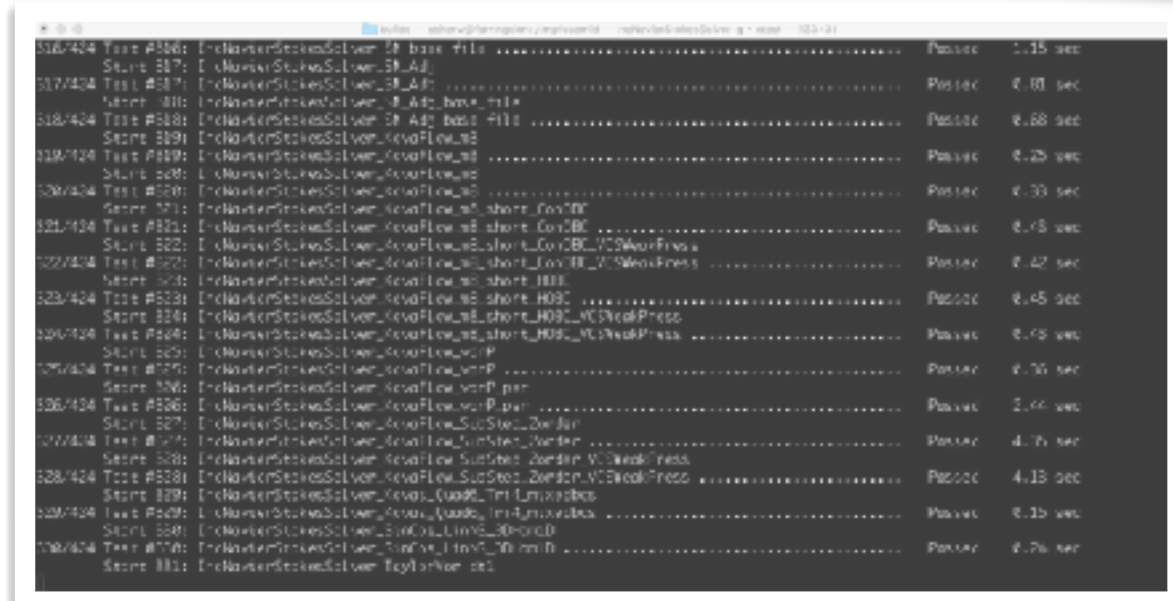




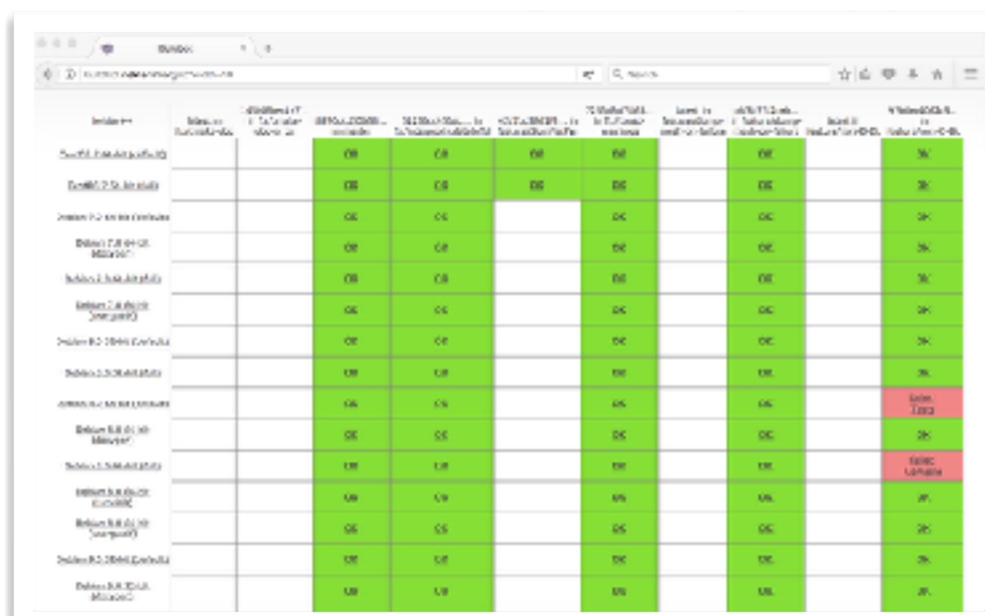
Open Source & Continual Integration



gitlab.nektar.info



ctest: 450 small tests
which can also act as demos

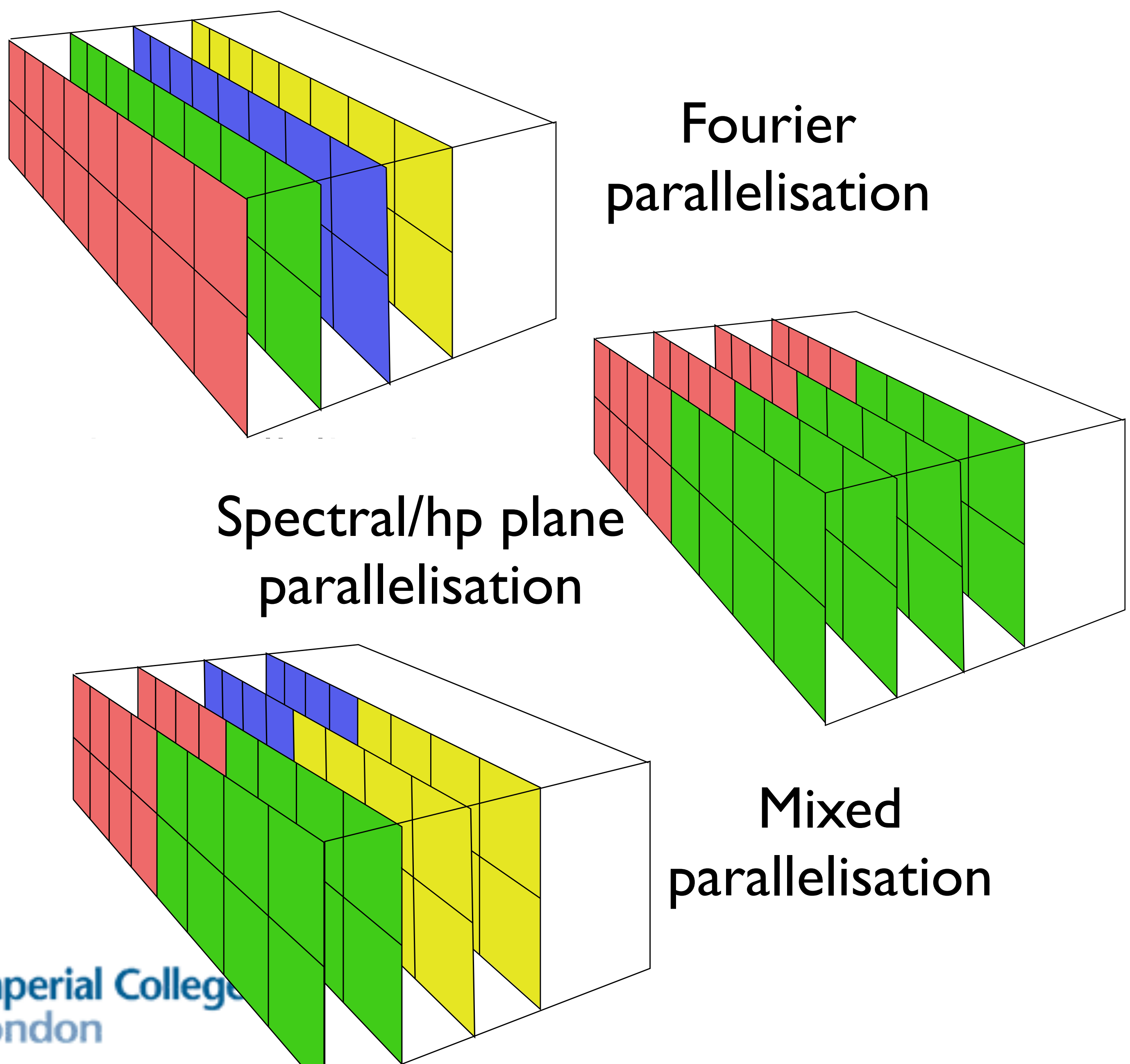


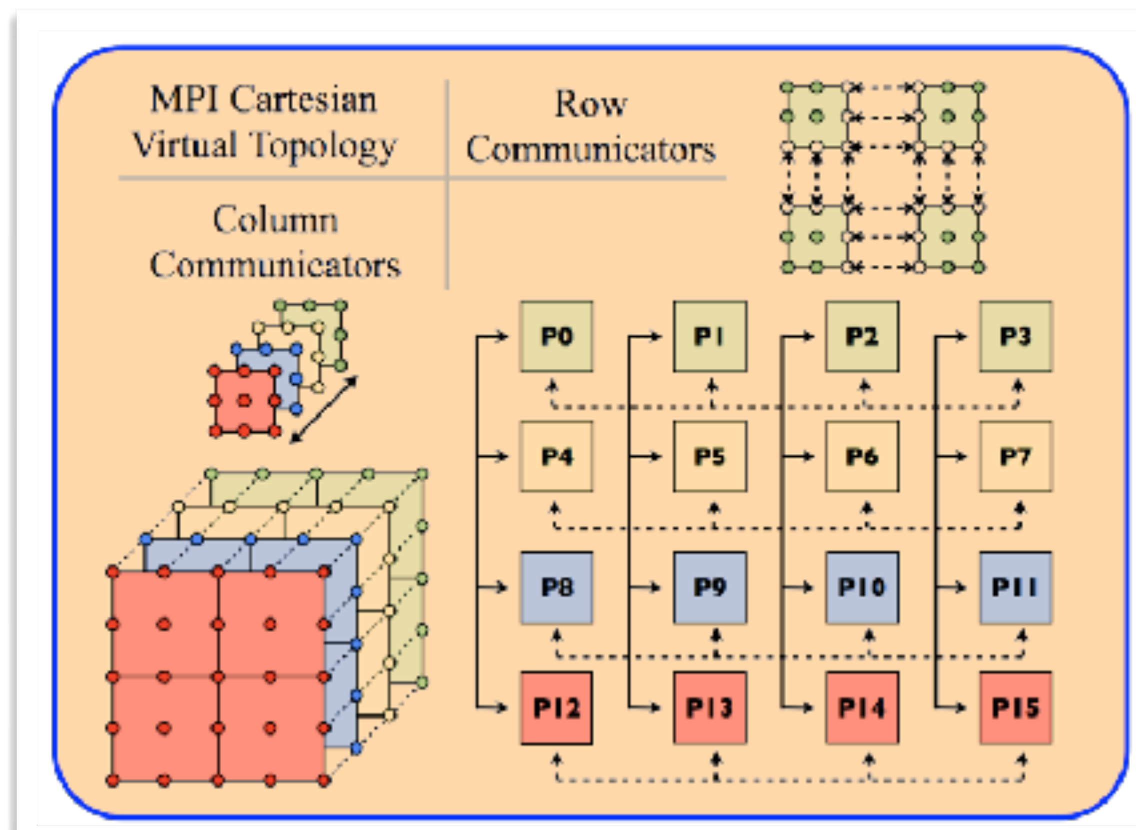
buildbot.
nektar.info

Linux: CentOS, Debian,
Fedora, Open SUSE
Mac: Mavericks,
Snow Leopard
Windows: XP, 7

Quasi-3D

- A. Bolis, C. Cantwell, D. Moxey, D. Serson, and S. Sherwin, “An adaptable parallel algorithm for the direct numerical simulation of incompressible turbulent flows using a fourier spectral/hp element method and mpi virtual topologies,” Computer physics communications, 2016.



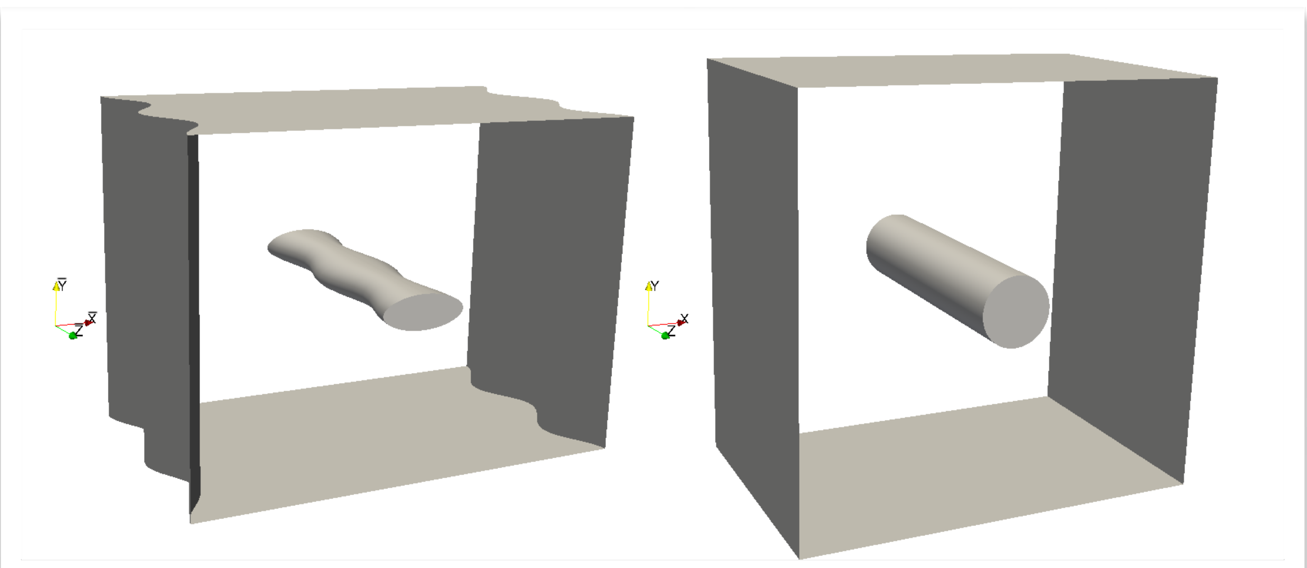


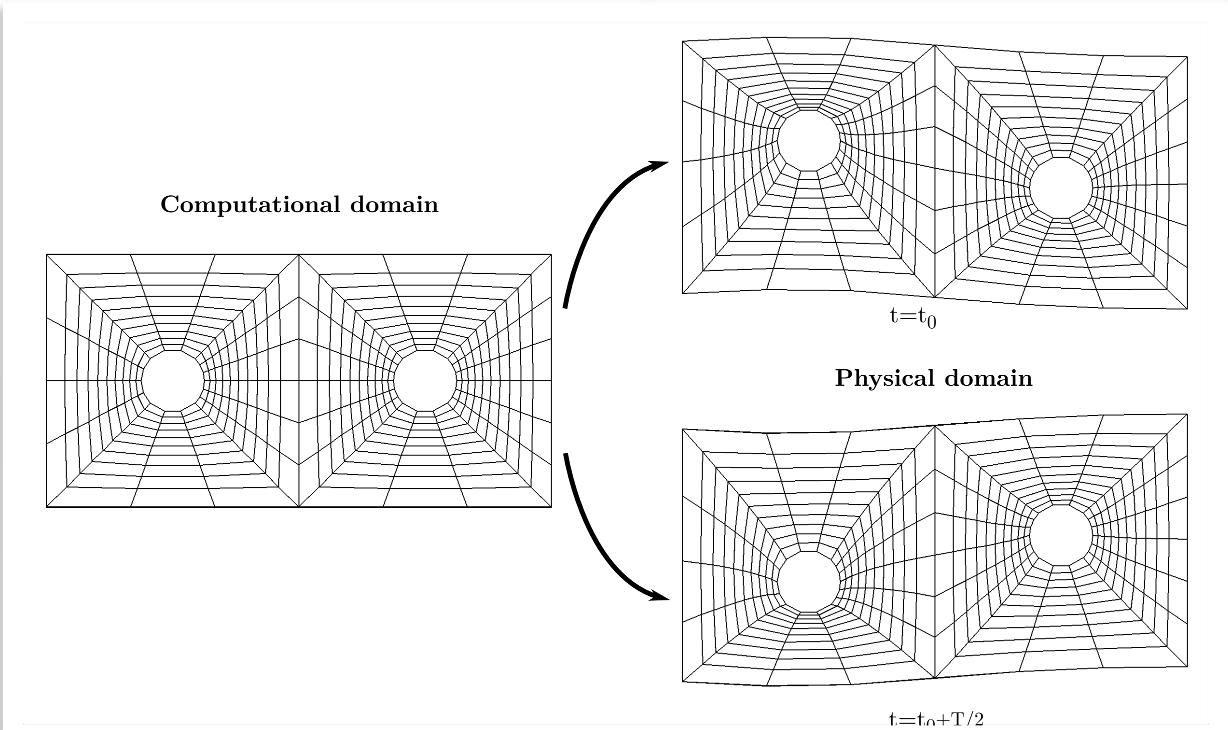
Imperial College London

UNIVERSITY OF EXETER

Generalised Mapping

- D. Serson, J. Meneghini, and S. Sherwin, “Velocity-correction schemes for the incompressible Navier-Stokes equations in general coordinate systems,” Journal of computational physics, 316, 243-254, 2016.



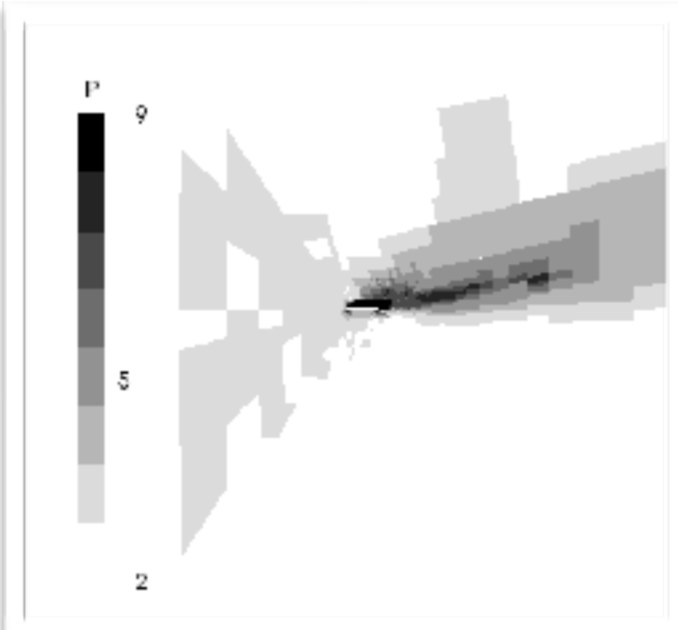


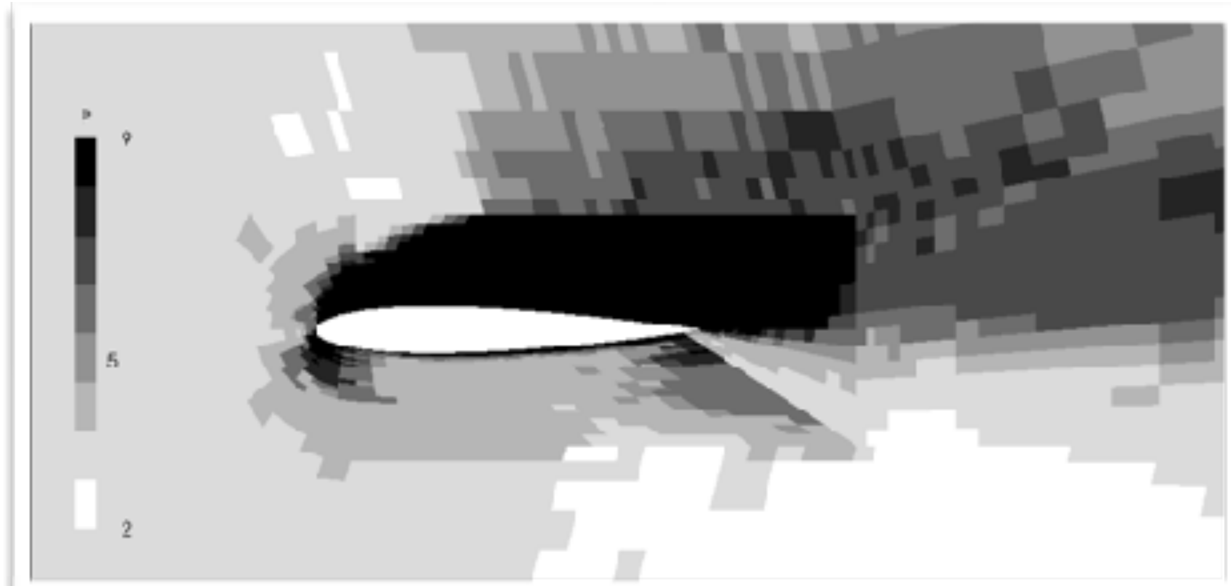
Imperial College London

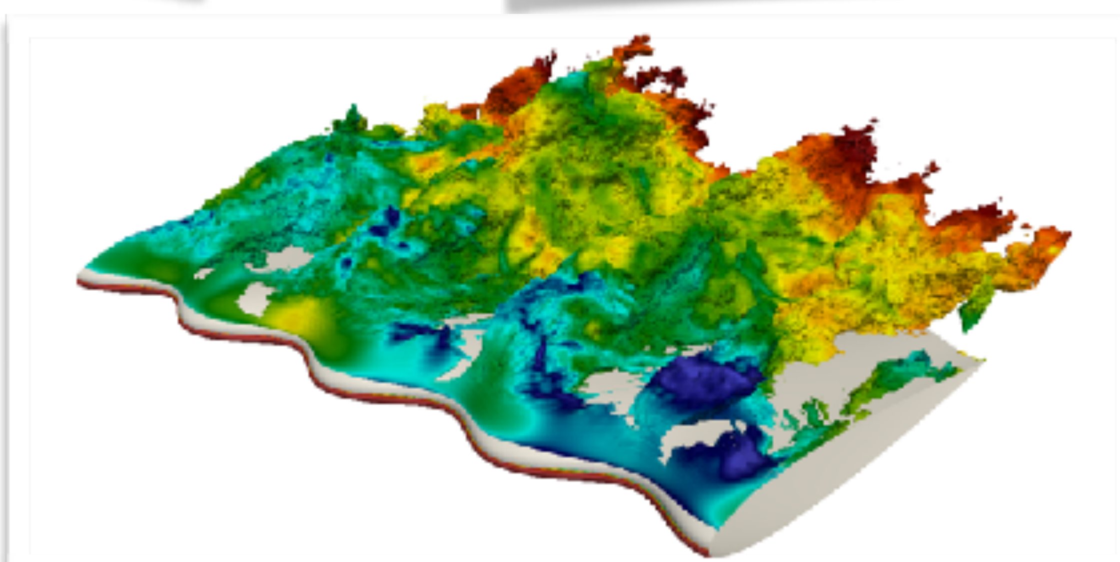
UNIVERSITY OF EXETER

Adaptive Polynomial Order

- Example: Naca0012 with Re=50,000 and alpha=15 (P_{min} = 2, P_{max} = 9)







- Variable P also available in 3D

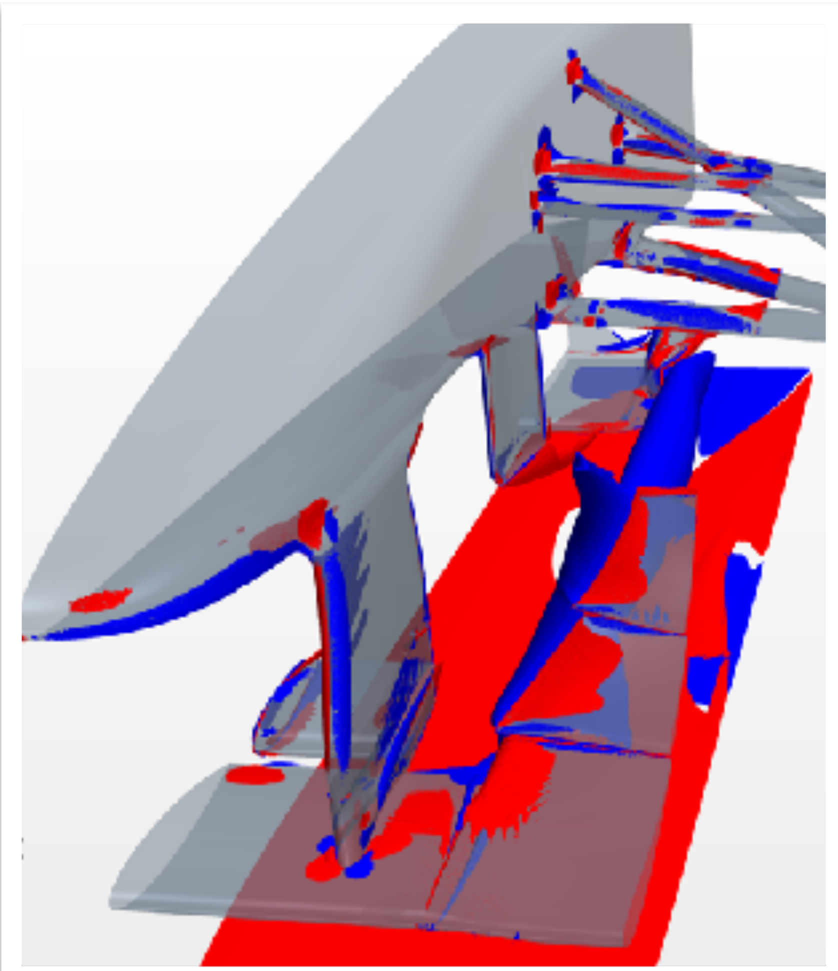
Imperial College London

Outline

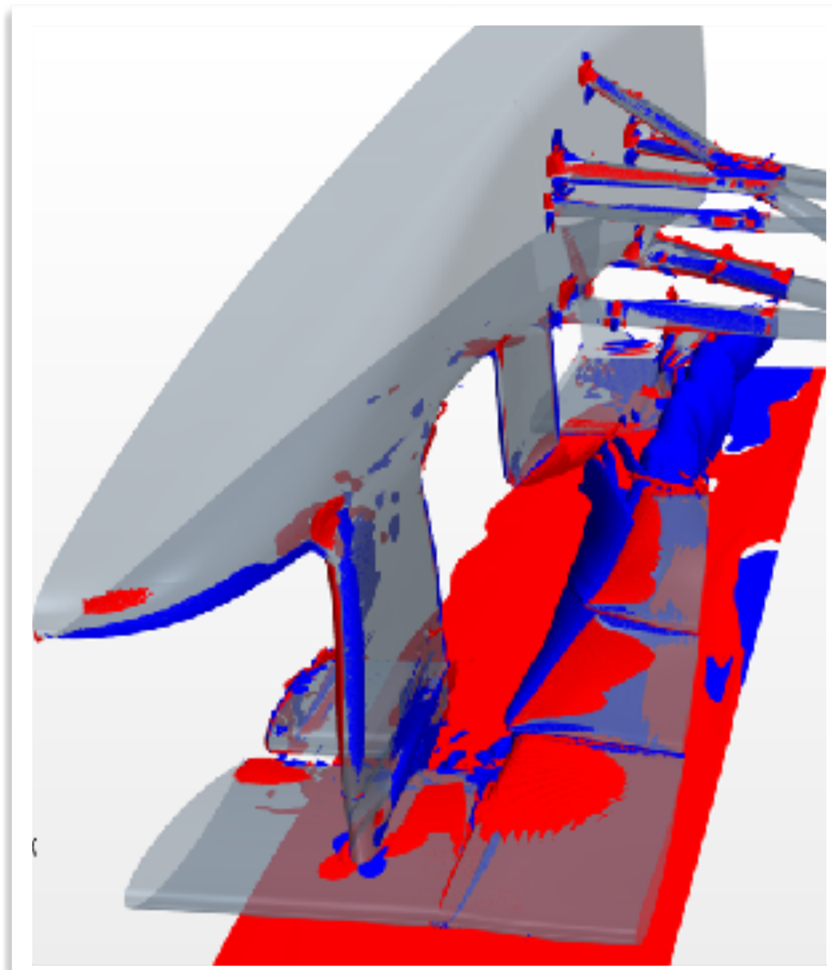
UNIVERSITY OF EXETER

- Spectral/hp element Modelling
  - Overview of Nektar++
- Challenges for modelling
  - Stabilisation and robustness
  - Performance
  - Mesh generation
- Applications
  - Hasegawa-Wakatani equations
  - Incompressible Flow Modelling

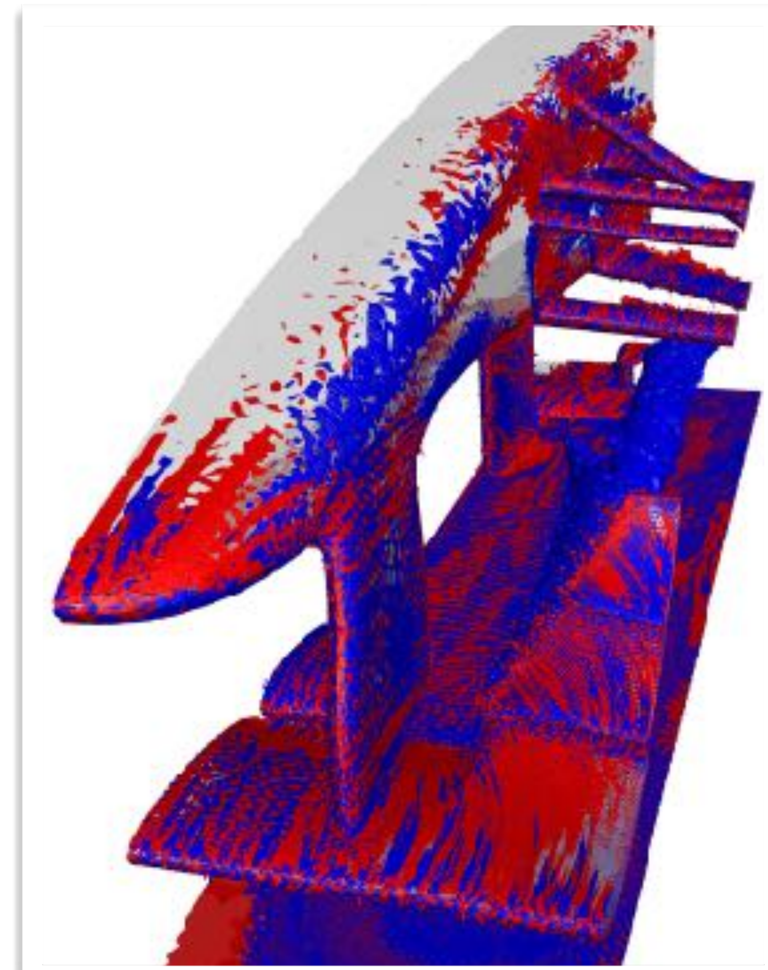
Scales of CFD



RANS

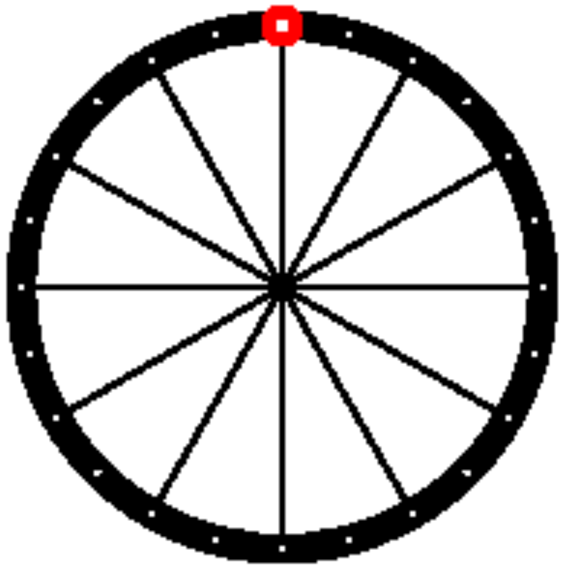


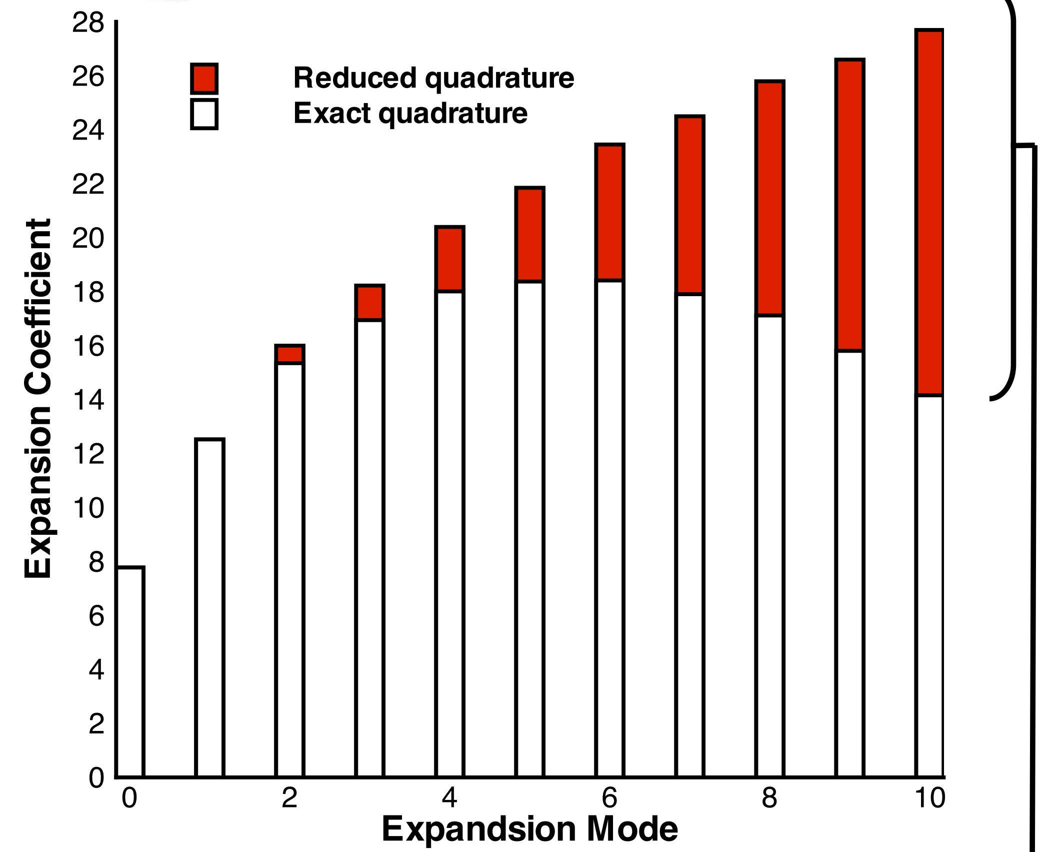
DES



iLES/uDNS

Aliasing



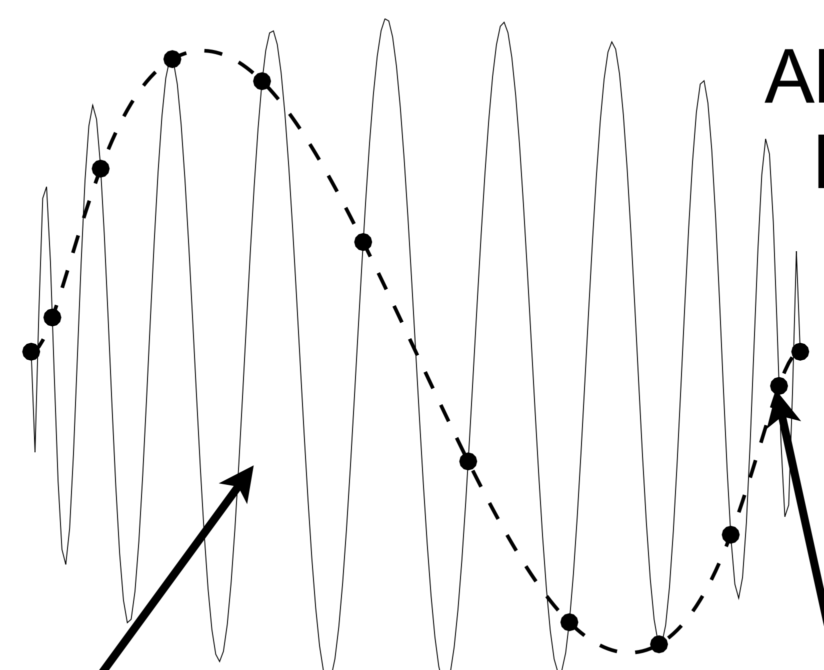


Example: $u(\xi) = \sum_{i=0}^{10} \phi_i(\xi)$

Galerkin projection of u^2 using:

- Q = 17 – exact Quadrature
- Q = 12 – sufficient for integrating 20th degree polynomials

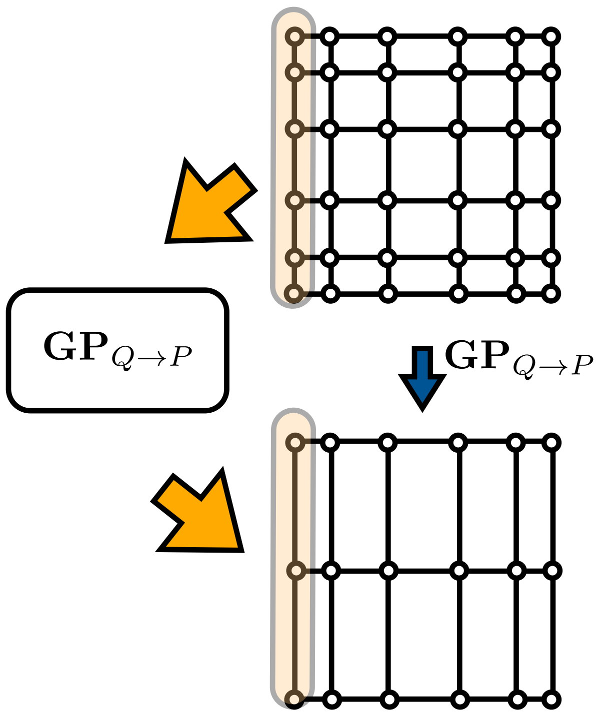
Aliasing Error

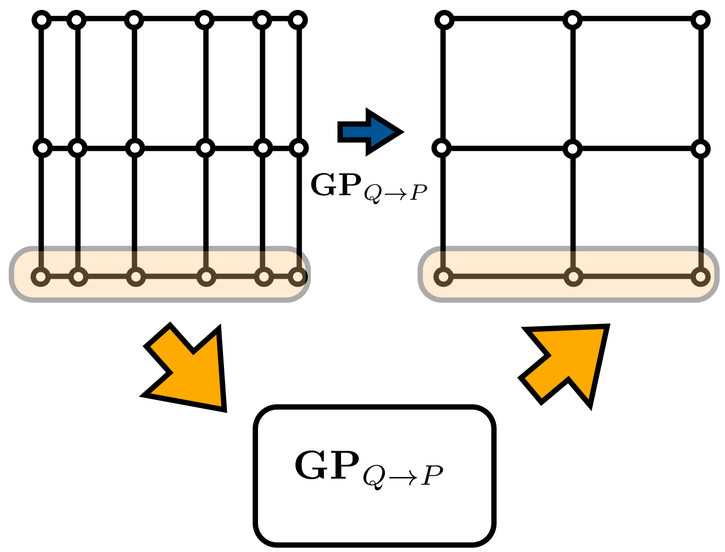


Example from Kirby & Karniadakis, J. Comp. Phys (2003)

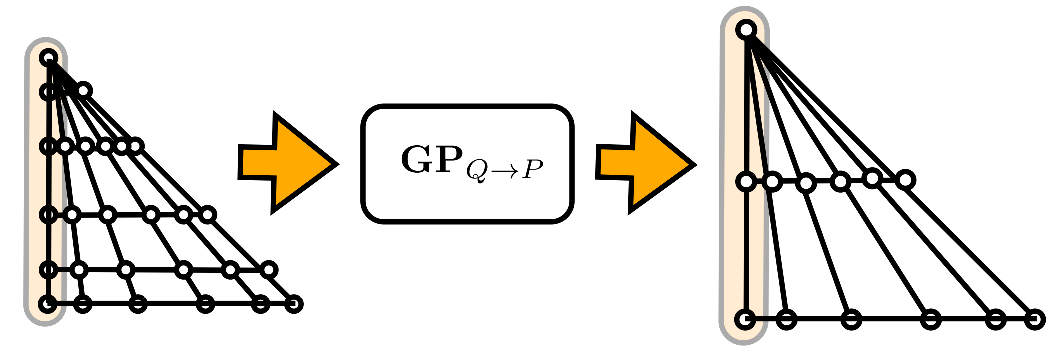
Imperial College London

Types of Aliasing

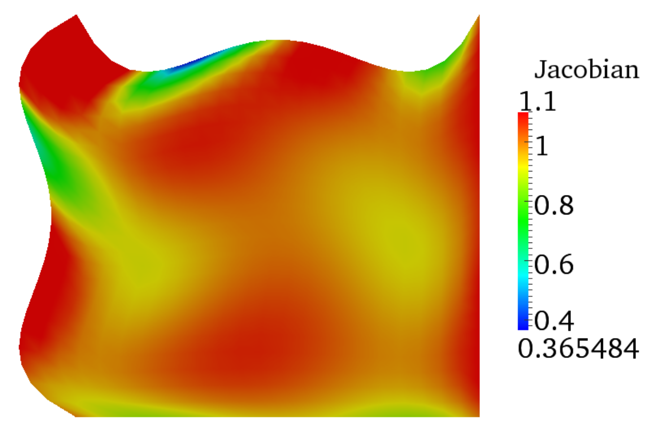


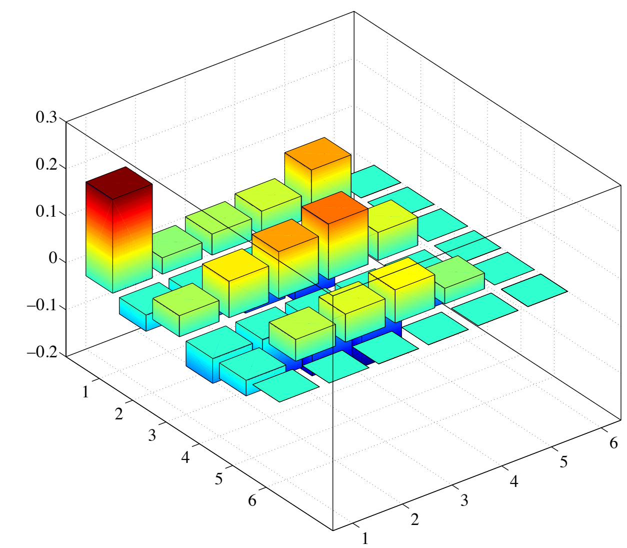


PDE non-linearities $u \cdot \nabla u$

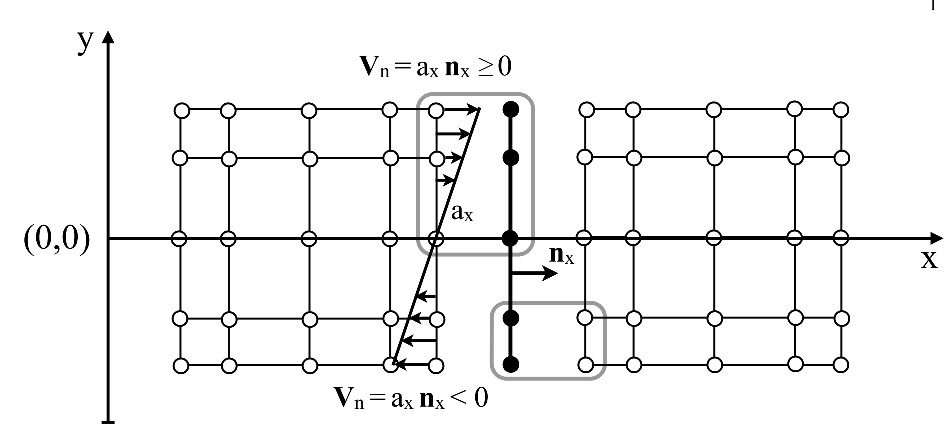


Geometric deformations





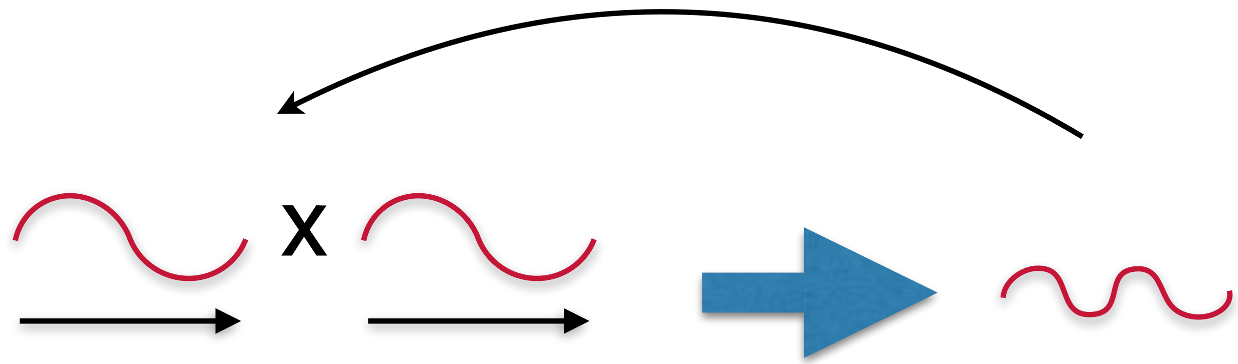
Interface aliasing

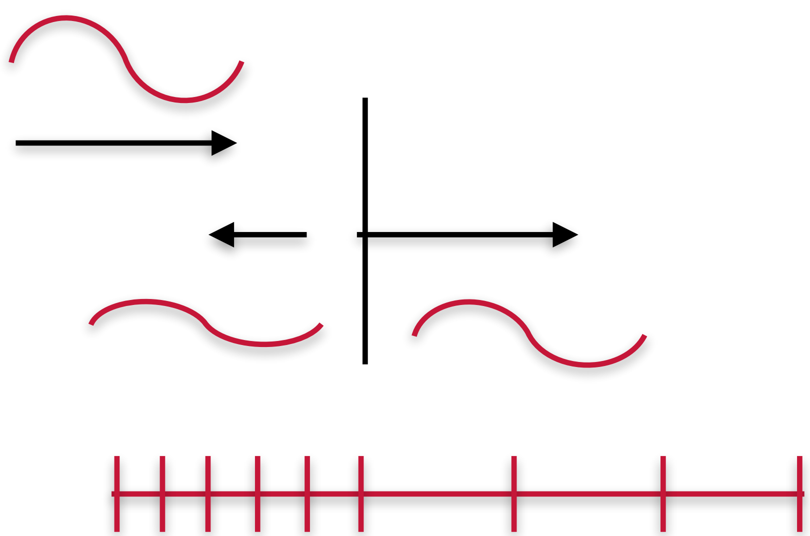


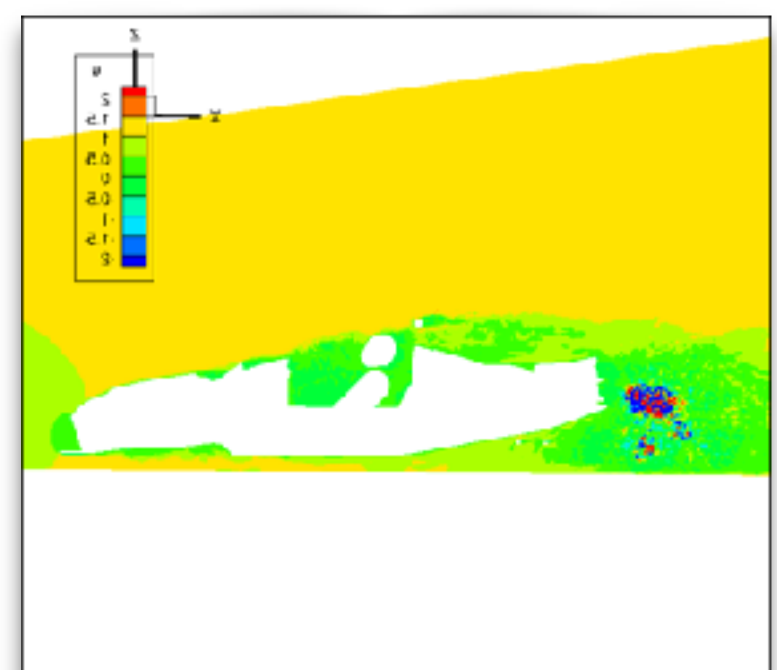
Imperial College London

G Mengaldo et al. J. Comp. Phys, 2015

Wave interaction & Wave trapping





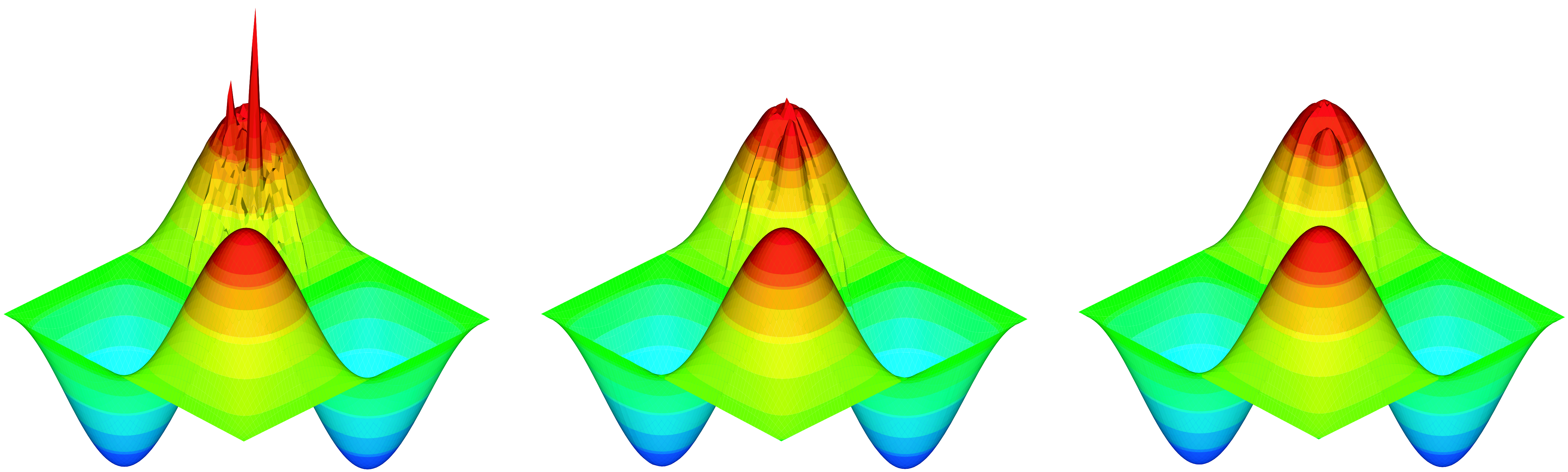


Imperial College
London

Spectral vanishing viscosity

Tadmor, (89) Maday, Kaber & Tadmor (93) :

$$\frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}\left[\frac{u^2(x,t)}{2}\right] = \epsilon\frac{\partial}{\partial x}\left[Q_\epsilon\frac{\partial u}{\partial x}\right] \qquad \hat{Q}_k = e^{-\frac{(k-N)^2}{(k-P_{\mathrm{cut}})^2}}, \quad k > P_{\mathrm{cut}}.$$



No SVV

$P_{cut} = 7,$ $\epsilon_{SVV} = 0.1$

$P_{cut} = 3,$ $\epsilon_{SVV} = 0.1$

Eigenanalysis informed SVV

R. Moura, J. Peiro, S. Sherwin

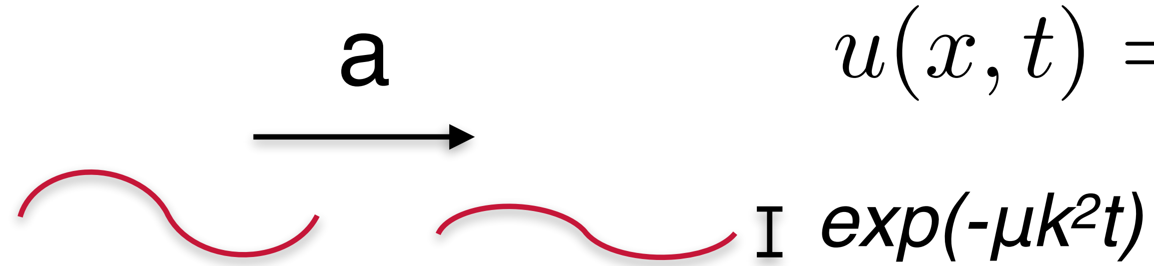
- Mengaldo G, De Grazia D, Moura RC, Sherwin SJ, 2018, Spatial eigensolution analysis of energy-stable flux reconstruction schemes and influence of the numerical flux on accuracy and robustness, Journal of Computational Physics, 358, 1-20
- Mengaldo G, Moura RC, Giralda B, Peiró J, Sherwin SJ 2017, Spatial eigensolution analysis of discontinuous Galerkin schemes with practical insights for under-resolved computations and implicit LES, Computers and Fluids, 0045-7930
- Moura RC, Sherwin SJ, Peiro J, 2016, Eigensolution analysis of spectral/hp continuous Galerkin approximations to advection-diffusion problems: Insights into spectral vanishing viscosity, Journal of Computational Physics, 307, 401-422
- Moura RC, Sherwin SJ, Peiro J, 2015, Linear dispersion-diffusion analysis and its application to under-resolved turbulence simulations using discontinuous Galerkin spectral/hp methods, Journal of Computational Physics, 298, 695-710,

Eigenanalysis

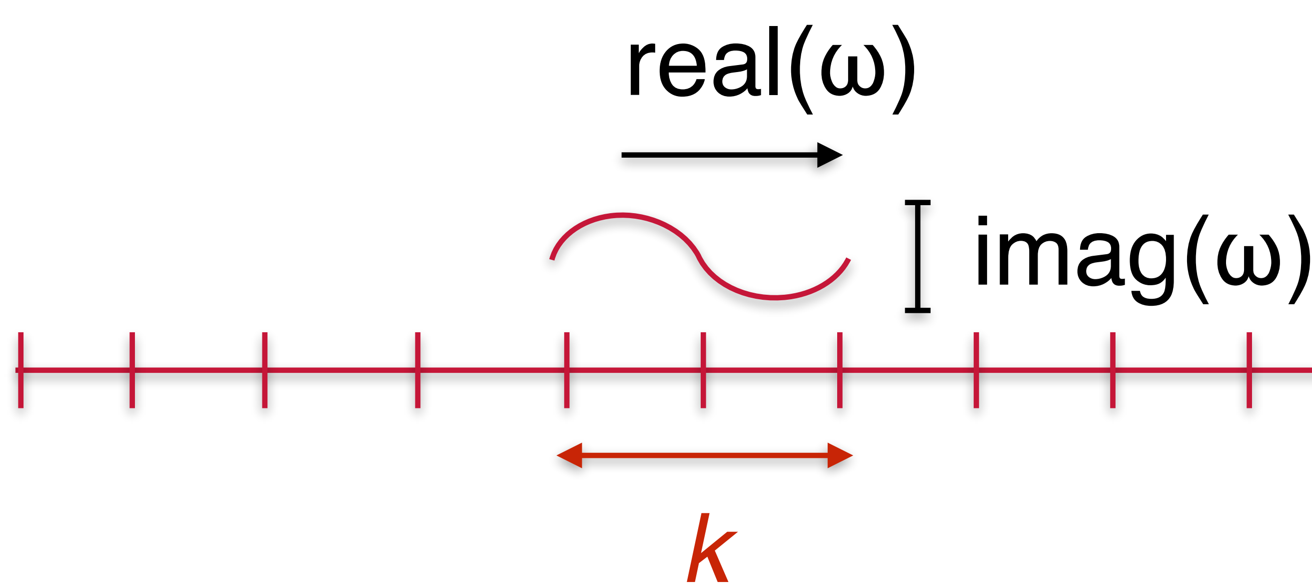
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu \frac{\partial^2 u}{\partial x^2}$$

$$u(x,t) = \hat{u} e^{i(kx - \omega t)}$$

$$i\omega\hat{u} = (\mu k^2 + iak)\hat{u}$$



Temporal analysis: (*k* real find ω = ω(k))



Imperial College London

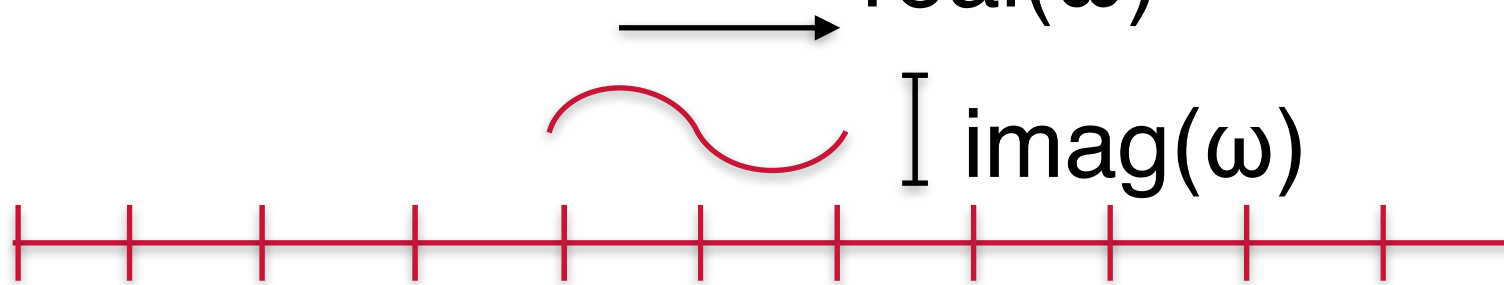
Eigenanalysis

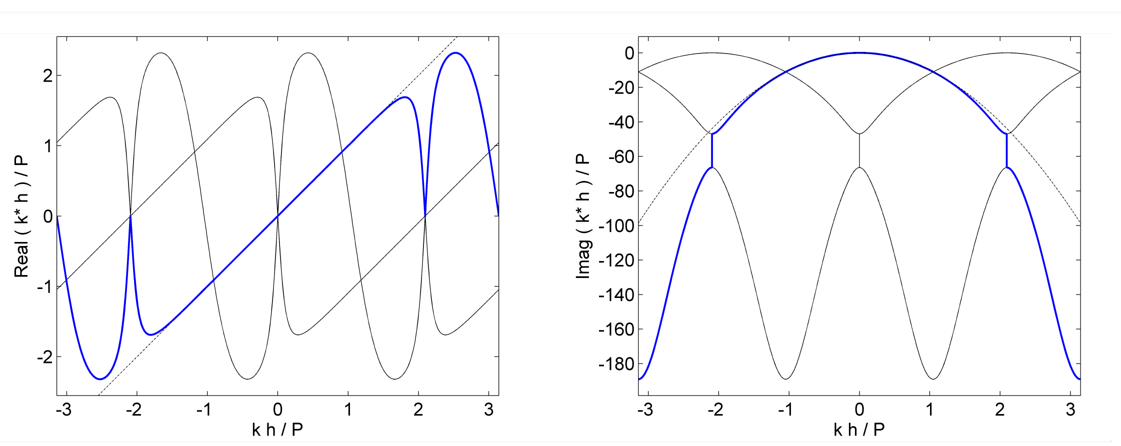
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}$$

$$u(x,t) = \hat{u}e^{i(kx-\omega t)}$$

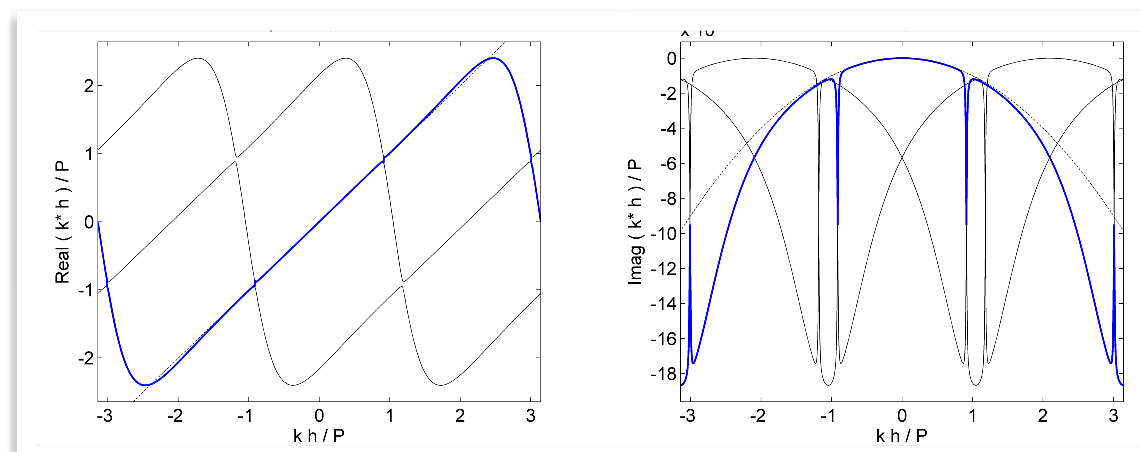
$$i\omega\hat{u} = (\mu k^2 + iak)\hat{u}$$

Temporal analysis:

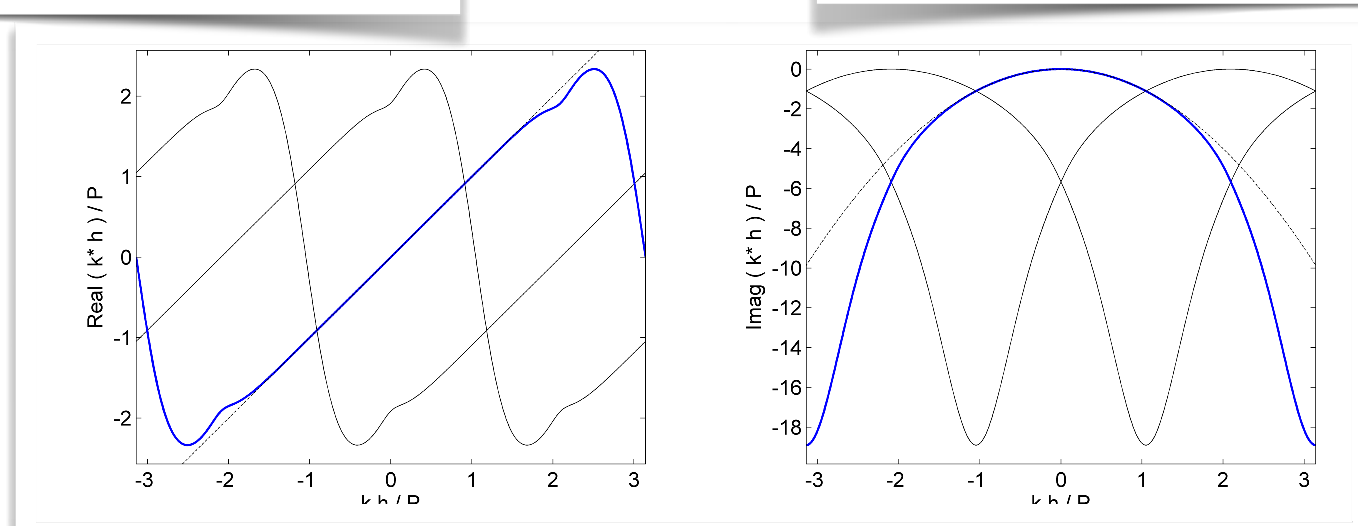




Pe=0.1



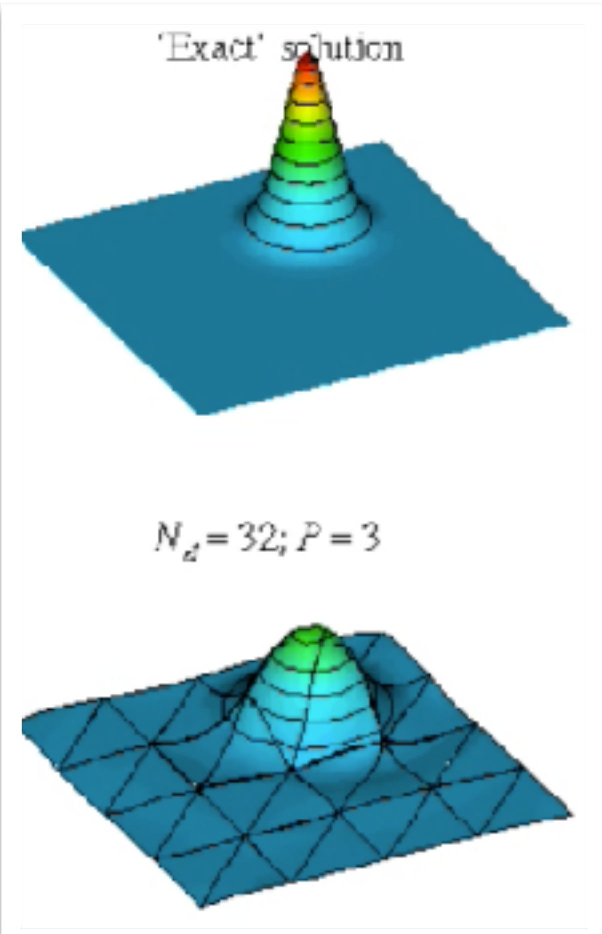
Pe=1000



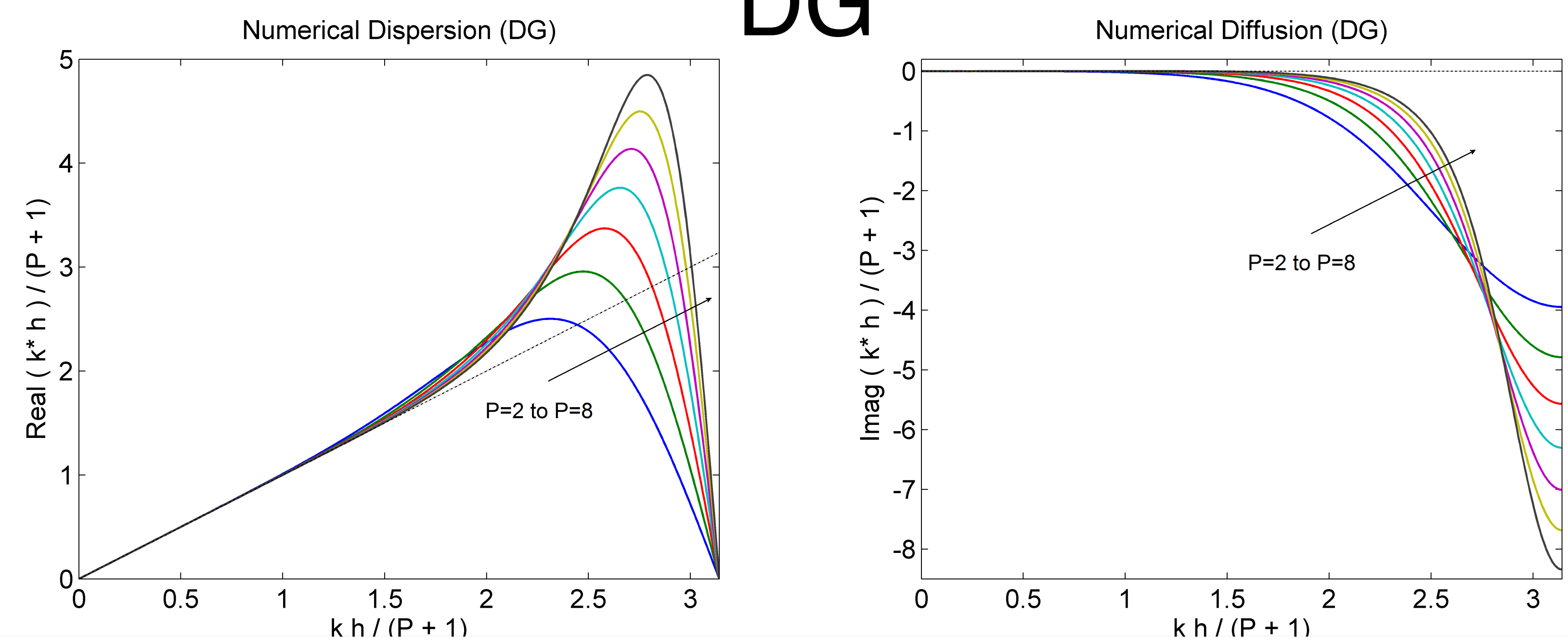
Pe = (a h/p)/μ=1

Choose Pe ~1

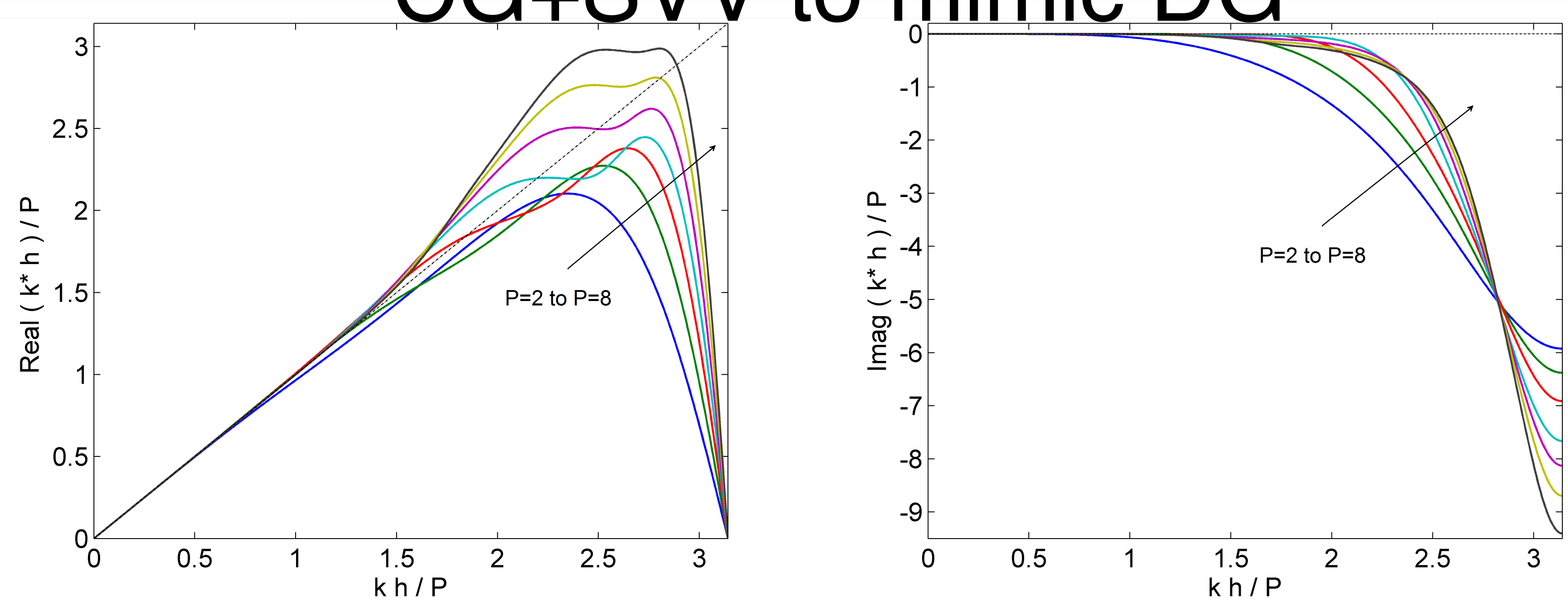
DG-like properties of CG+SVV



DG



CG+SVV to mimic DG



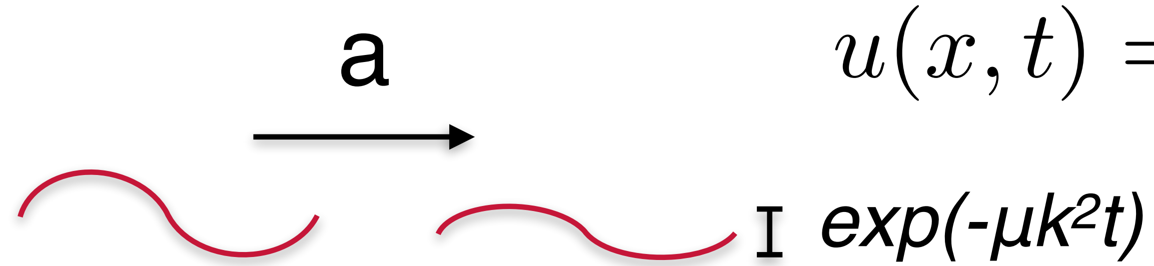
R. C. Moura, S. J. Sherwin, and J. Peiró, J. computational physics, 307. 401-422, 2016.

Eigenanalysis

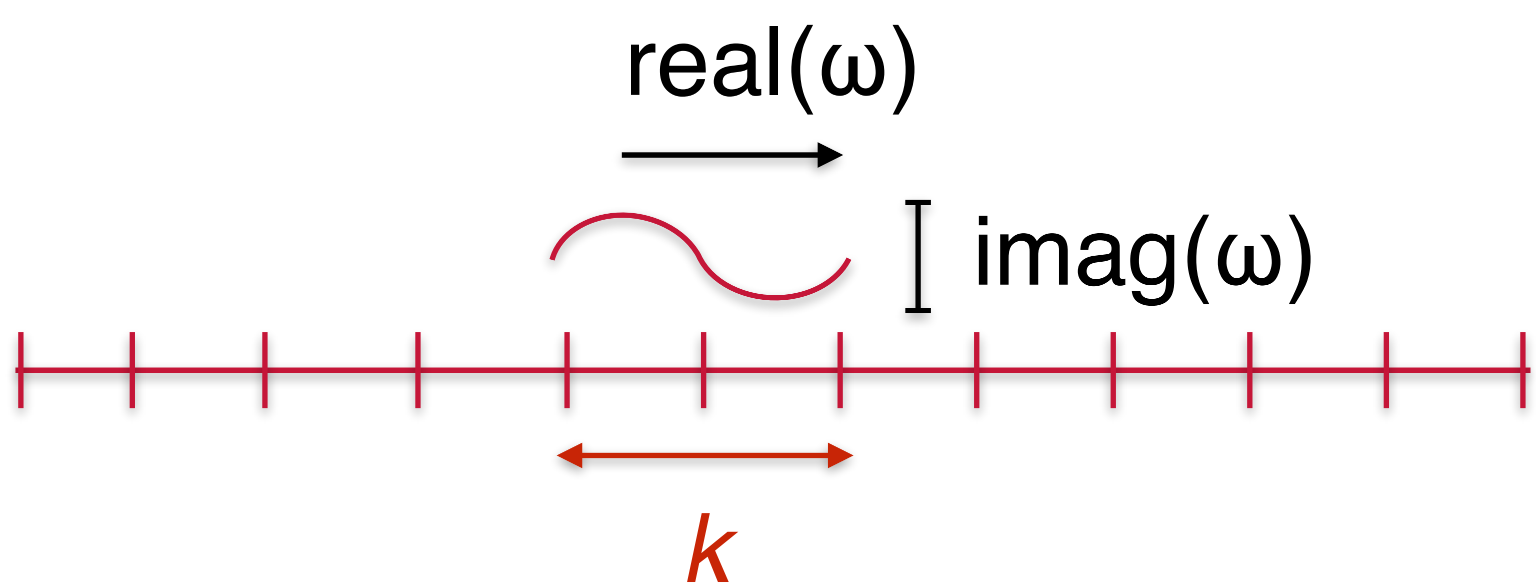
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}$$

$$u(x,t) = \hat{u}e^{i(kx-\omega t)}$$

$$i\omega\hat{u} = (\mu k^2 + iak)\hat{u}$$



Temporal analysis: (*k* real find ω = ω(k))



Imperial College London

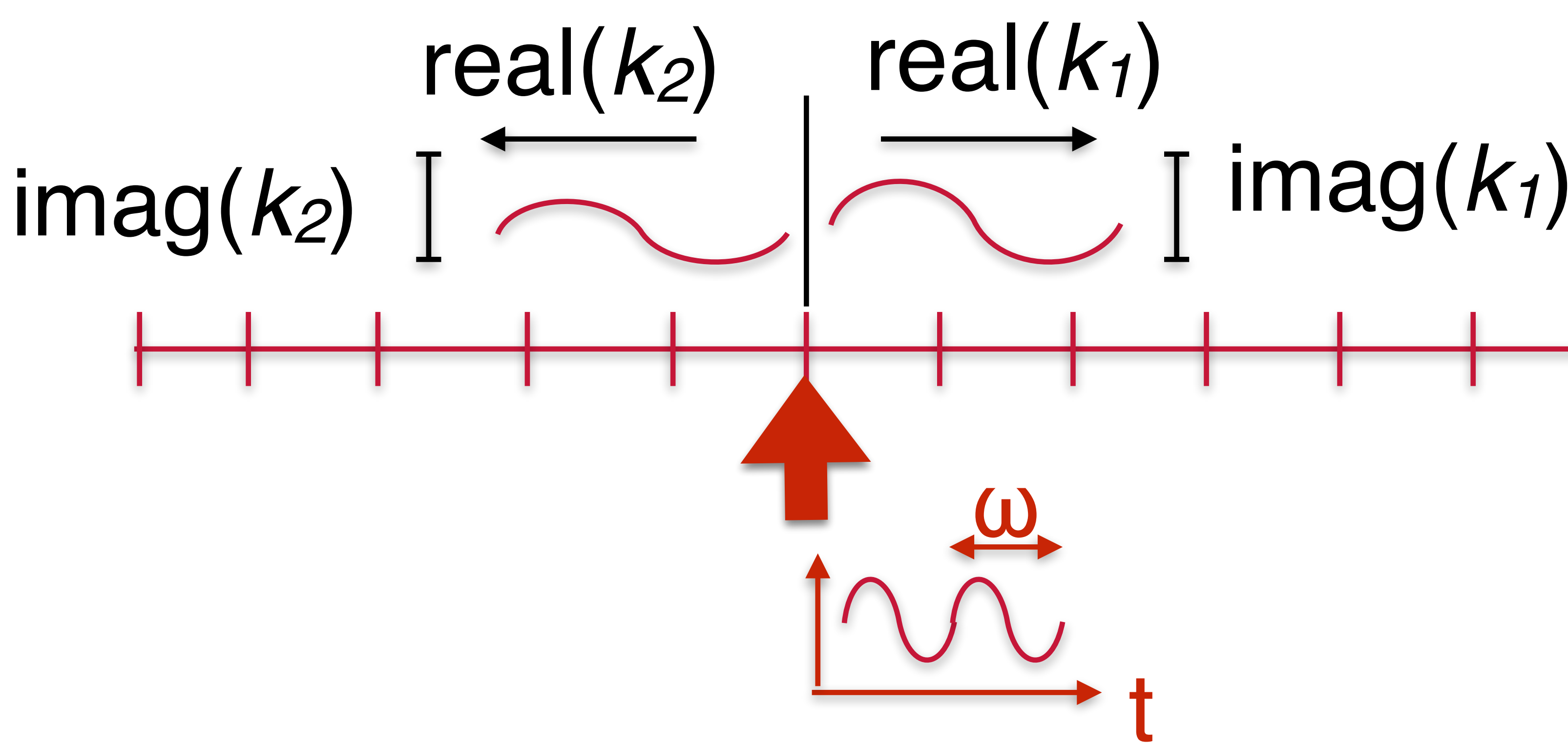
Eigenanalysis

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}$$

$$u(x,t) = \hat{u}e^{i(kx-\omega t)}$$

$$i\omega\hat{u} = (\mu k^2 + iak)\hat{u}$$

Spatial analysis: (ω real find $k = k(\omega)$)



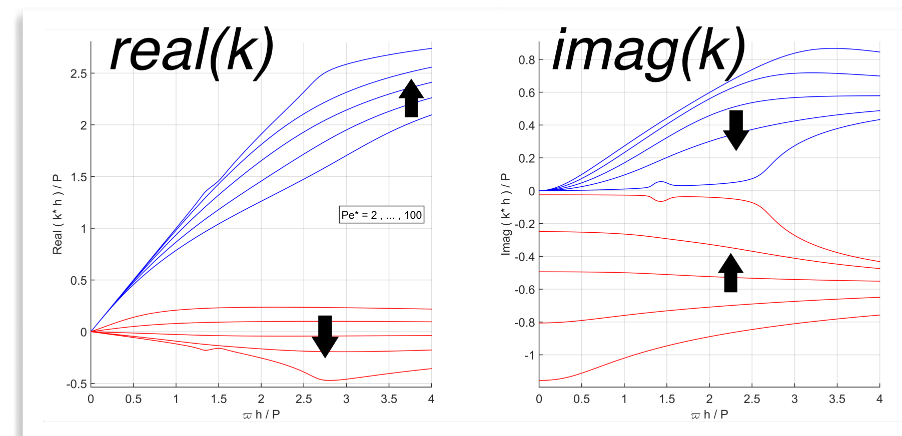
Eigenanalysis

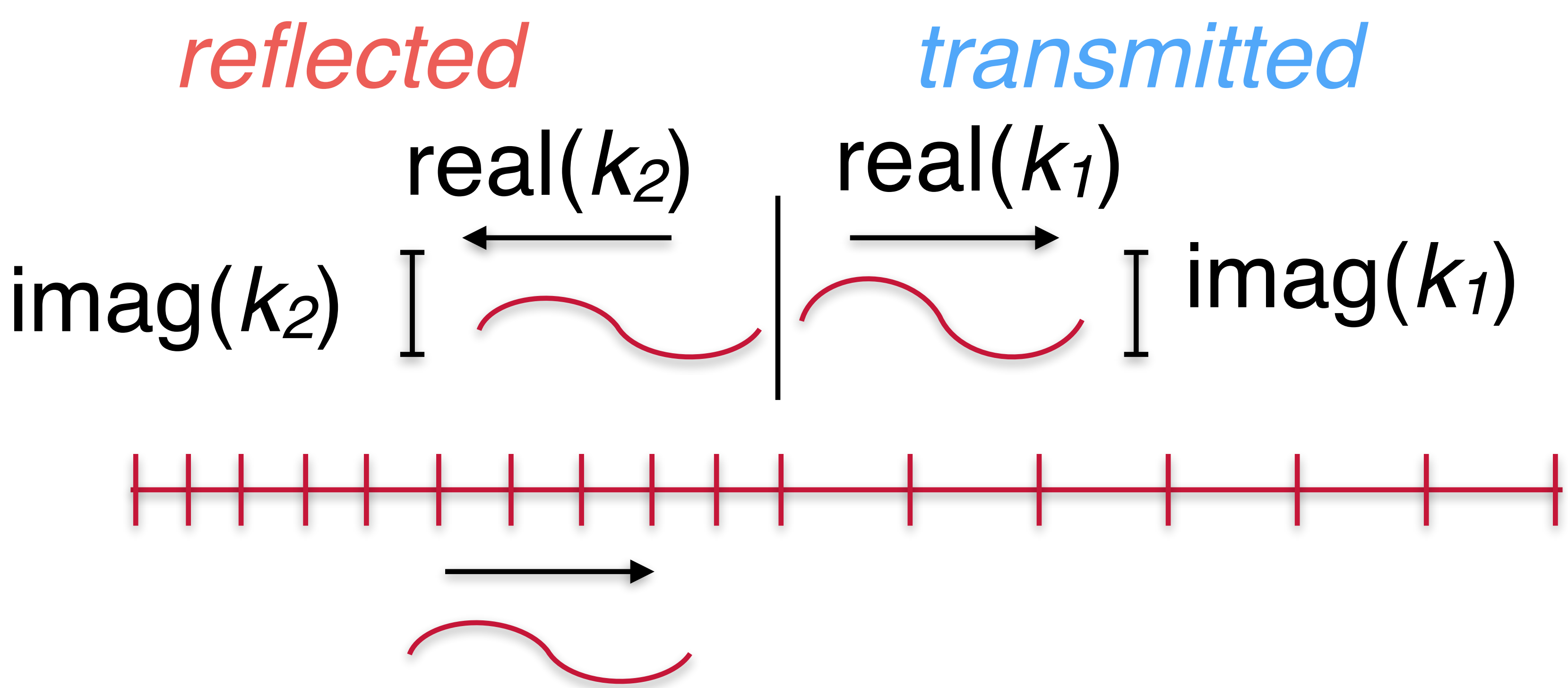
$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}$$

$$u(x,t) = \hat{u}e^{i(kx-\omega t)}$$

$$i\omega\hat{u} = (\mu k^2 + iak)\hat{u}$$

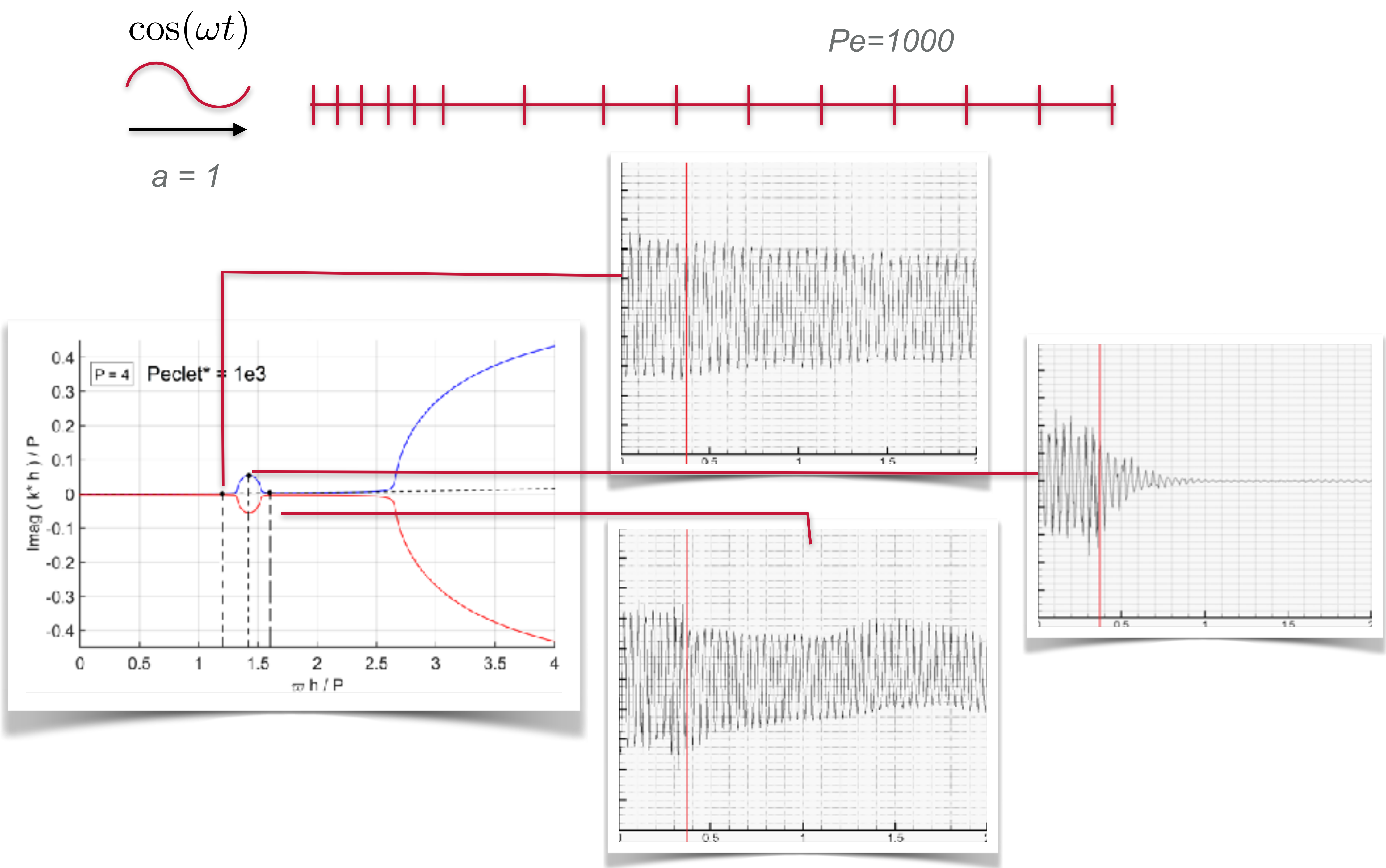
Spatial analysis: (ω real find $k = k(\omega)$)





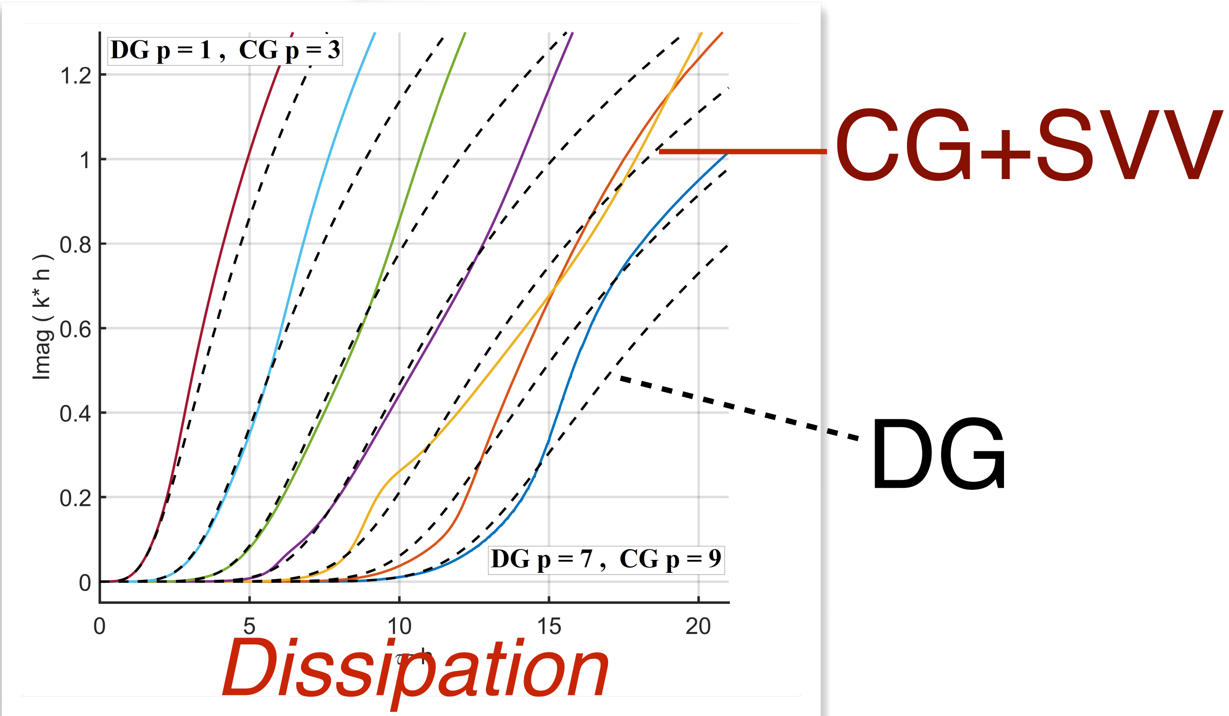
Imperial College London

Example of CG or DG (Hyper upwinding) problem

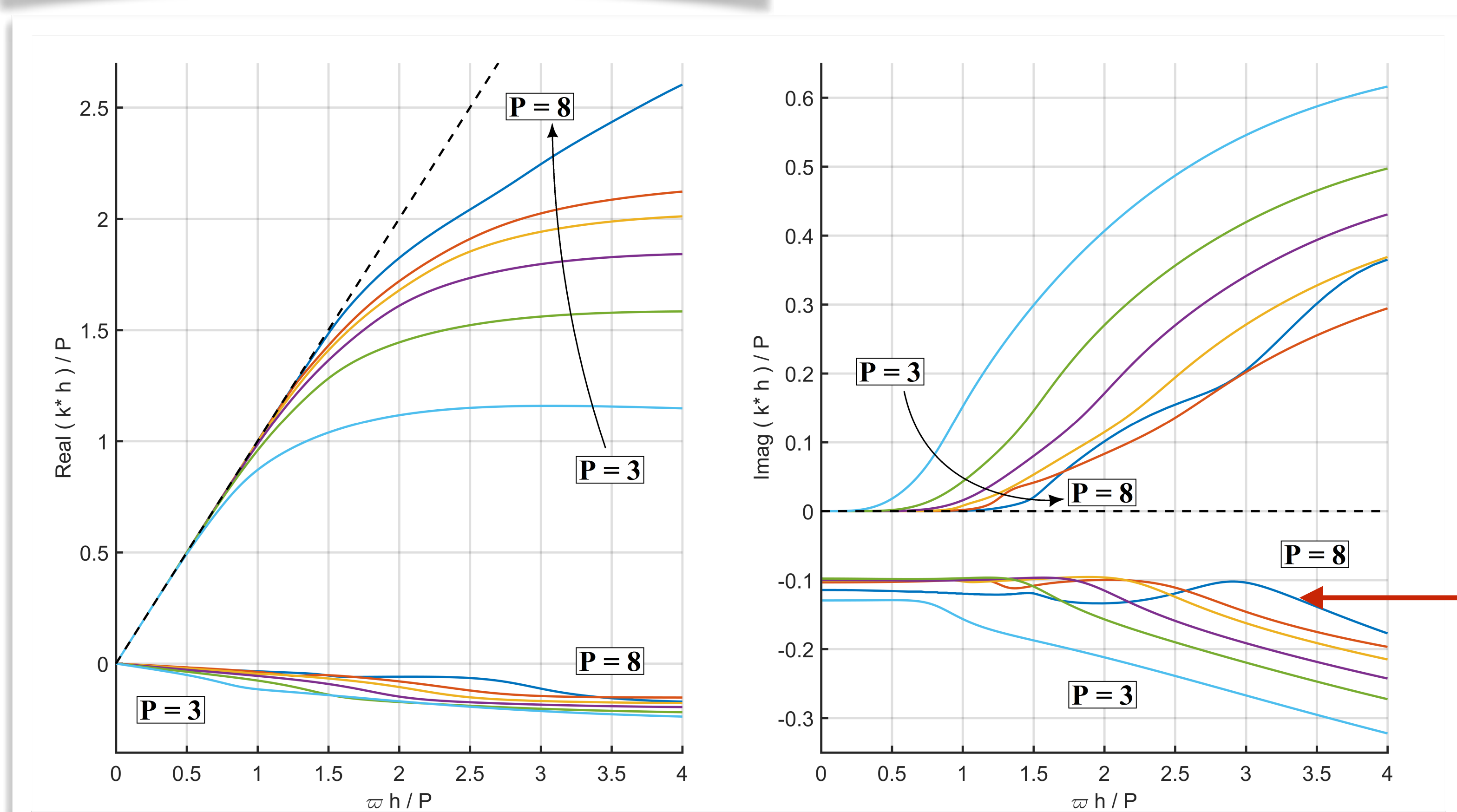


Imperial College London

DG-like properties of CG+SVV



- Match log-log slope near origin
- Match CG+SVV at P with DG at P-2 to remove bubbles
- Have higher diffusion at higher frequencies
- SVV coefficient scaled to maintain mesh Peclet no.



Reflected waves have strong diffusion enforced as part of optimisation

Outline

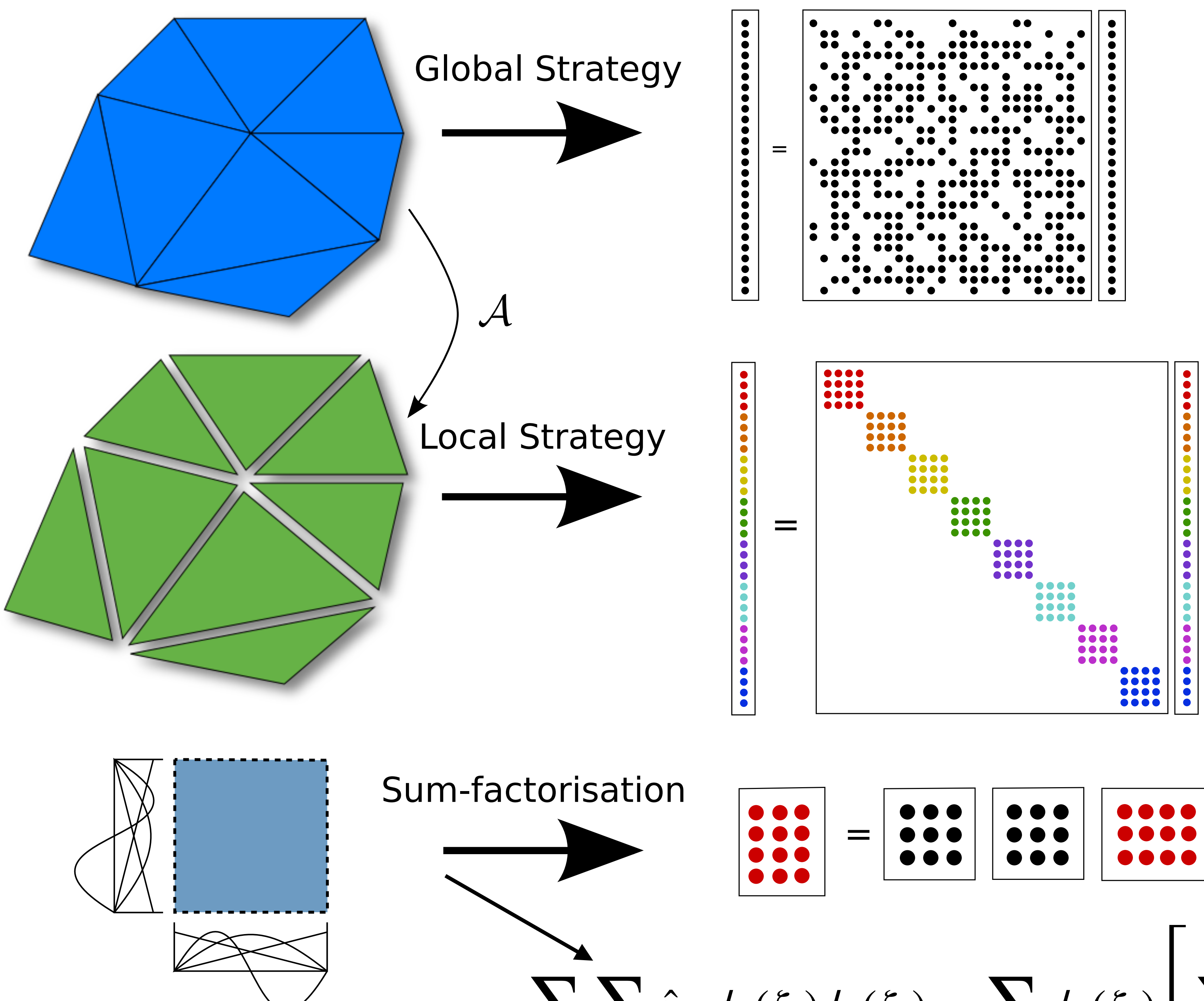
- Spectral/hp element Modelling
  - Overview of Nektar++
- Challenges for modelling
  - Stabilisation and robustness
  - Performance
  - Mesh generation
- Applications
  - Hasegawa-Wakatani equations
  - Incompressible Flow Modelling

Challenges in modern HPC

- Ideally: we want really fast single-core nodes with lots of memory bandwidth
- Instead, we get lots of FLOPS using many cores per node, lower clock speed
- Complicated memory hierarchies
- Very limited memory bandwidth

Need algorithms with high arithmetic intensities that can use available FLOPS: high-order methods

Performance



$$\sum_p \sum_q \hat{u}_{pq}\phi_p(\xi_1)\phi_q(\xi_2) = \sum_p \phi_p(\xi_1)\left[\sum_q \hat{u}_{pq}\phi_q(\xi_2)\right]$$

Increasing polynomial order

Single-core HPC

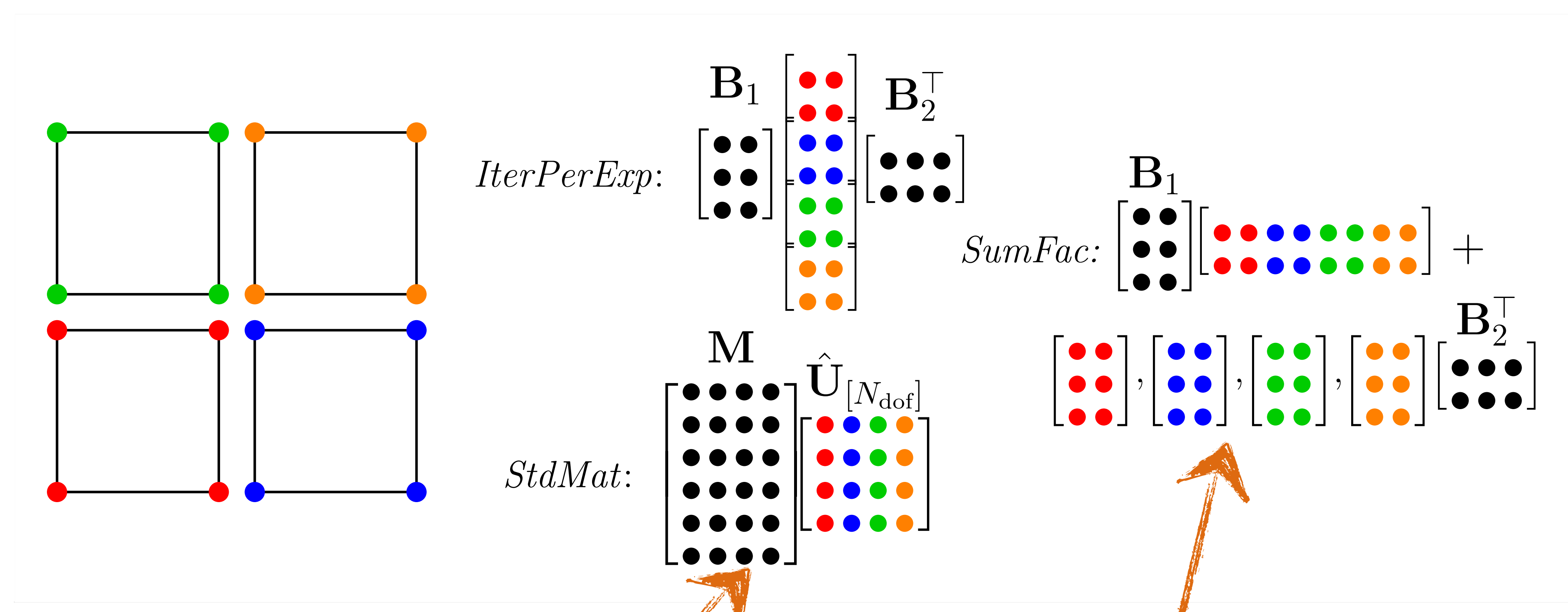
More localised memory access

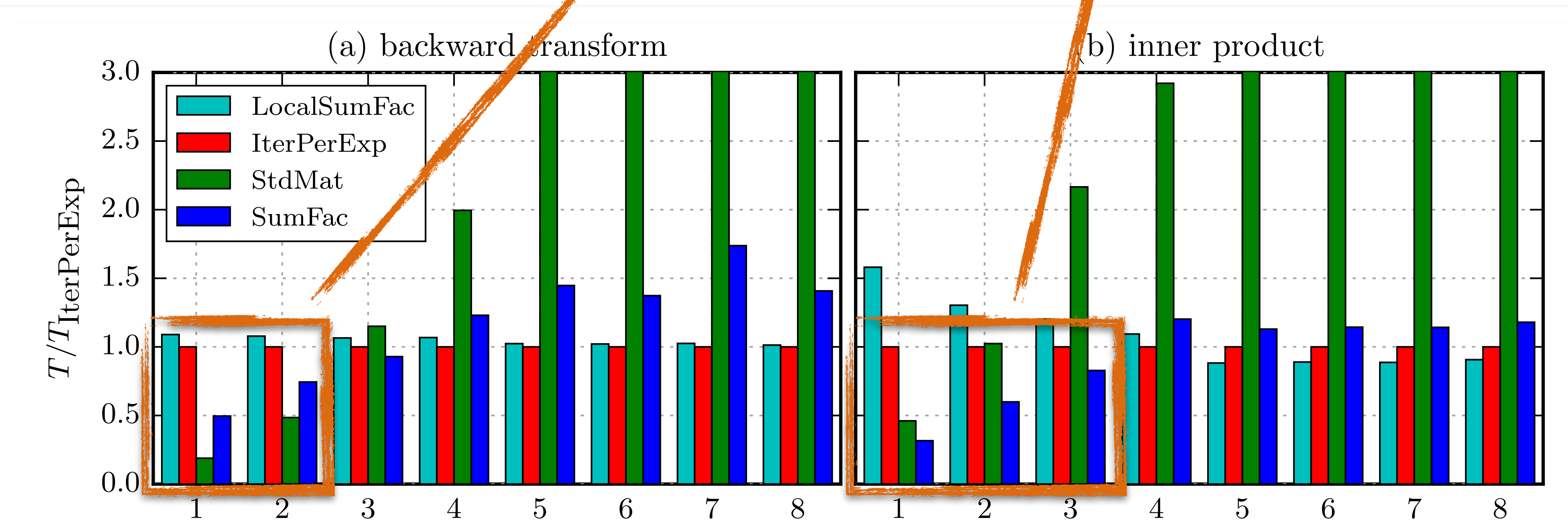
Many-core HPC

Imperial College London

UNIVERSITY OF EXETER

Implementations





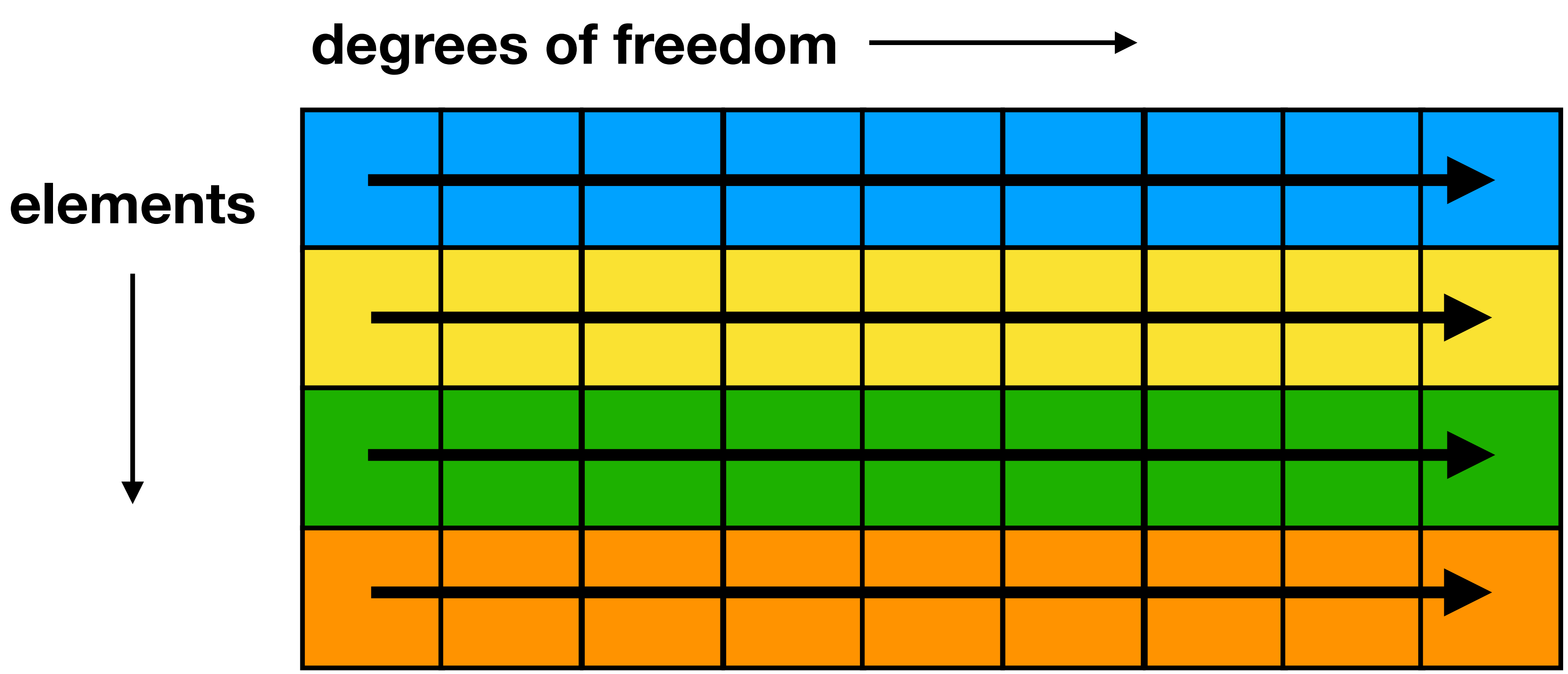
Moxey et al. CMAME, 2016

Exploiting vectorisation

- Key to achieving peak performance is exploiting vectorisation (SSE, FMA, AVX, AVX-512, …)
- Recent work has focused on achieving this with tuned kernels for key operators
- Particular focus on ‘matrix-free’ approaches that avoid construction of a matrix per-element

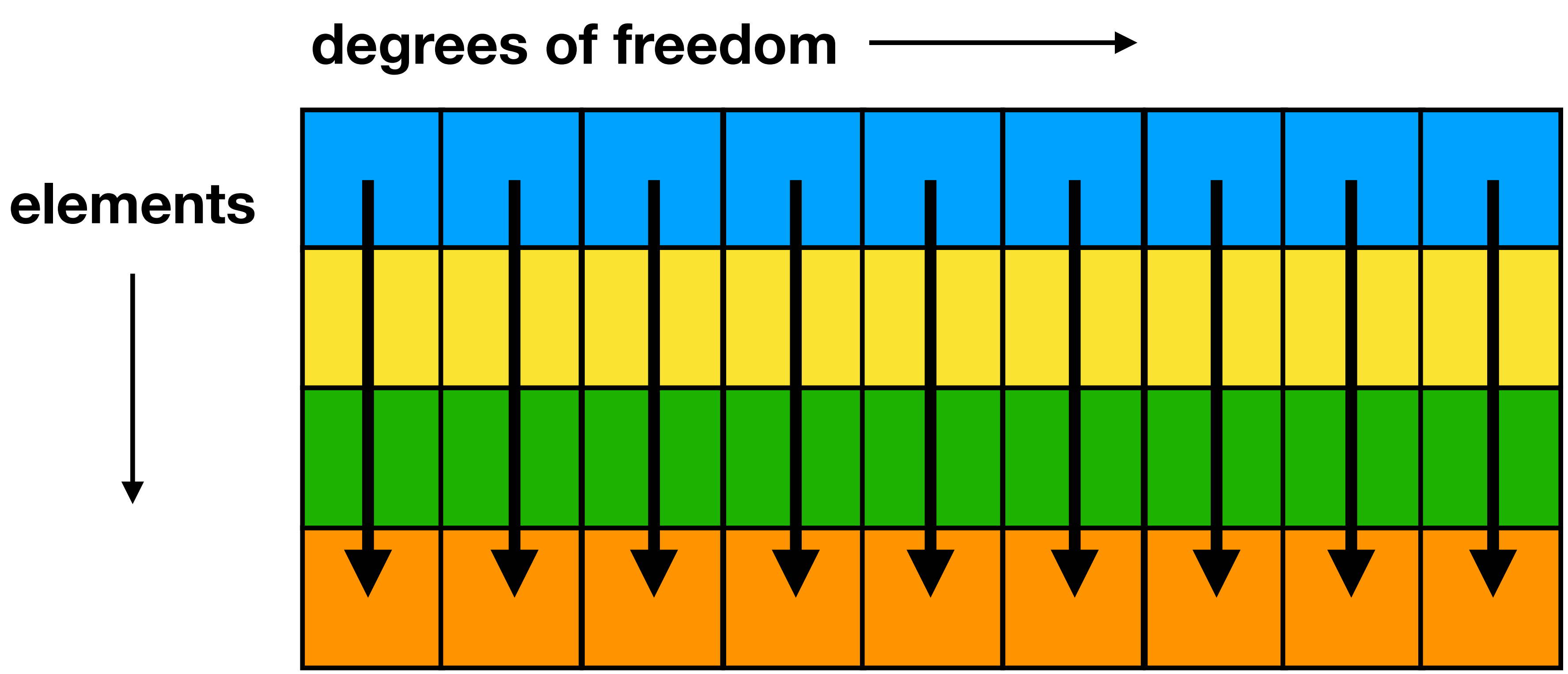
Data layout

Natural to consider data laid out element by element



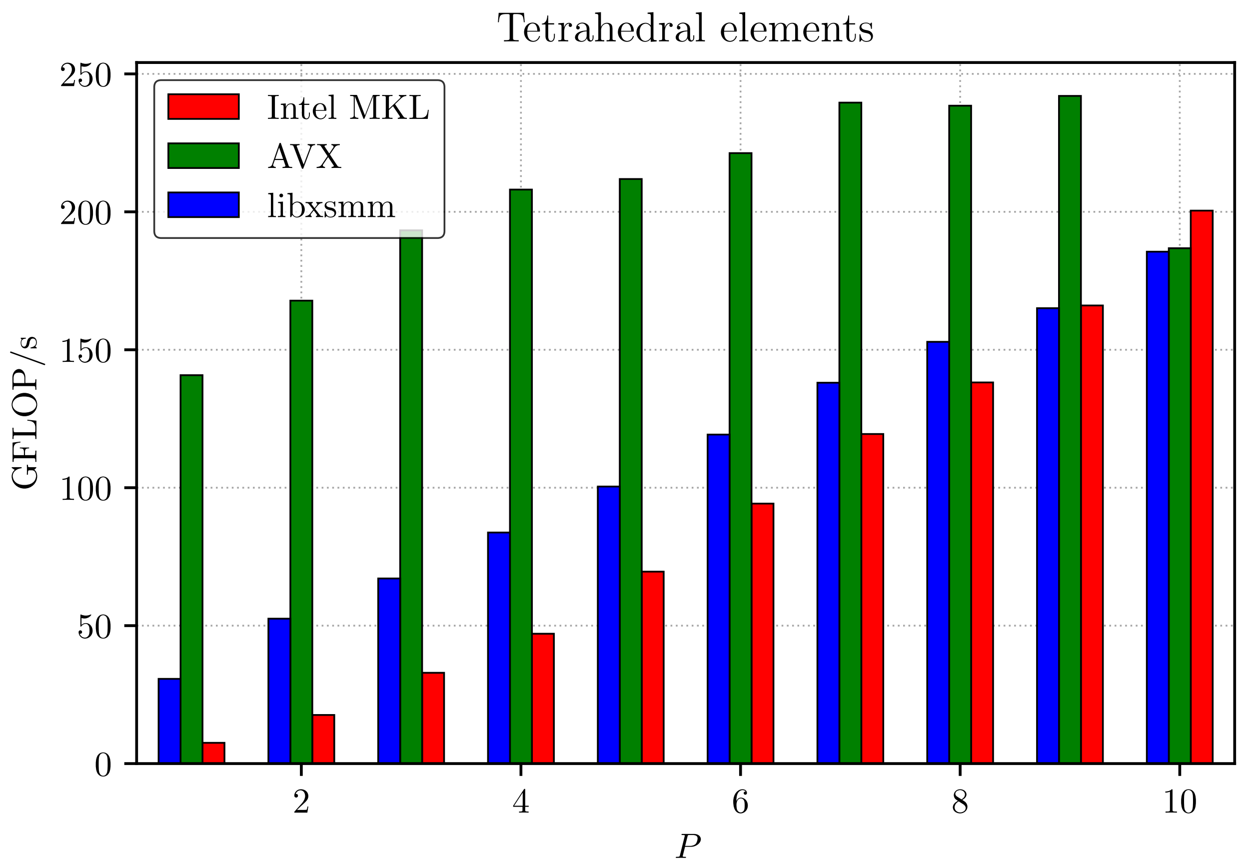
Data layout

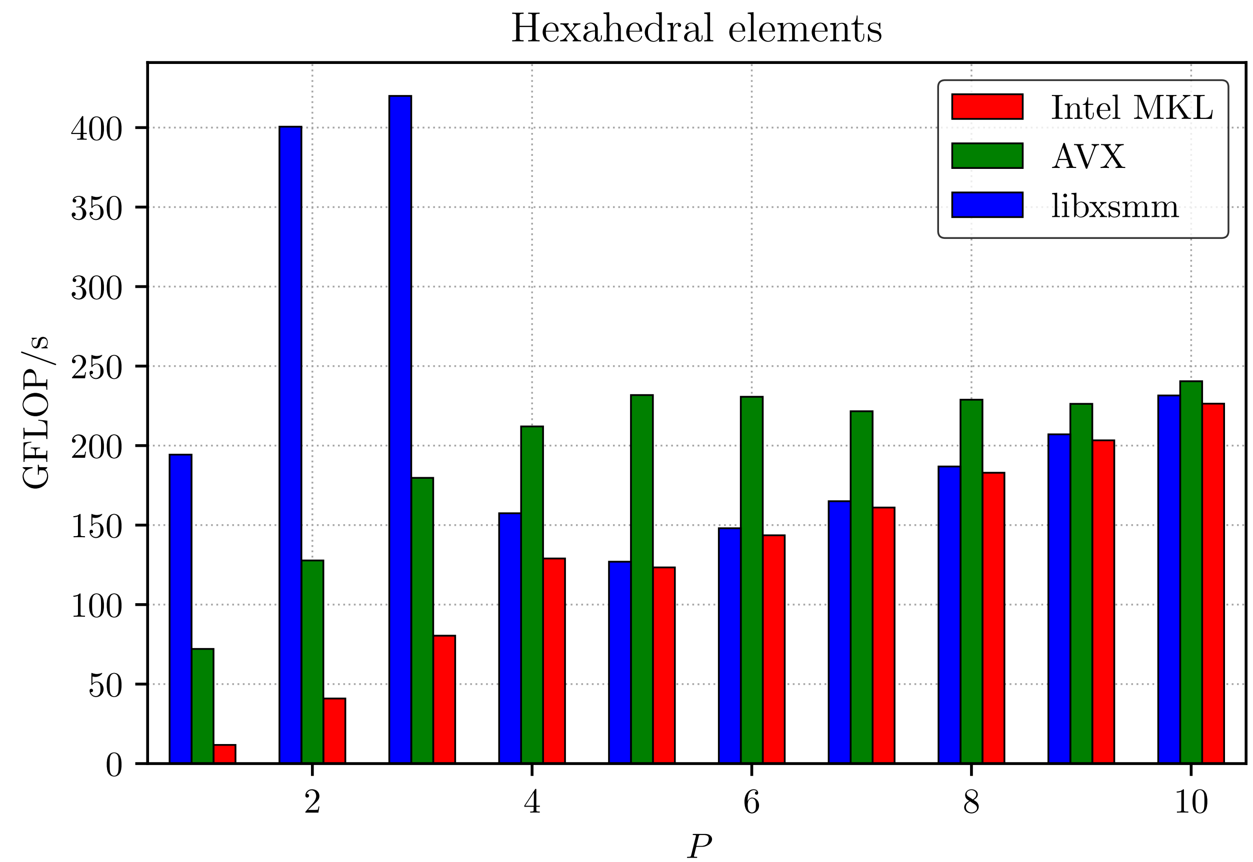
Instead, could consider blocks that are equivalent to processor vector width (e.g. AVX = 4 doubles)



Exploiting vectorisation

- Perform multiple operations in a single cycle
- Combine with:
  - Hand-written loops for sum-factorisation
  - Explicit intrinsics for vector operations
  - C++ templating to allow for loop unrolling





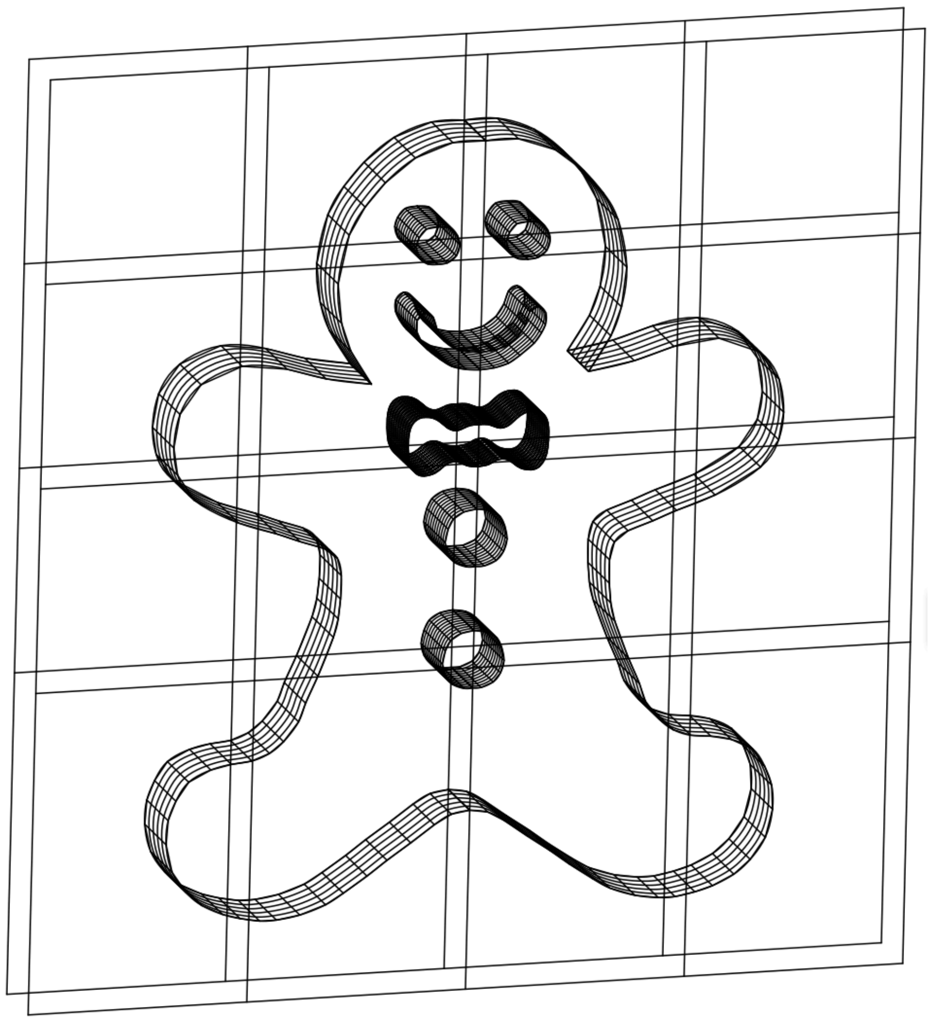
Significant performance gains against benchmark solutions (libxsmm, Intel MKL) for backward transform

Still ongoing work!

Outline

- Spectral/hp element Modelling
  - Overview of Nektar++
- Challenges for modelling
  - Stabilisation and robustness
  - Performance
  - Mesh generation
- Applications
  - Incompressible Flow Modelling
  - Hasegawa-Wakatani equations

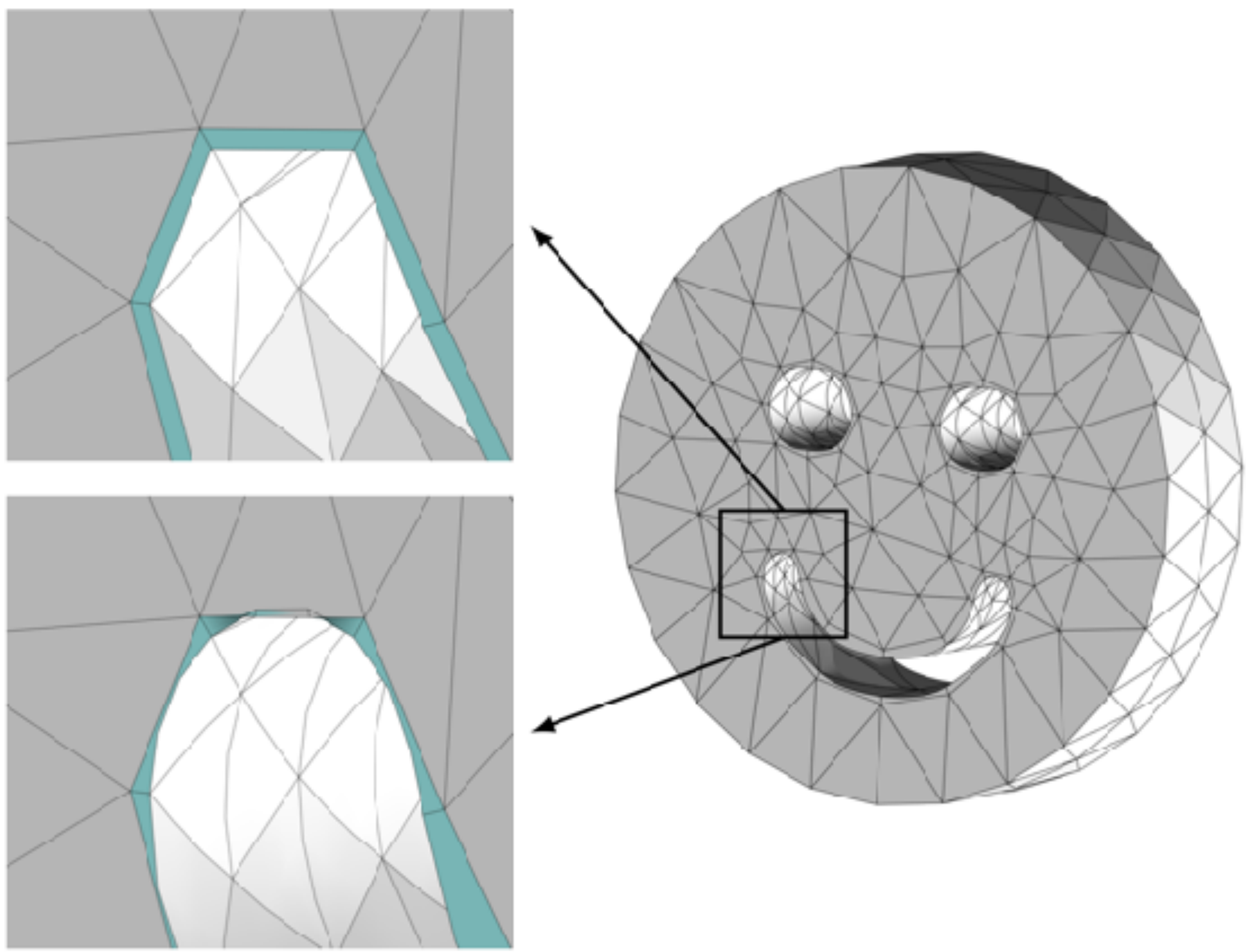
High-order mesh



B-Rep

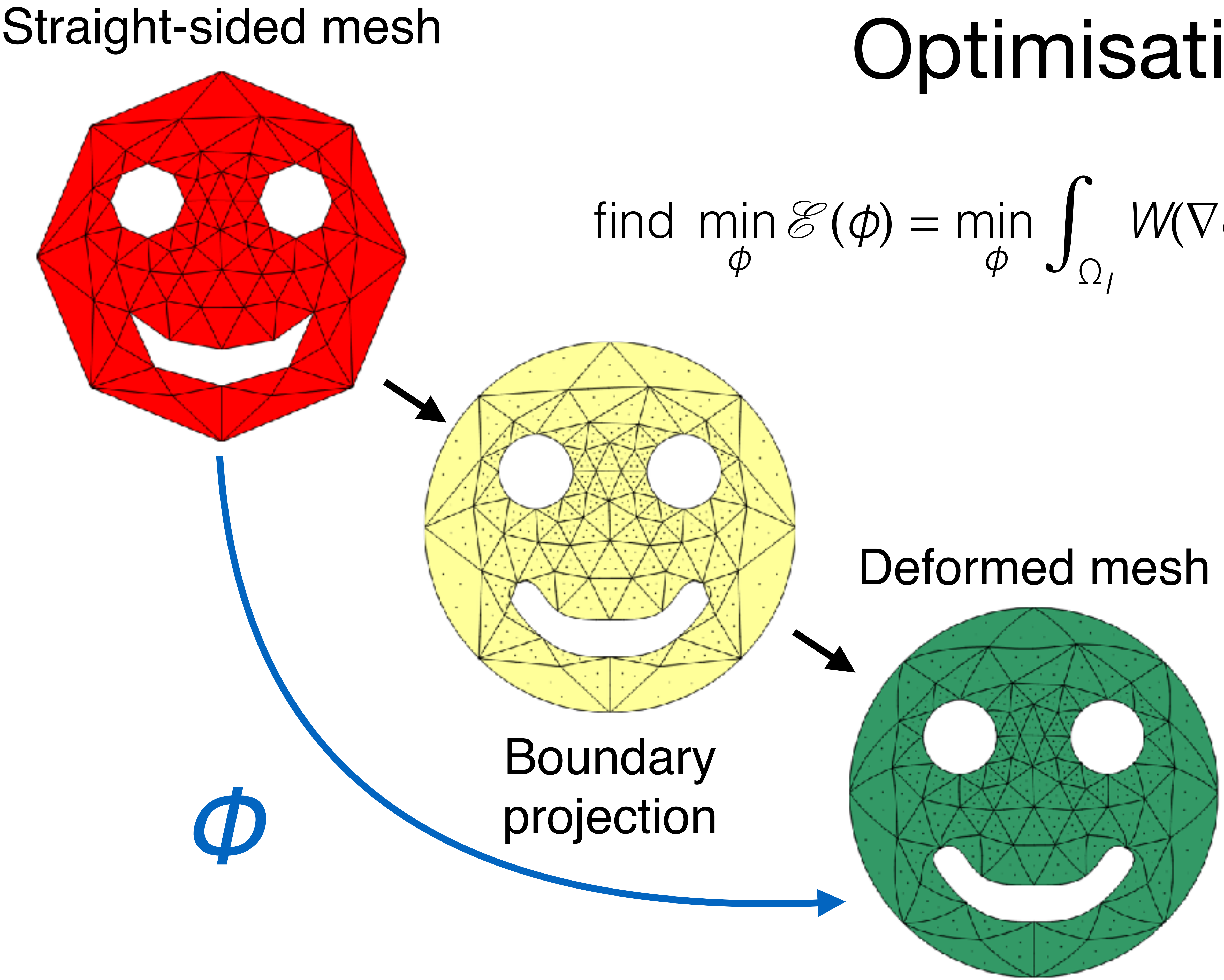
High-order mesh generation

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



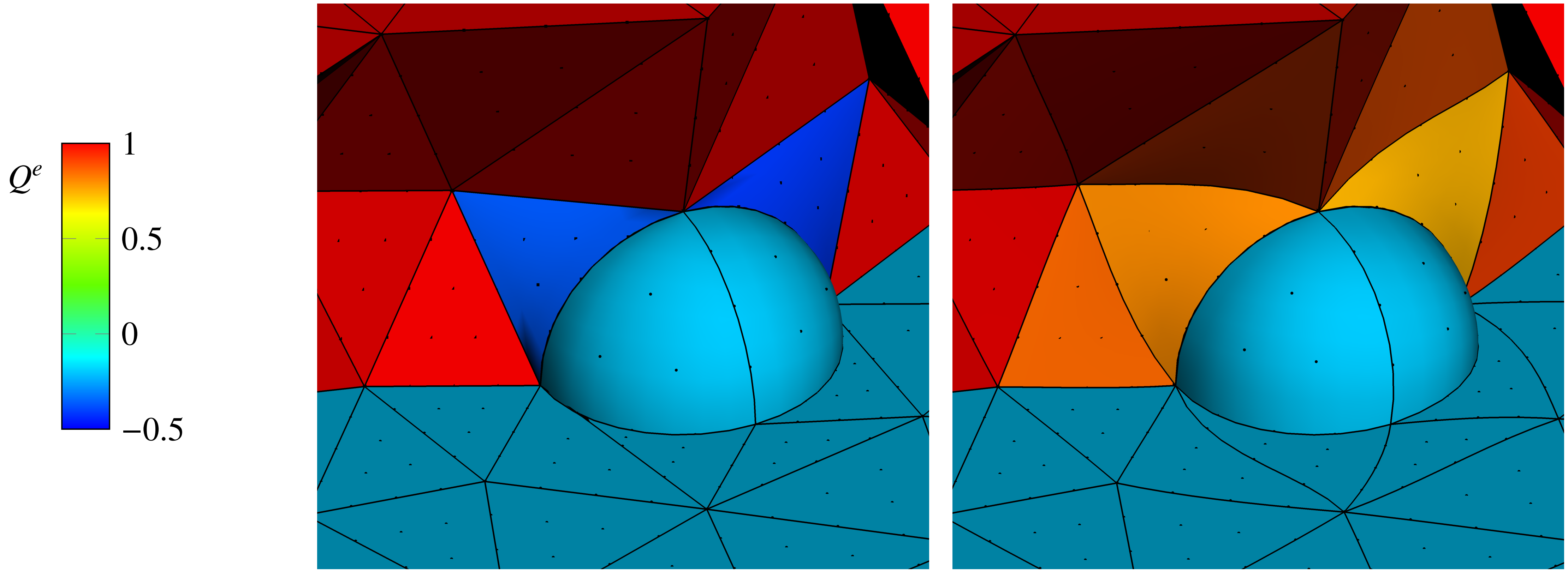
Optimisation

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



Surface mesh sliding

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



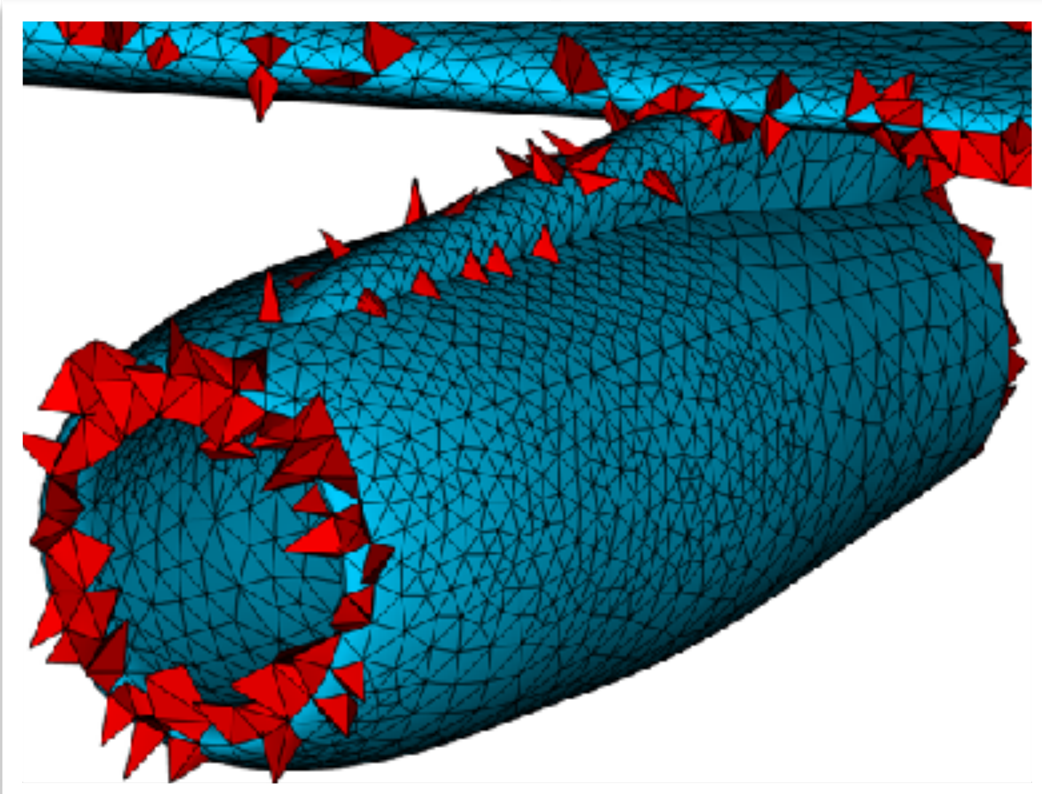
Imperial College London

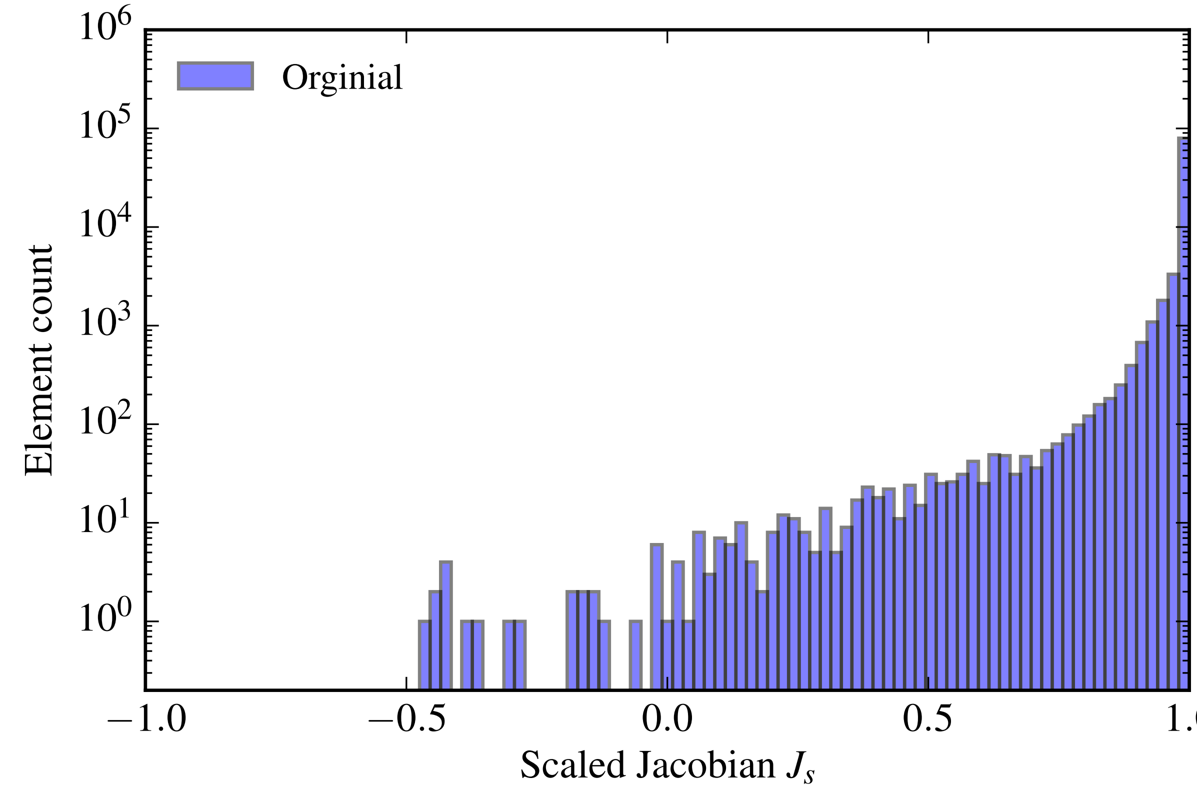
UNIVERSITY OF EXETER

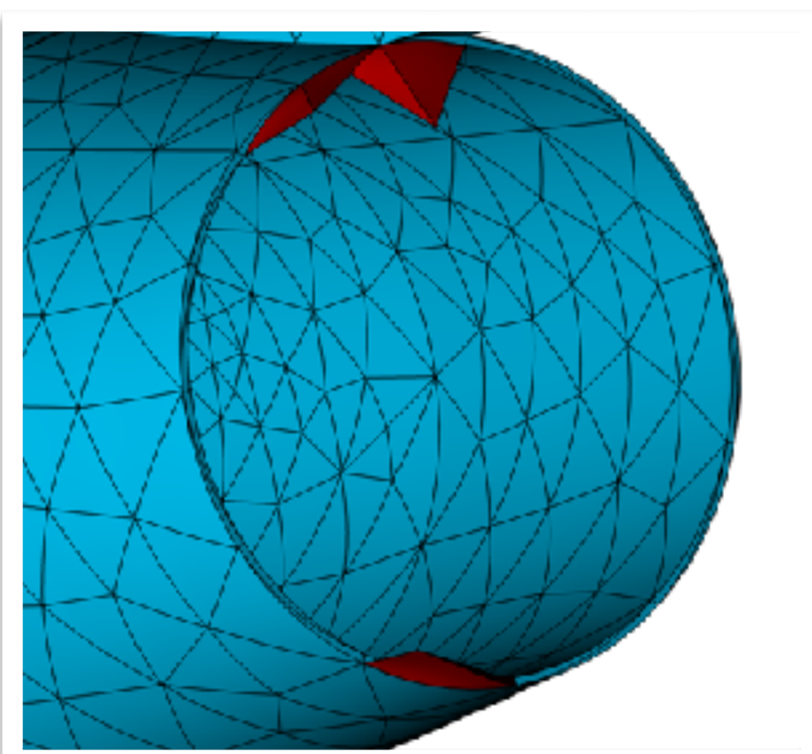
Imperial College London

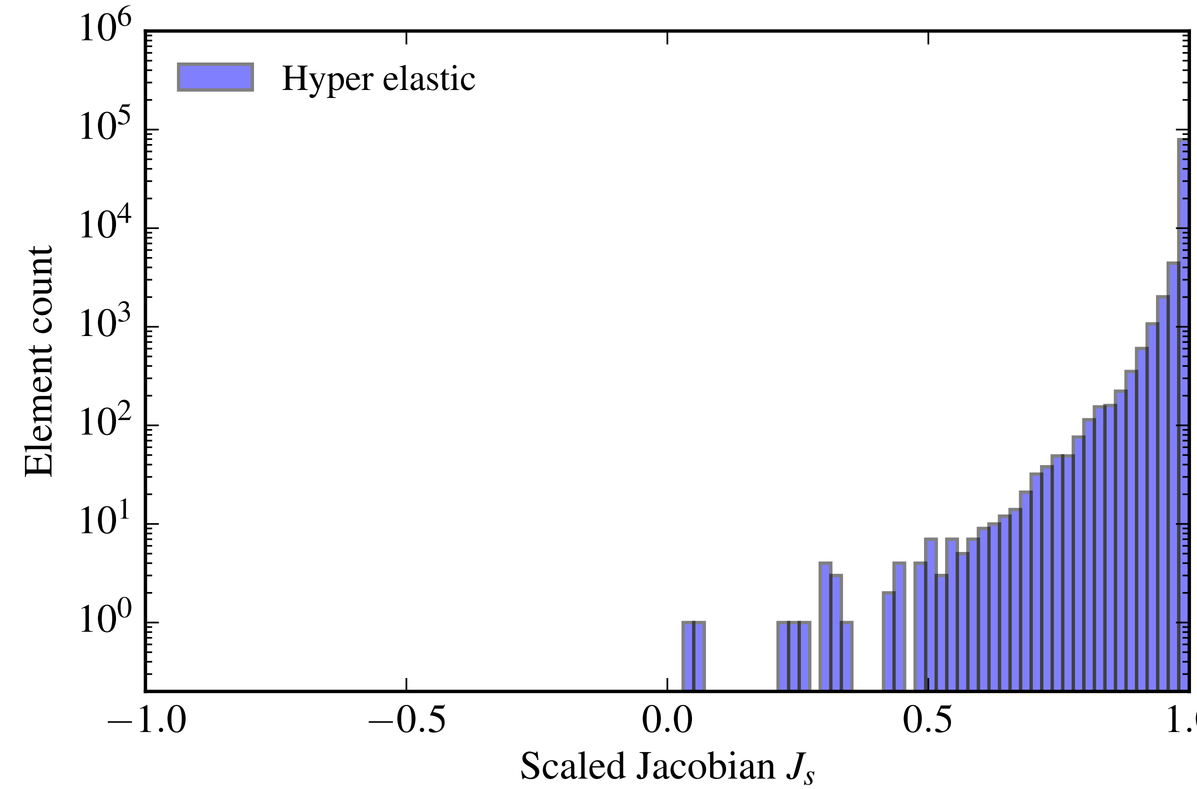
DLR F6 engine

UNIVERSITY OF EXETER









Outline

- Spectral/hp element Modelling
  - Overview of Nektar++
- Challenges for modelling
  - Stabilisation and robustness
  - Performance
  - Mesh generation
- Applications
  - Incompressible Flow Modelling
  - Hasegawa-Wakatani equations

Hasegawa-Wakatani

- Model for drift wave instability in fusion edge turbulence simulations
- **State variables:** ϕ is the electrostatic potential, ζ is vorticity, n is the perturbed number density
- **Parameters:** α is the adiabaticity operator, κ is a constant density gradient scale-length
- Determine how to implement these equations

Imperial College London

UNIVERSITY OF EXETER

Hasegawa-Wakatani equations

$$\frac{\partial \zeta}{\partial t} + \{\phi, \zeta\} = \alpha(\phi - n) \boxed{- \mu\nabla^4 \zeta}$$

$$\frac{\partial n}{\partial t} + \{\phi, n\} = \alpha(\phi - n) - \kappa\frac{\partial \phi}{\partial y} \boxed{- \mu\nabla^4 n}$$

$$\nabla^2 \phi = \zeta$$

$\{a,b\}$ is the canonical Poisson bracket operator

$$\{a, b\} = \frac{\partial a}{\partial x}\frac{\partial b}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial b}{\partial x}$$

Hasegawa-Wakatani equations

- Mixture of convective term and elliptic solve for ζ from ϕ
- Therefore can leverage **mixed** formulation: CG for elliptic solve, DG for convection
- Since we're using DG, omit the hyper diffusion term for stability
- Rewrite in conservative form
- Leverage Nektar++ framework for rapid development

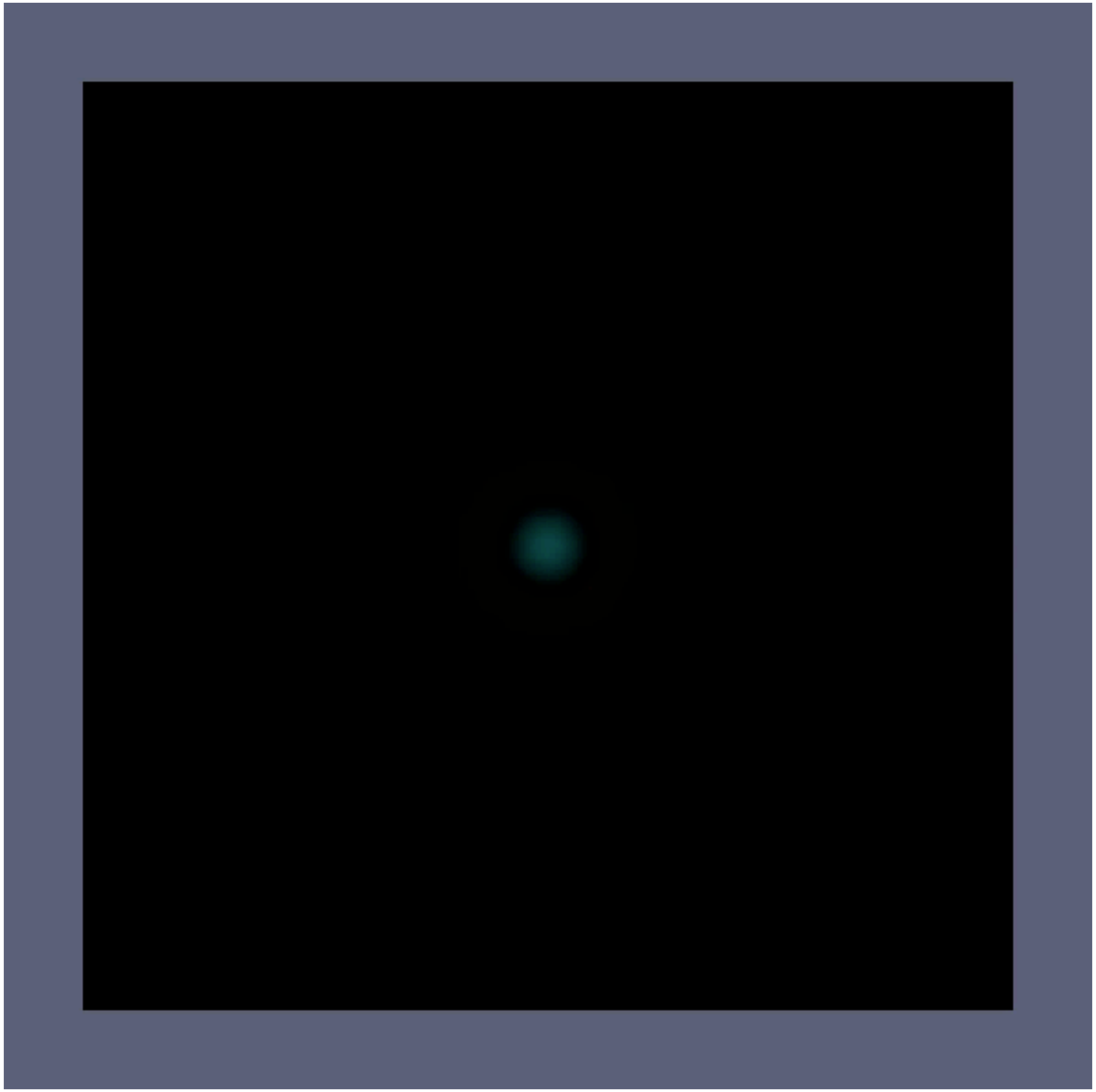
Hasegawa-Wakatani equations

$$\mathbf{u}_E = \left(-\frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial x}\right)$$

$$\{\phi, \zeta\} = \nabla\cdot(\mathbf{u}_E\zeta)$$
$$\{n, \zeta\} = \nabla\cdot(\mathbf{u}_E n)$$

Apply DG formulation with $\mathbf{u}_E$ and upwinding

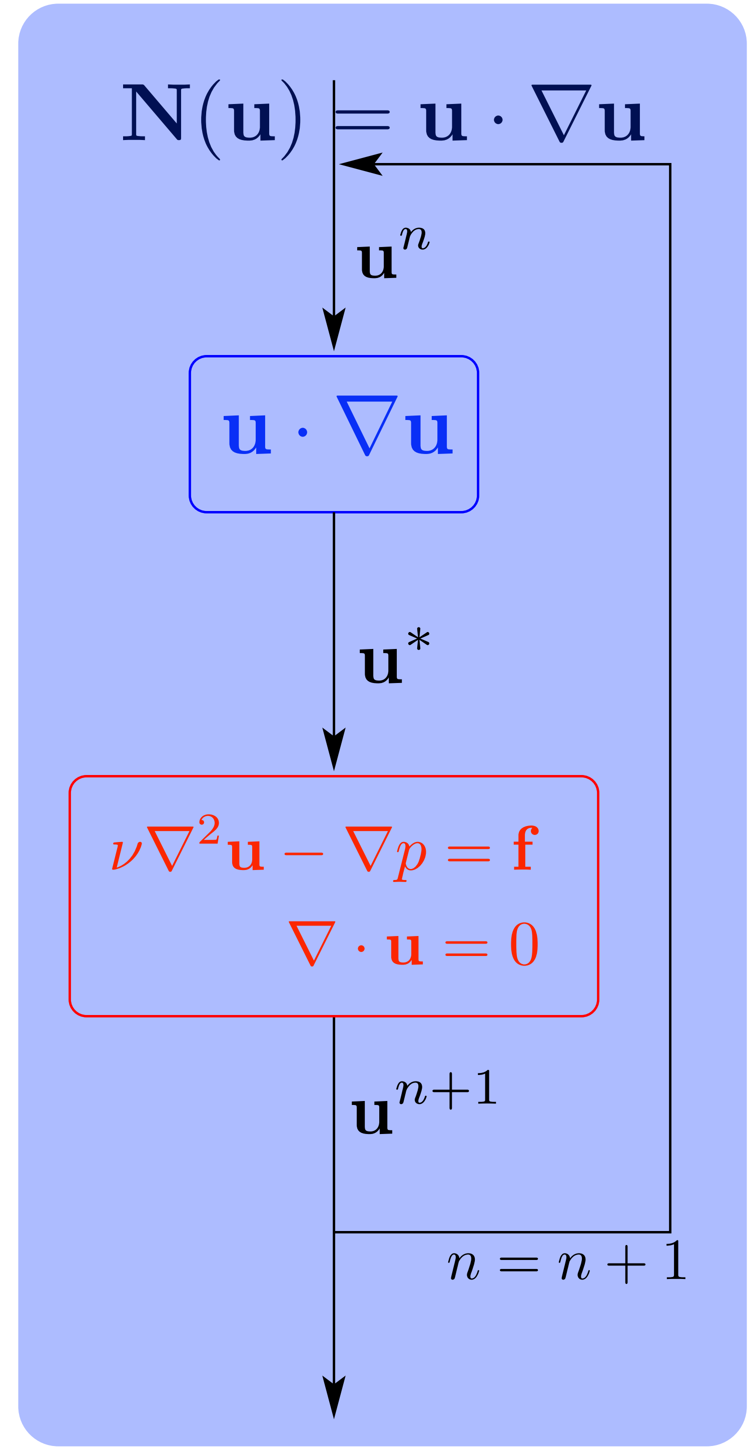
Sample output



- spectral/*hp* simulation of the Hasegawa-Wakatani equations
- Joint CG/DG implementation
- Simple model for drift-wave turbulence

Navier-Stokes Time-stepping

Navier–Stokes:
$$\boxed{\boxed{\partial_t \mathbf{u} + \mathbf{N}(\mathbf{u})} = -\nabla p + \nu \nabla^2 \mathbf{u} \\ \nabla \cdot \mathbf{u} = 0}$$



Velocity correction scheme *(aka stiffly stable):*

Orszag, Israeli, Deville (90), Karnaidakis Israeli, Orszag (1991), Guermond & Shen (2003)

Advection: $u^* = -\sum_{q=1}^{J} \alpha_q \mathbf{u}^{n-q} - \Delta t \sum_{q=0}^{J-1} \beta_q \mathbf{N}(\mathbf{u}^{n-q})$

Pressure Poisson:
$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*$$

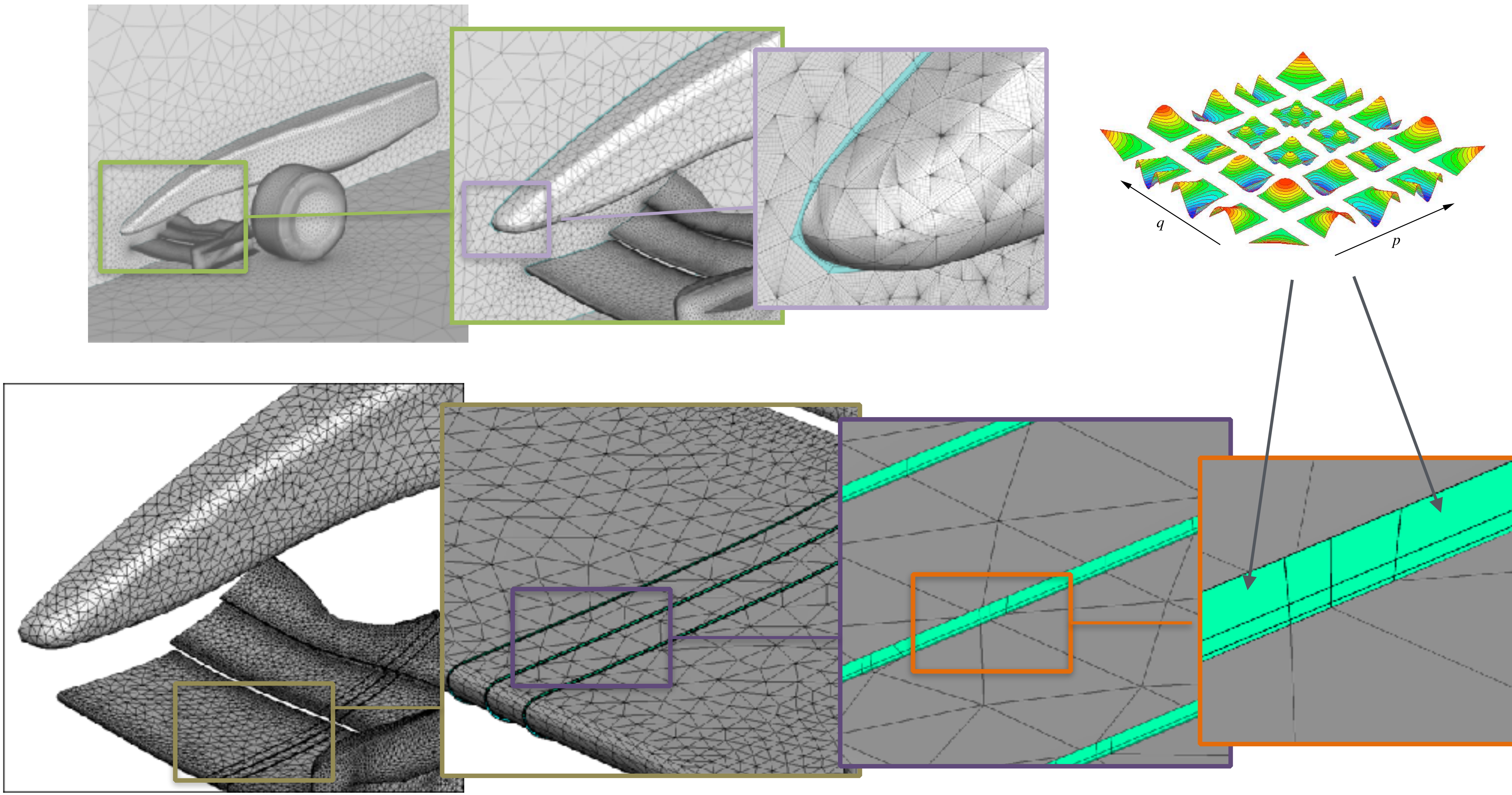
Helmholtz:
$$\nabla^2 \mathbf{u}^{n+1} - \frac{\alpha_0}{\nu \Delta t} \mathbf{u}^{n+1} = -\frac{\mathbf{u}^*}{\nu \Delta t} + \frac{1}{\nu} \nabla p^{n+1}$$

Imperial College
London

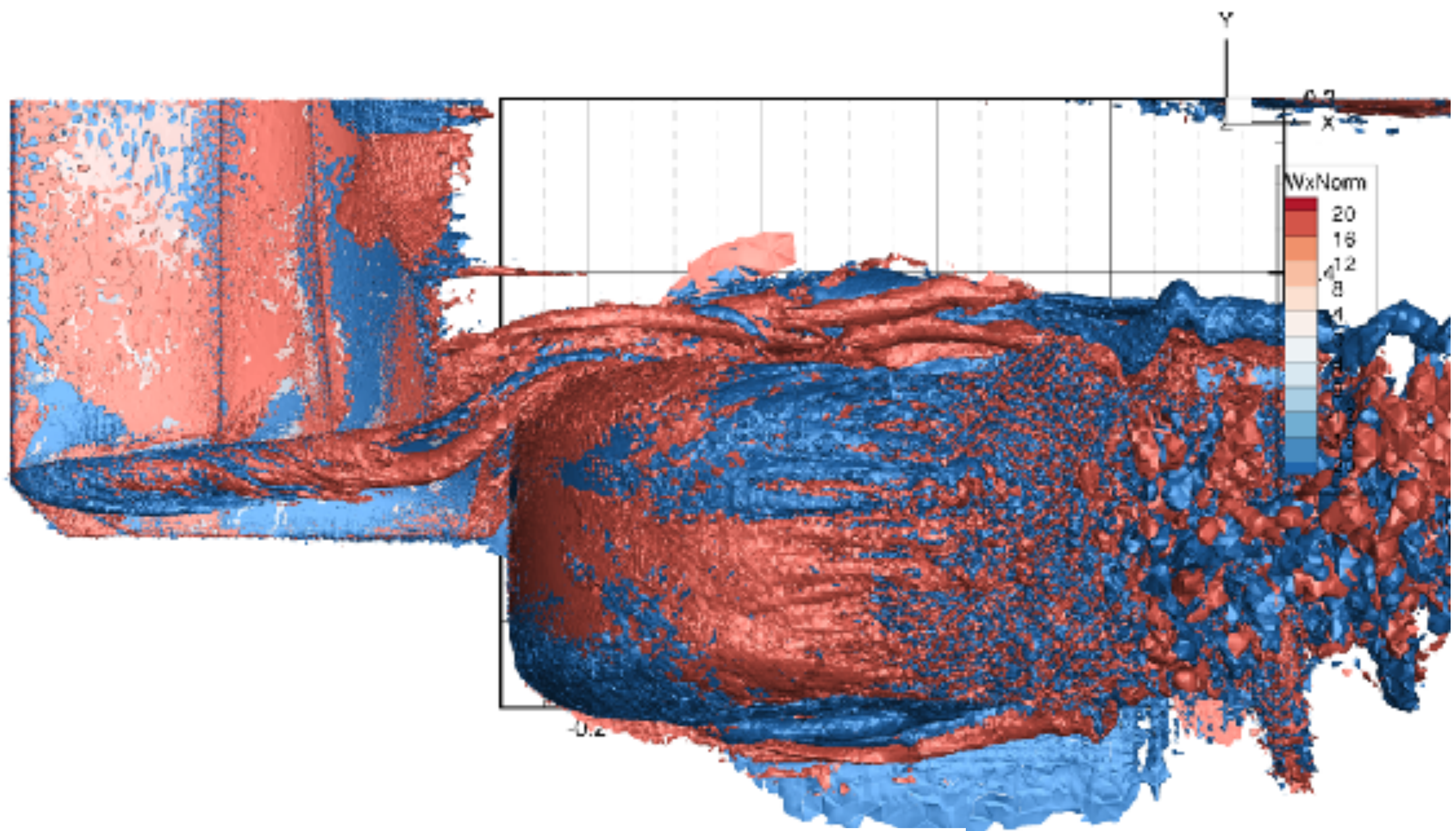
McLaren

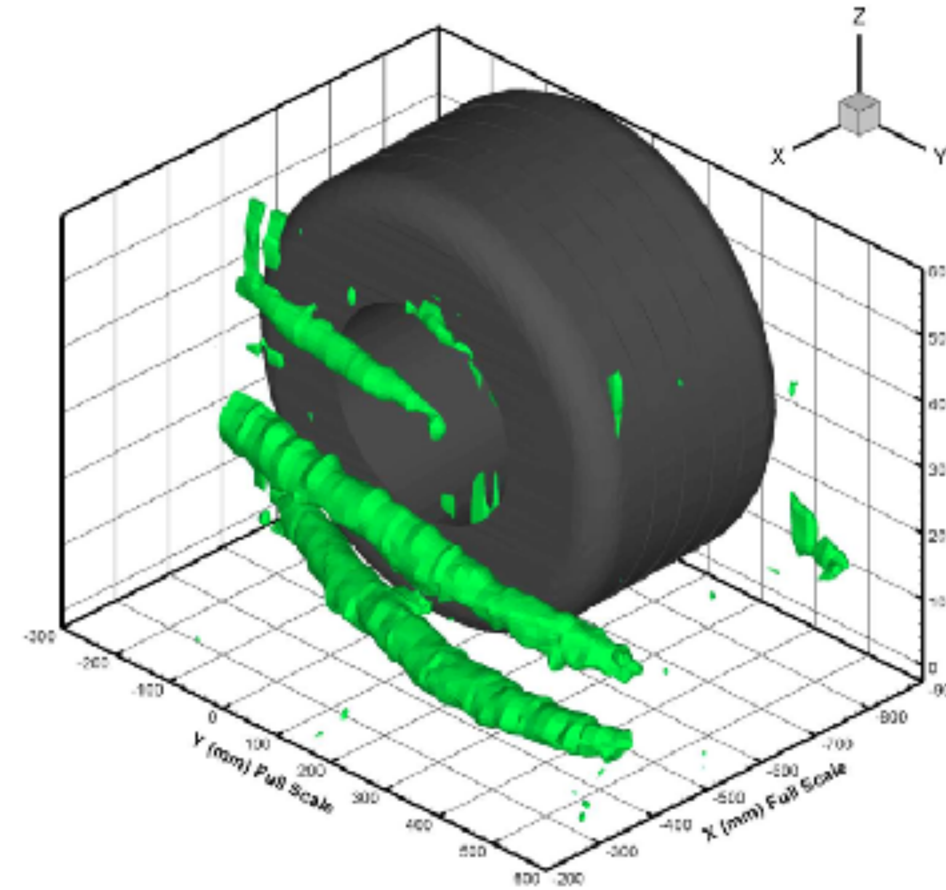
Meshing for F1 applications

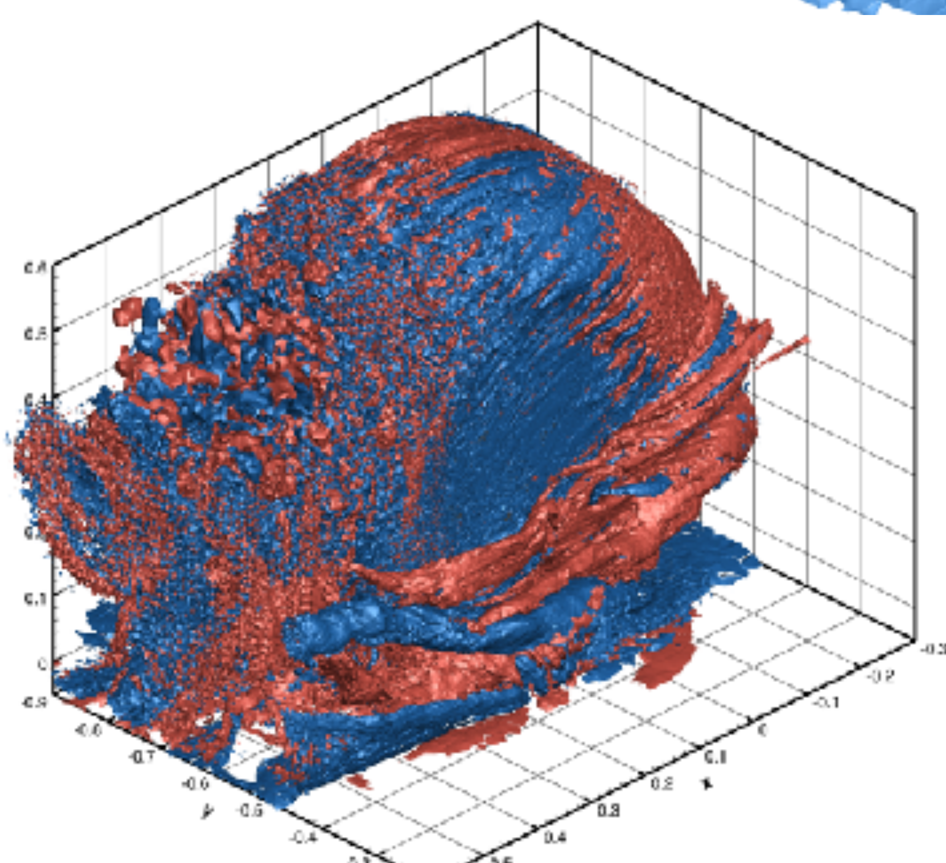
Meshing

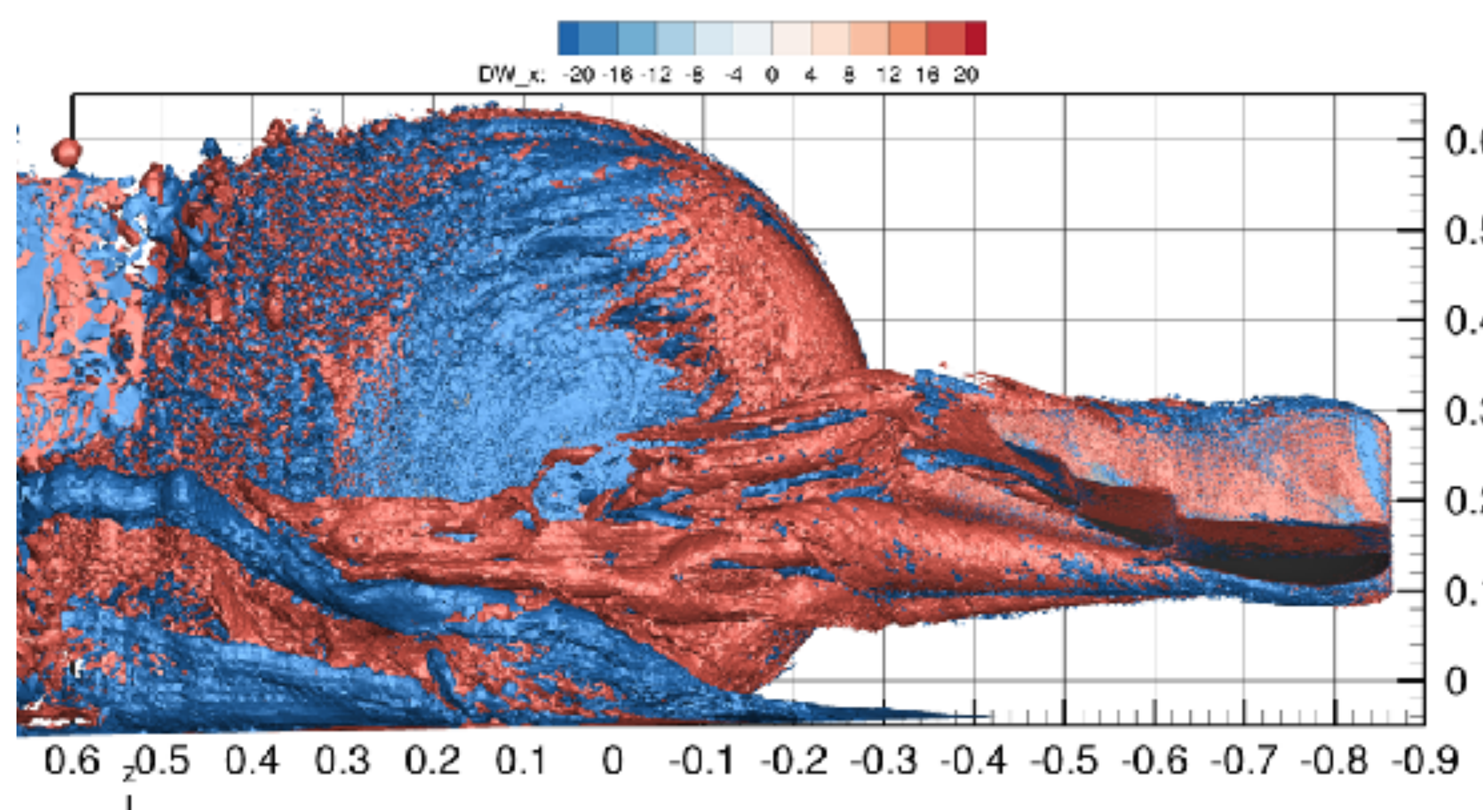


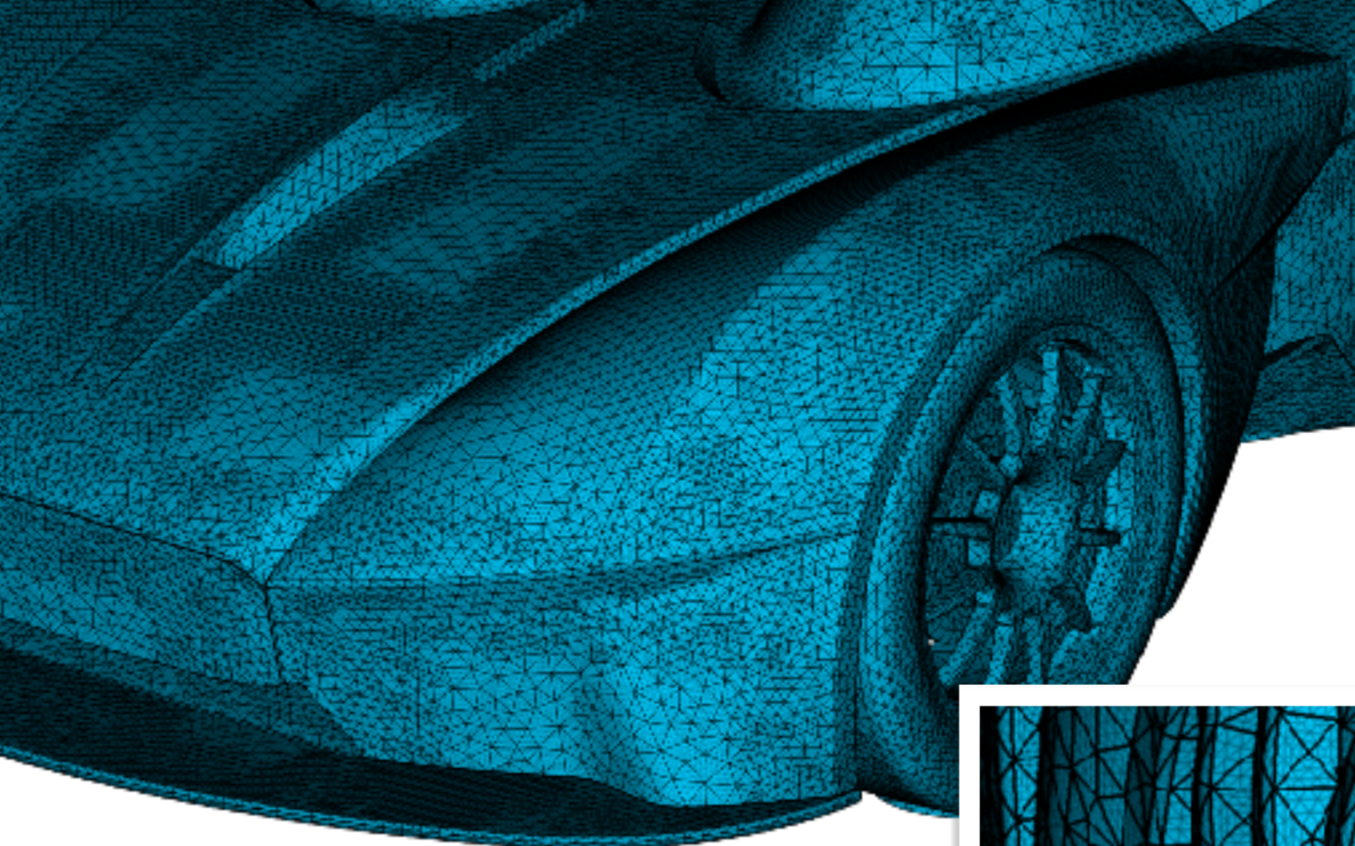
F1 Geometry

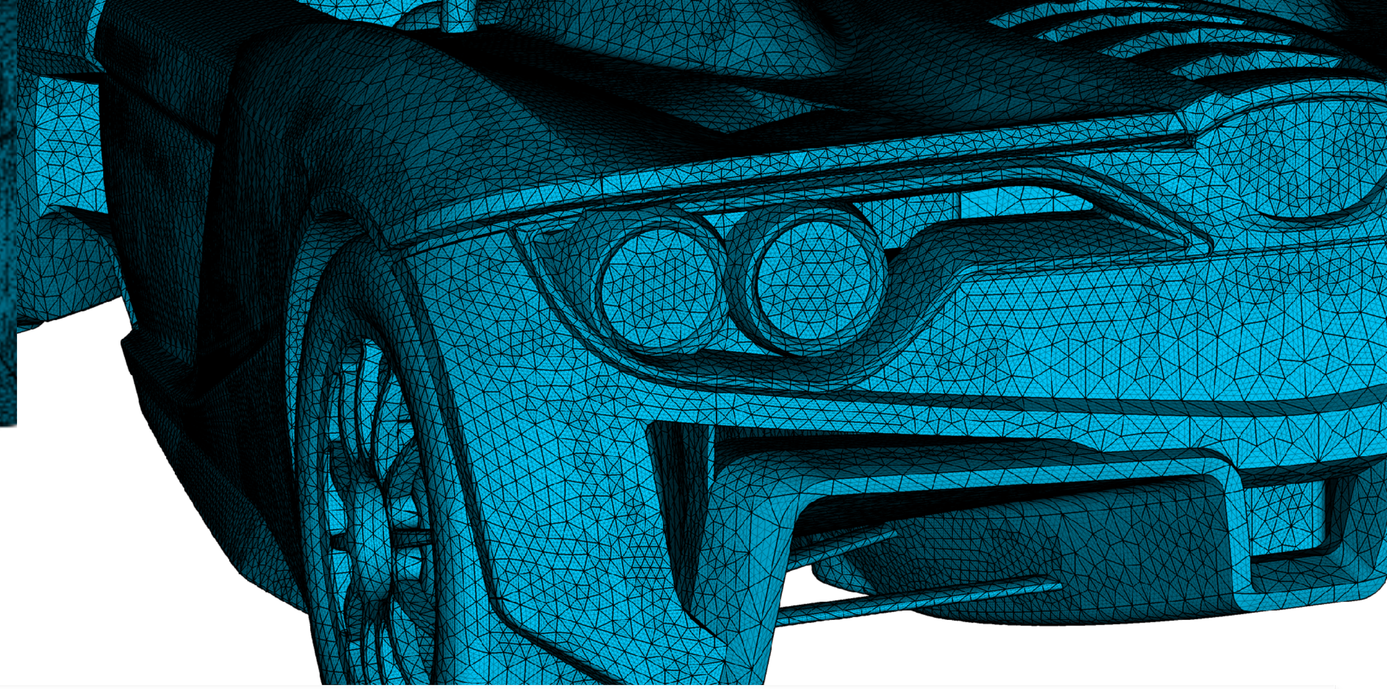

WxNorm
20
16
12
8


Y (mm) Full Scale
X (mm) Full Scale
Z (mm) Full Scale



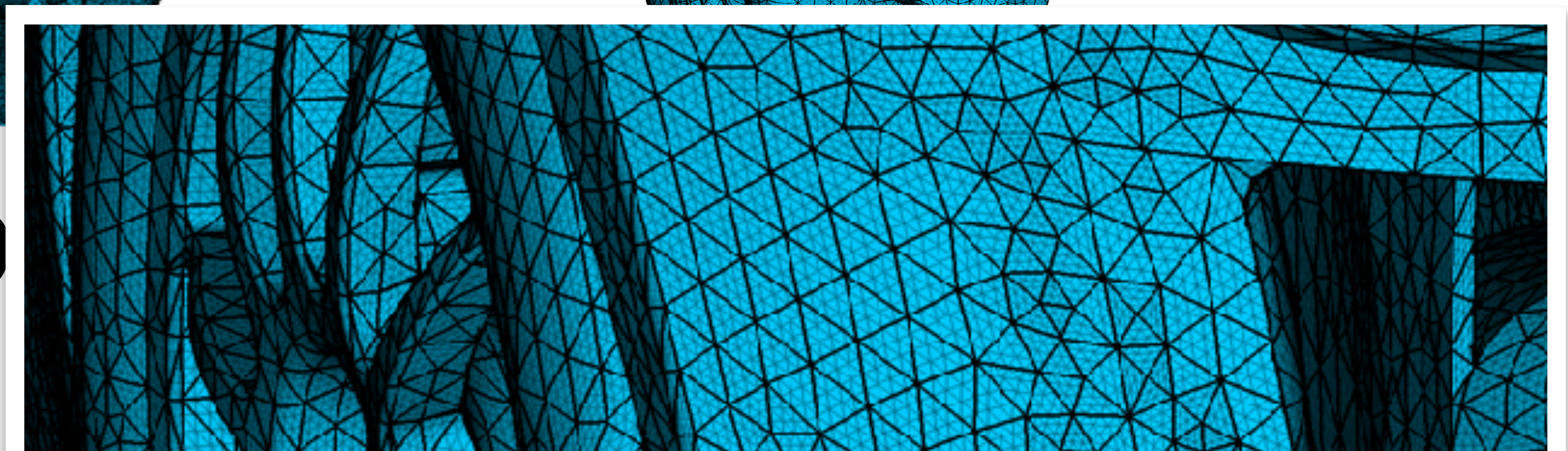

DW_x: -20 -16 -12 -8 -4 0 4 8 12 16 20
0.6
0.5
0.4
0.3
0.2
0.1
0
0.6 0.5 0.4 0.3 0.2 0.1 0 -0.1 -0.2 -0.3 -0.4 -0.5 -0.6 -0.7 -0.8 -0.9

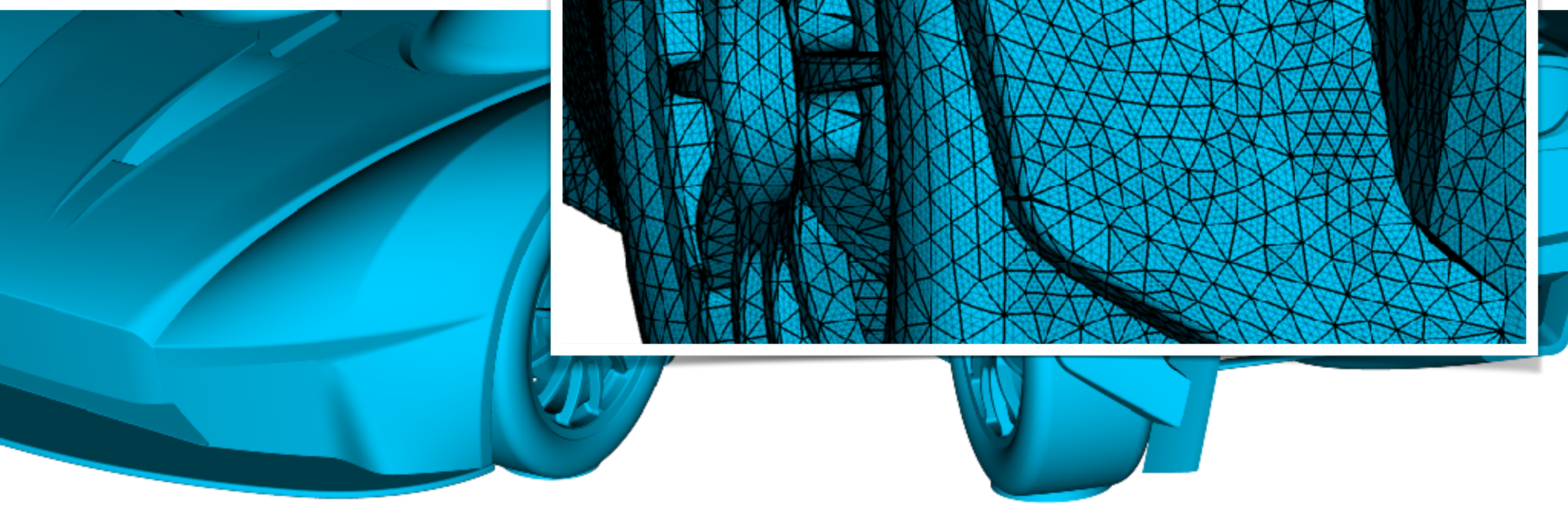




Elemental ro

$P = 4$

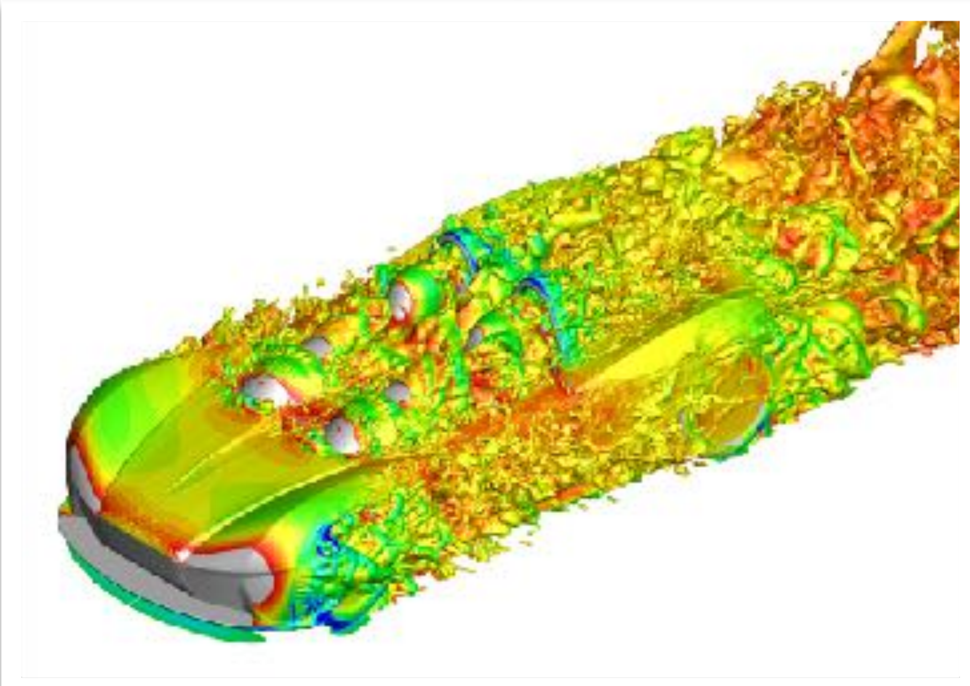




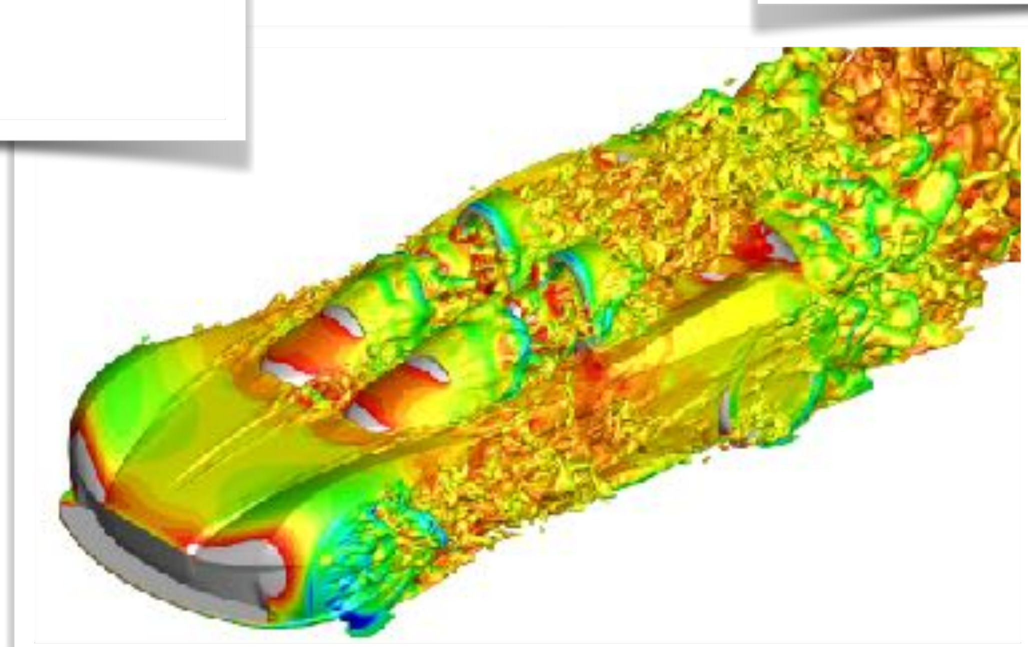
Imperial College London

Elemental Road Race Car

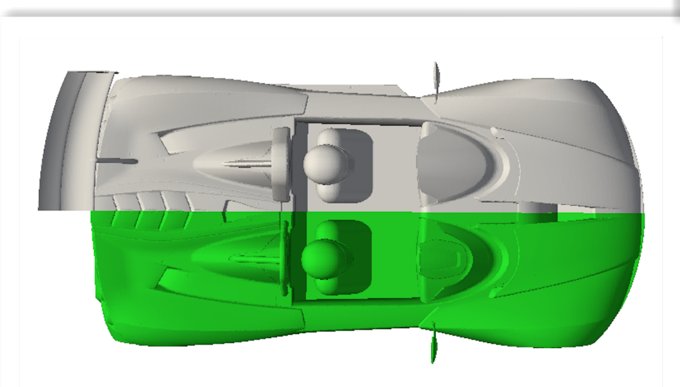
UNIVERSITY OF EXETER

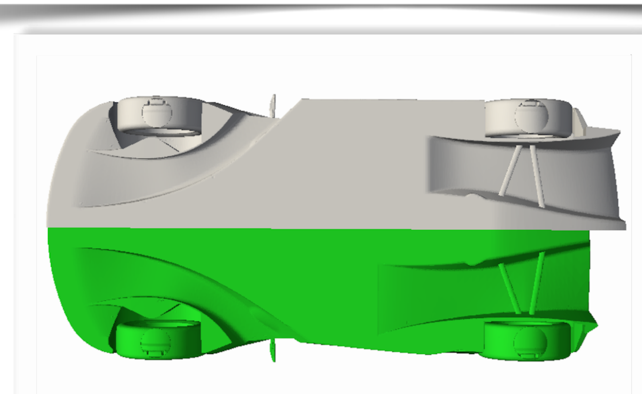


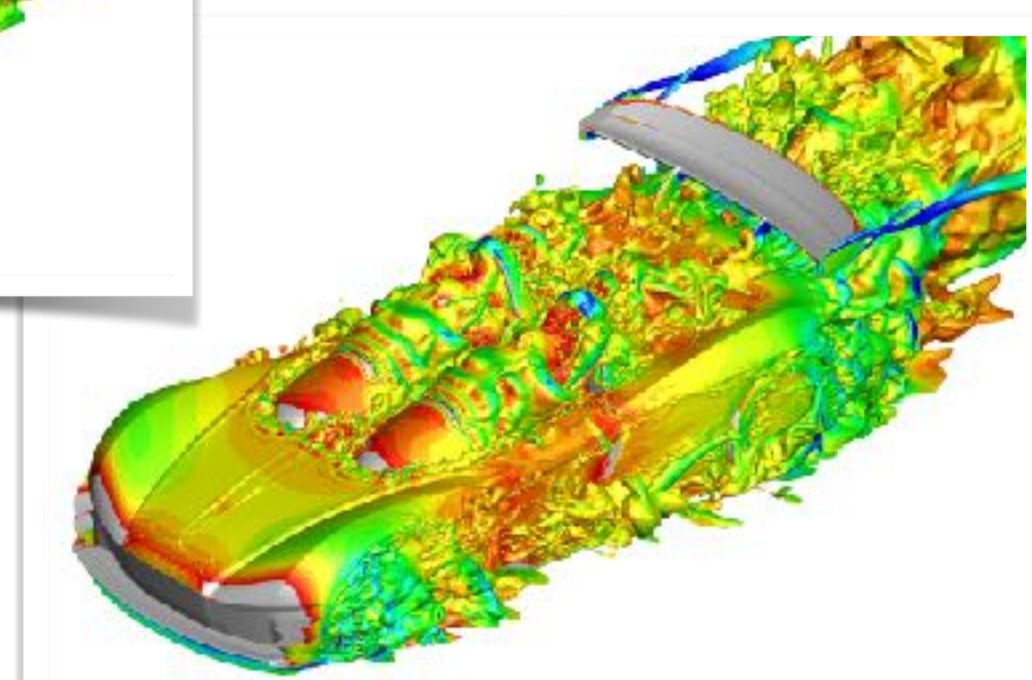
5th order
Re=1m



Design 2: +33% Downforce



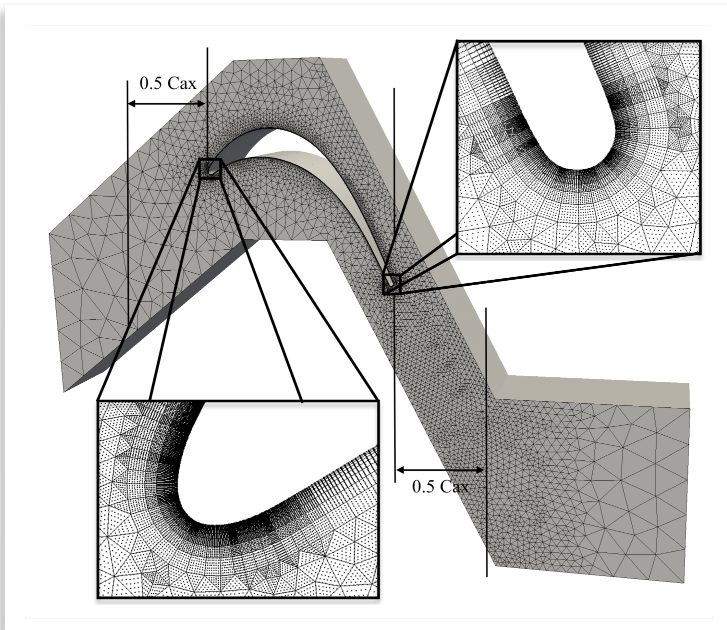




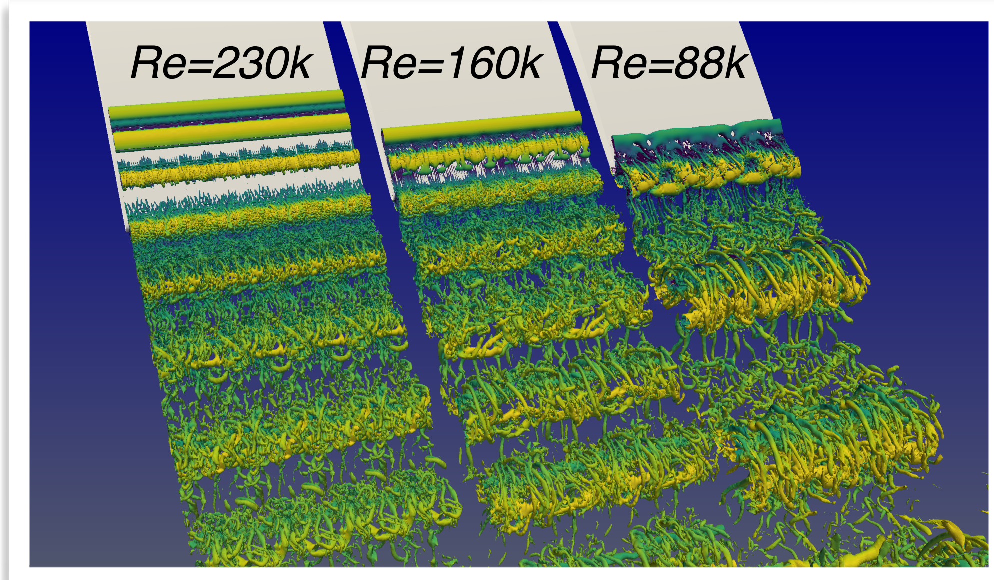
Moxey, Turner, Jassim, Taylor Sherwin

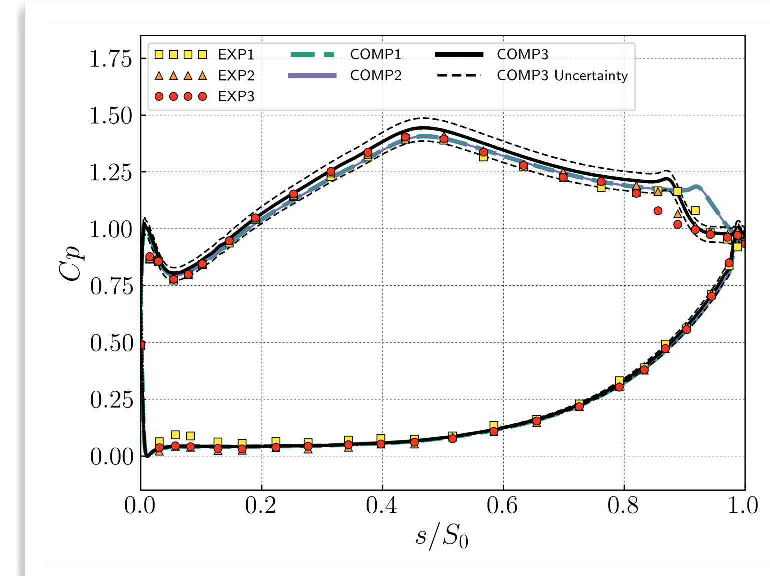
Design 3: +270% Downforce

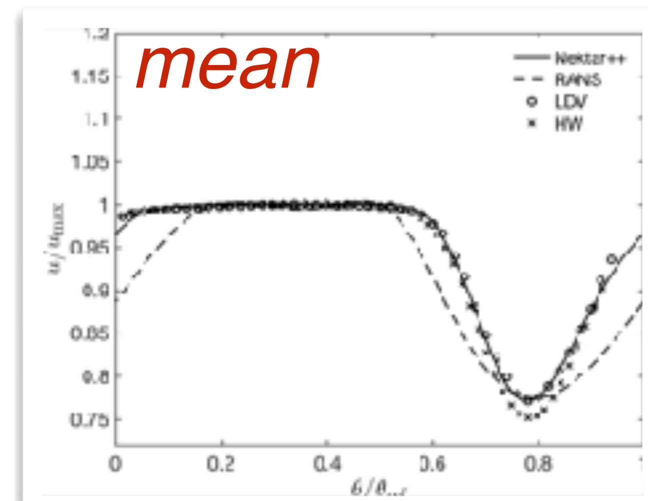
Turbomachinery application

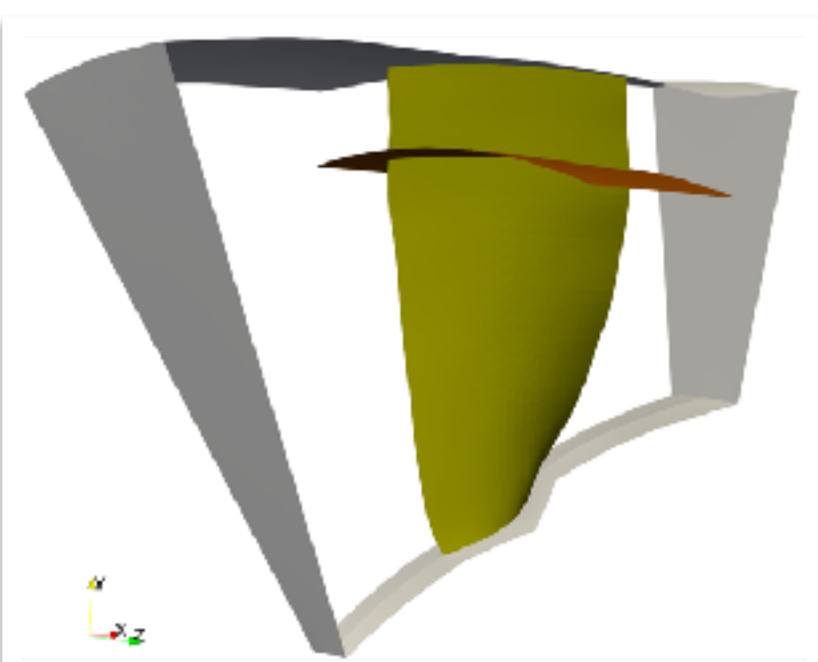


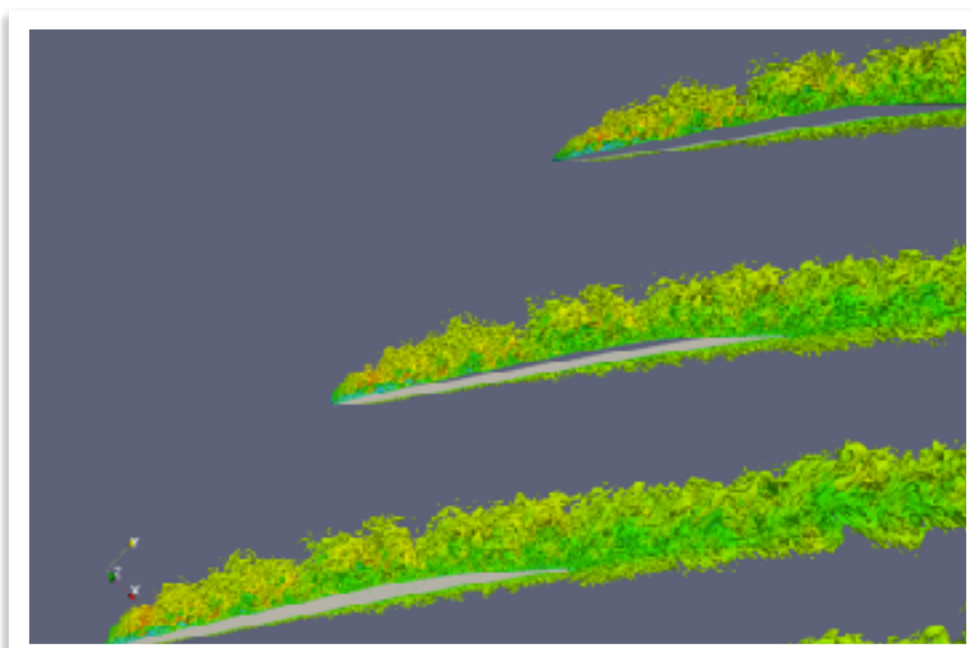
T106 using NekMesh



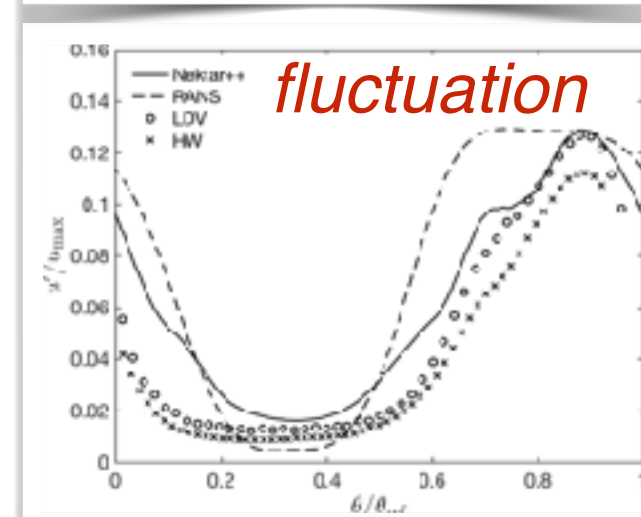








Re=1.5M



Imperial College London

UNIVERSITY OF EXETER

Summary/Confessions

- Applying spectral/hp element techniques to challenging industrial flow problems.
- Compactness of methods attractive on current/ emerging HPC architecture
- Accurate, transient flow modelling is an enabling technology for high-end engineering/physics.
- But …
  - Meshing for 3D geometries is a specialist skill
  - Robustness has required careful analysis and probably requires more!

ROYAL ACADEMY OF ENGINEERING

NEKTAR++ SPECTRAL/HP ELEMENT FRAMEWORK

EPSRC