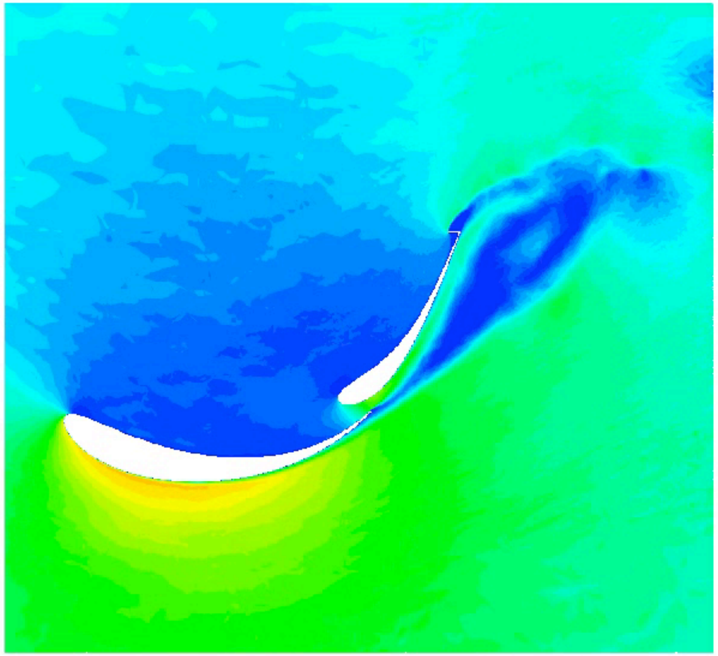
# Nektar++: A Progress Report

Spencer Sherwin, Chris Cantwell,
Dave Moxey, Mike Kirby
*Department of Aeronautics, Imperial College London*
*SCI Institute, University of Utah*



Dynamic pressure Re=100 000

Imperial College London

THE UNIVERSITY OF UTAH

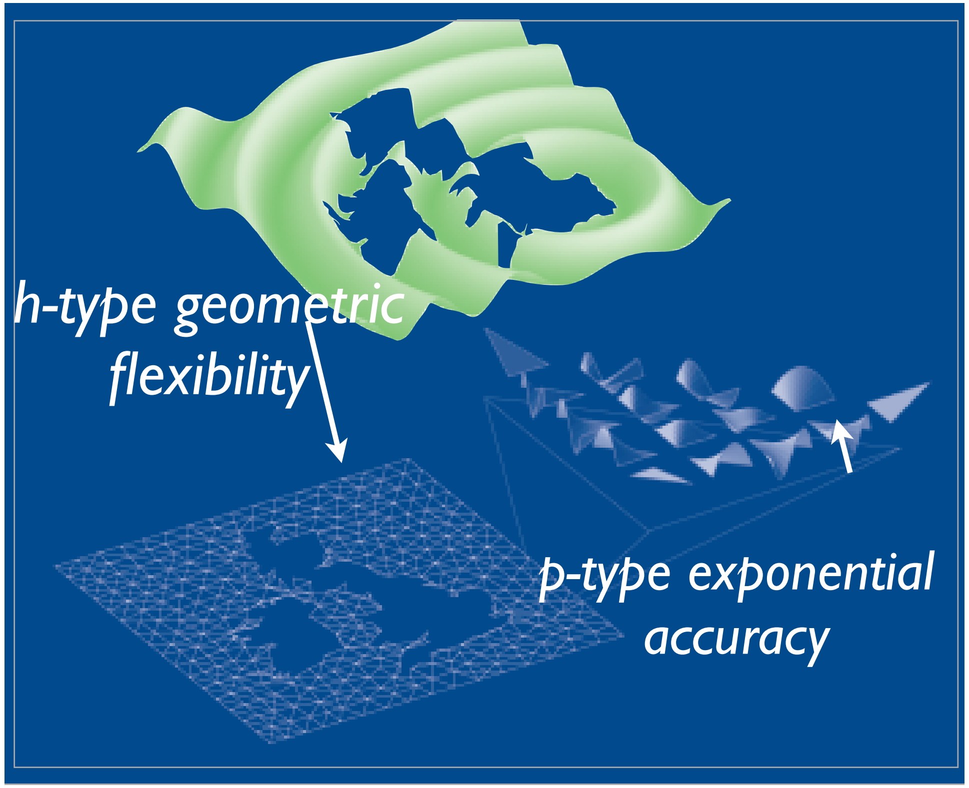
# Outline

- What are we doing?
- Optimizing our implementations
  - Variable p
  - Cloud computing

Imperial College London

# Nektar++: An h to p finite element framework

*Provide an unified interface to an open environment which blends high- and low-order finite element methods.*




Imperial College London

NEKTAR++ SPECTRAL/HP ELEMENT FRAMEWORK

# Nektar++: www.nektar.info

Helmholtz problem: $$\nabla^2 u + \lambda u = f$$

$$\downarrow$$

Weak form + IBP: $$-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$$

# Nektar++: www.nektar.info

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

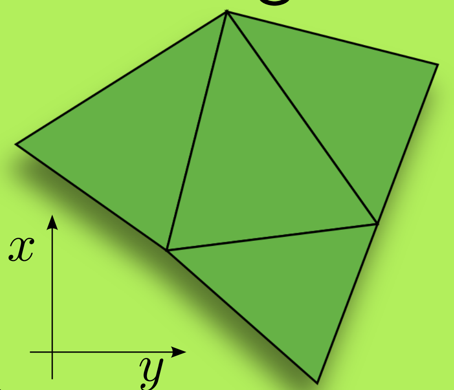
Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem: $\nabla^2 u + \lambda u = f$

$$\downarrow$$

Weak form + IBP: $-(\nabla u, \nabla v) + \lambda(u,v) + (\nabla u, v)|_{d\Omega} = (f,v)$

MultiRegions




$$\mathbf{f}[i] = \int_\Omega \Phi_i(x) f(x)\, dx$$

# Nektar++: www.nektar.info

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$

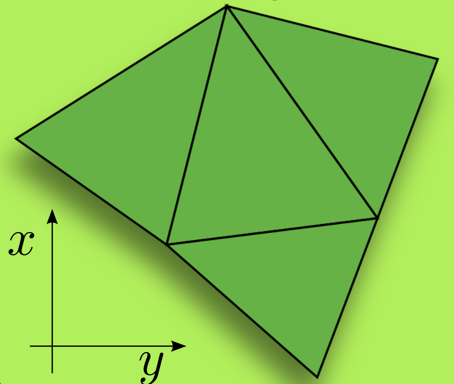
Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem: $\nabla^2 u + \lambda u = f$

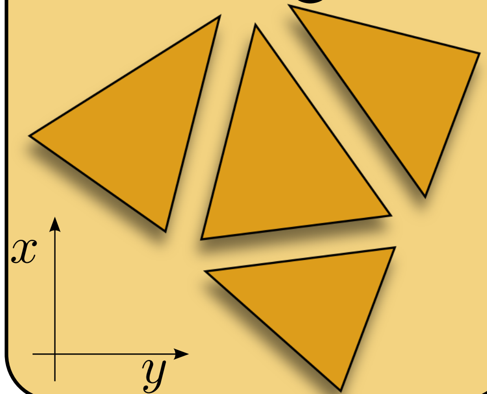
$$\downarrow$$

Weak form + IBP: $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

MultiRegions




LocalRegions




$$\mathbf{f}[i] = \boxed{\int_\Omega \Phi_i(x) f(x)\,dx} = \boxed{\sum_e^{N^{el}} \sum_p \int_{\Omega^e} \phi_p(x) f(x)\,dx}$$

# Nektar++: www.nektar.info

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

Nektar++ is a spectral/hp element toolkit written in C++.

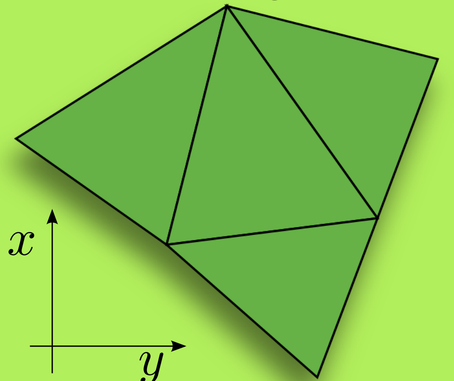
$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$

Helmholtz problem: $\nabla^2 u + \lambda u = f$

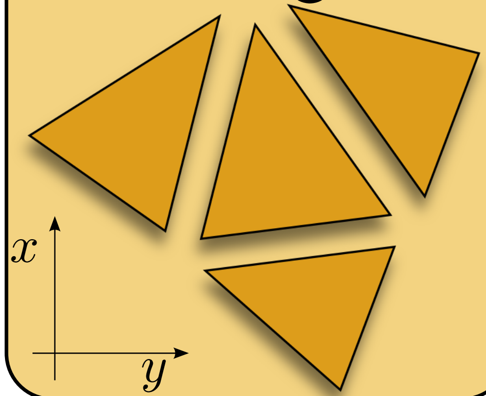
$$\downarrow$$

Weak form + IBP: $-(\nabla u, \nabla v) + \lambda(u,v) + (\nabla u, v)|_{d\Omega} = (f, v)$

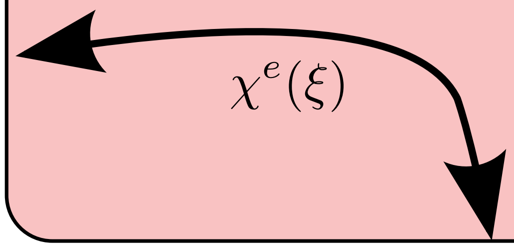
MultiRegions




LocalRegions




SpatialDomains




$$\mathbf{f}[i] = \int_\Omega \Phi_i(x) f(x)\,dx = \sum_e^{N^{el}} \sum_p \int_{\Omega^e} \phi_p(x) f(x)\,dx$$

$$= \sum_e \sum_p \int_{\Omega^e} \phi_p(\chi^e(\xi)) f(\chi^e(\xi)) J^e \, d\xi$$

Imperial College London

# Nektar++: www.nektar.info

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x)$$

$$u_e^\delta = \sum_p \hat{u}_p \phi_p(x)$$

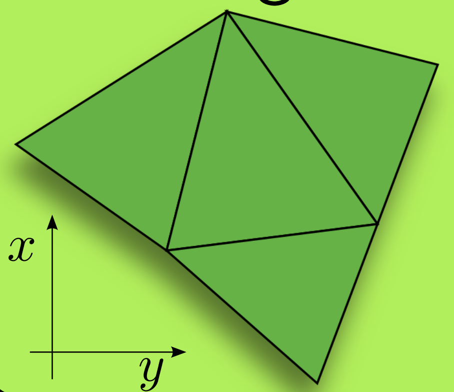
Nektar++ is a spectral/hp element toolkit written in C++.

Helmholtz problem: $\nabla^2 u + \lambda u = f$

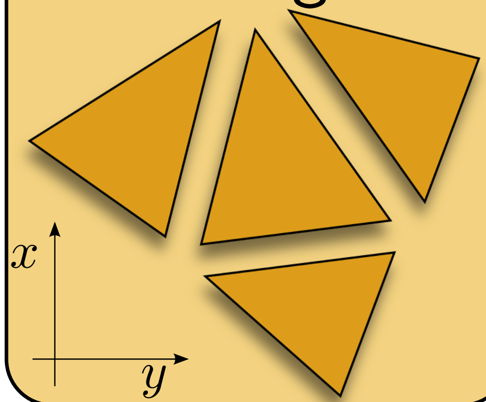
$$\downarrow$$

Weak form + IBP: $-(\nabla u, \nabla v) + \lambda(u, v) + (\nabla u, v)|_{d\Omega} = (f, v)$

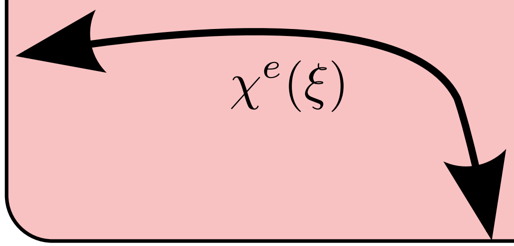
## MultiRegions



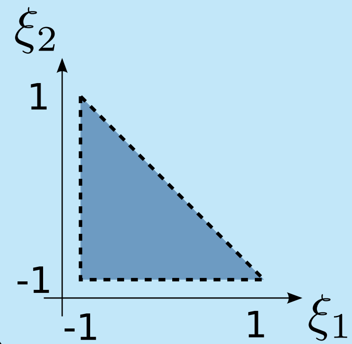

## LocalRegions




## SpatialDomains



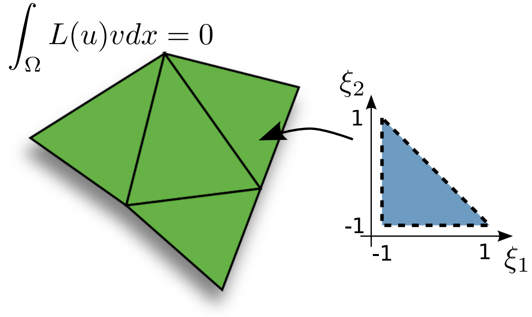

## StdRegions




$$\mathbf{f}[i] = \int_\Omega \Phi_i(x) f(x)\,dx = \sum_e^{N^{el}} \sum_p \int_{\Omega^e} \phi_p(x) f(x)\,dx$$

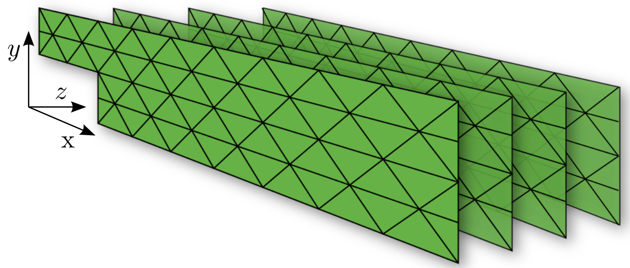
$$= \sum_e \sum_p \int_{\Omega^e} \phi_p(\chi^e(\xi)) f(\chi^e(\xi)) J^e\,d\xi$$

Imperial College London



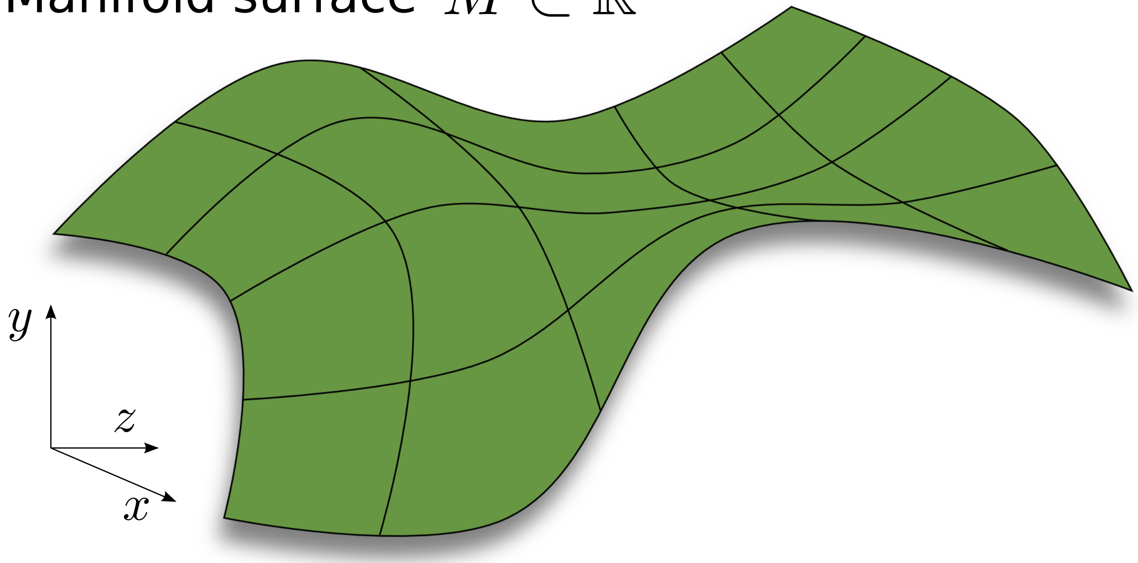

# Mathematical Construction

Expose different discretisations (CG, DG) by combining and reusing low-level elemental mathematical constructs.

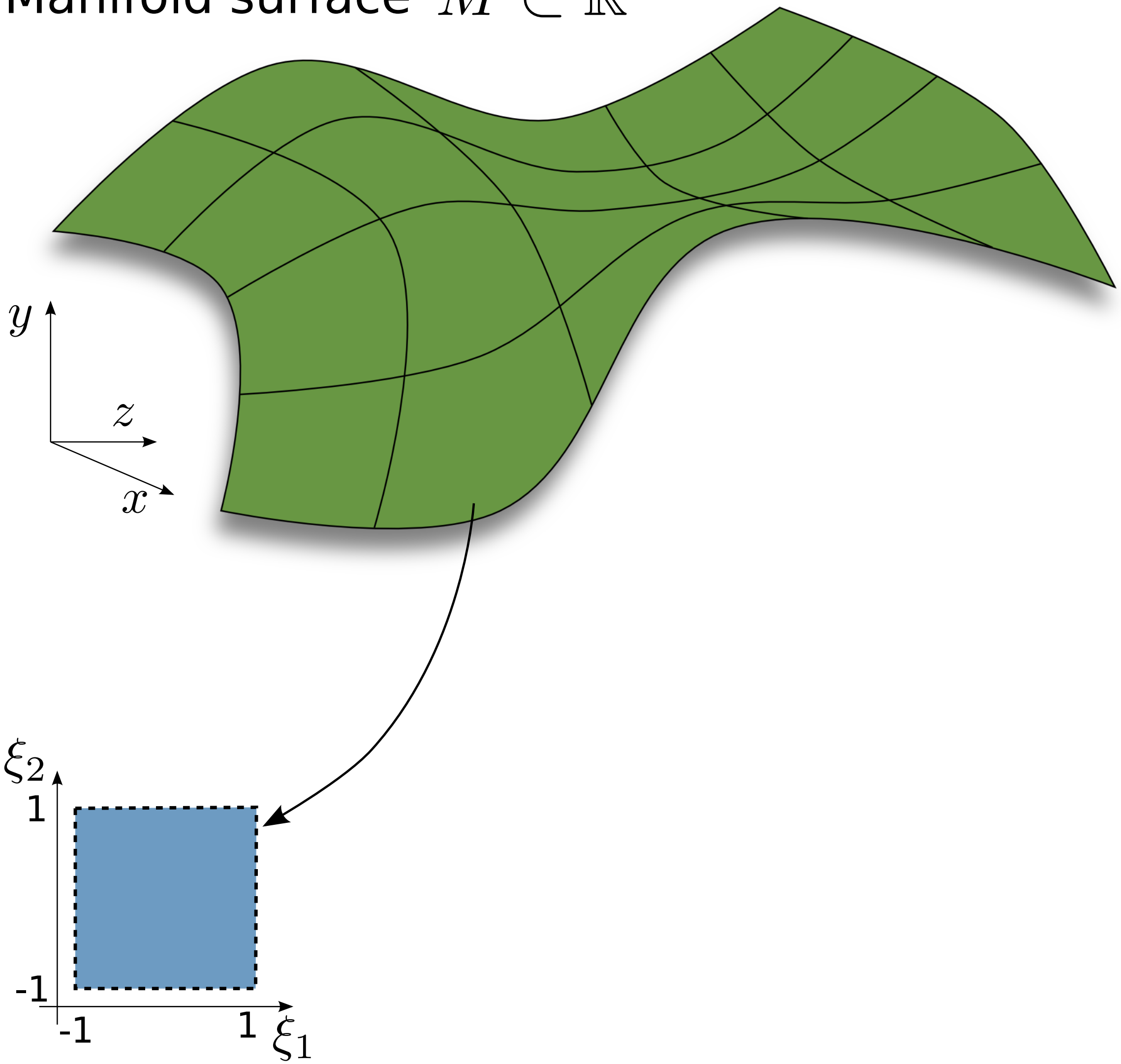



Retain and exploit domain symmetries and embeddings (homogeneous, cylindrical, manifold)

Manifold surface $M \subset \mathbb{R}^3$

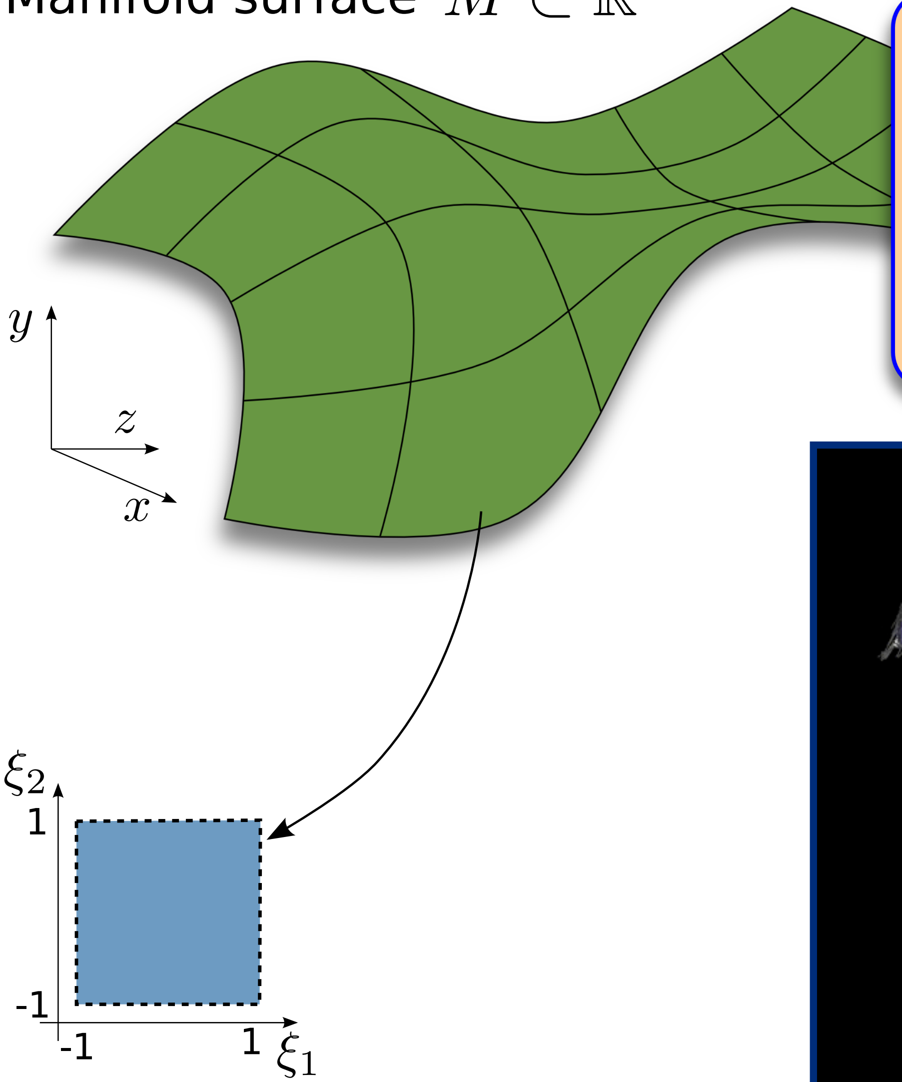


Manifold surface $M \subset \mathbb{R}^3$

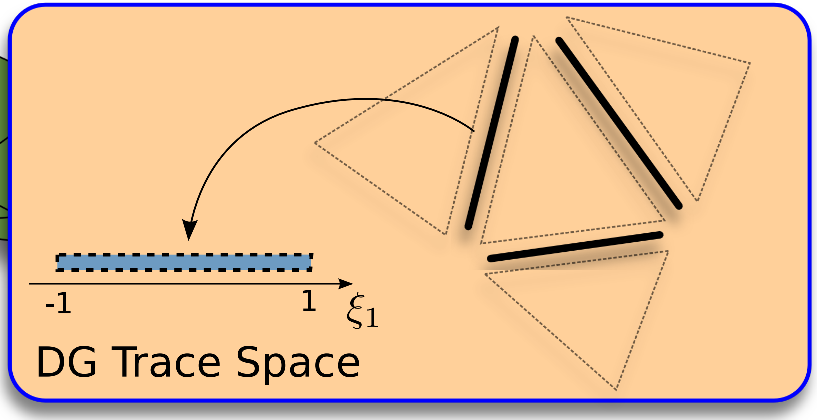


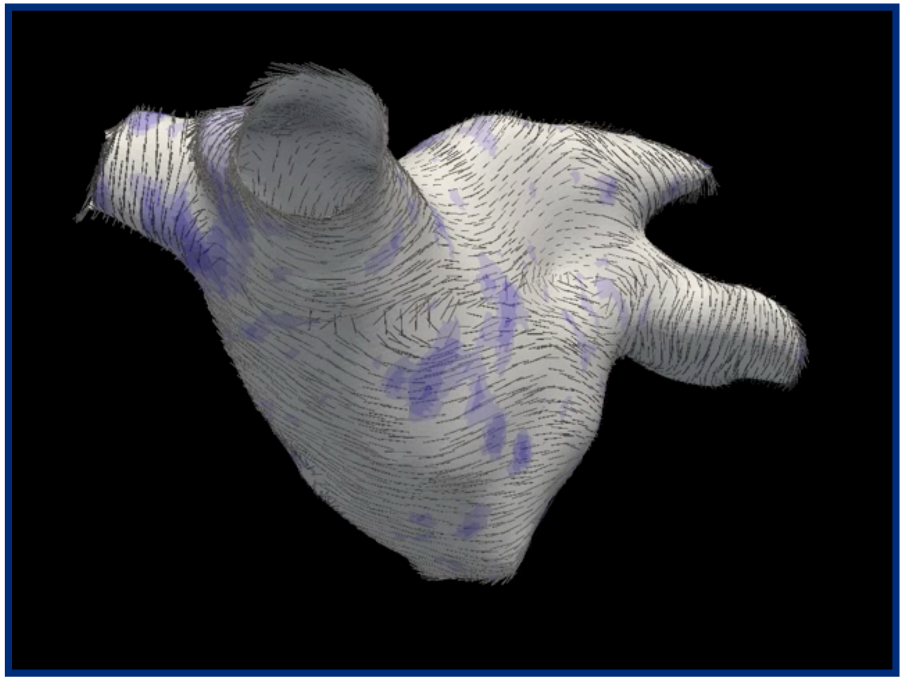

e.g. Laplace operator generalises to Laplace-Beltrami

Manifold surface $M \subset \mathbb{R}^3$



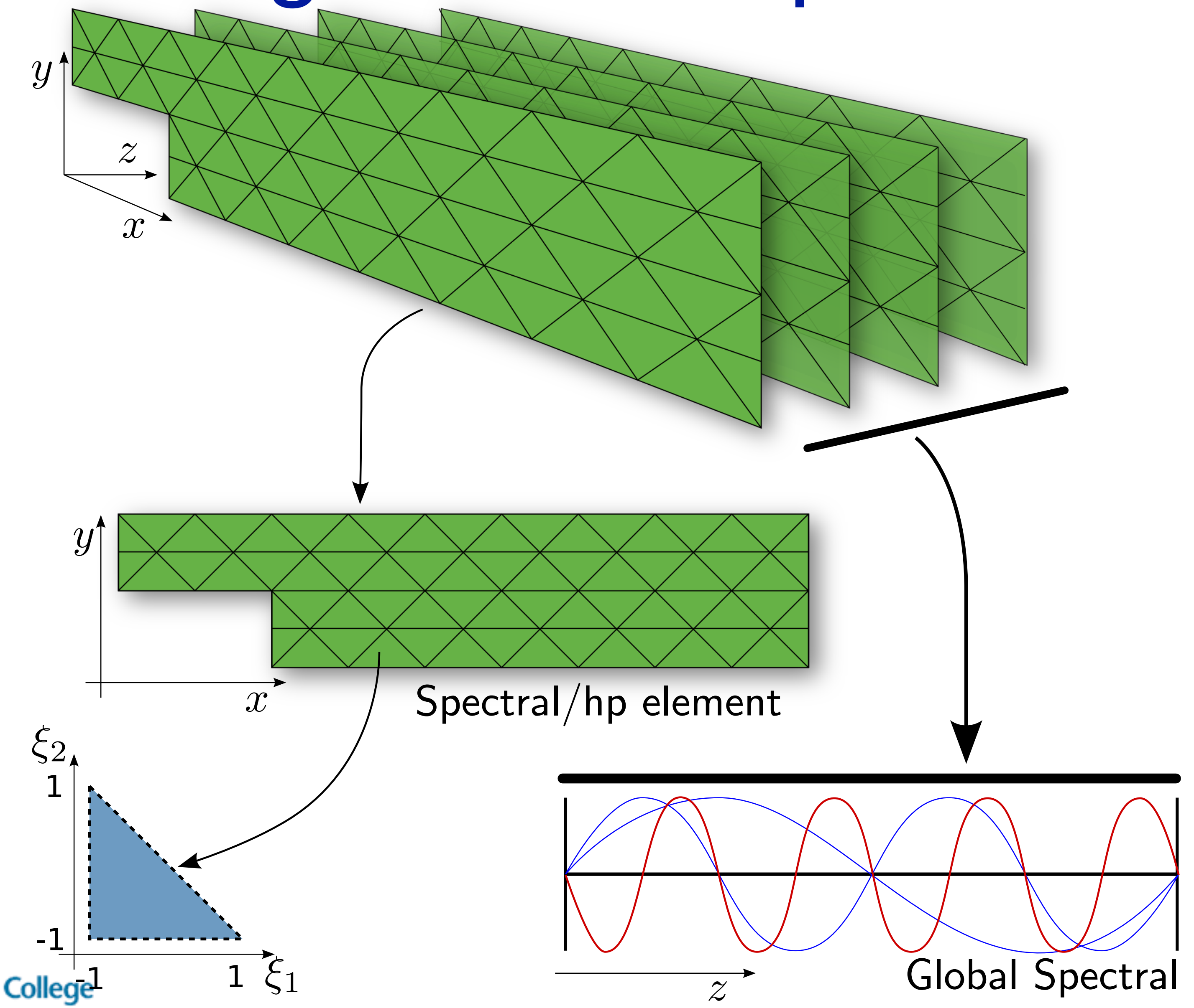

Examples



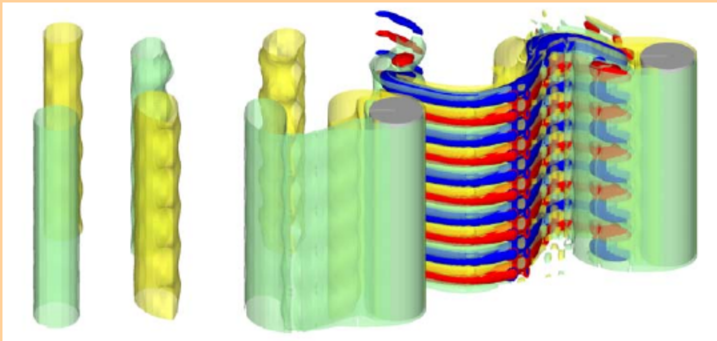
e.g. Laplace operator generalises to Laplace-Beltrami

# Homogeneous Expansions

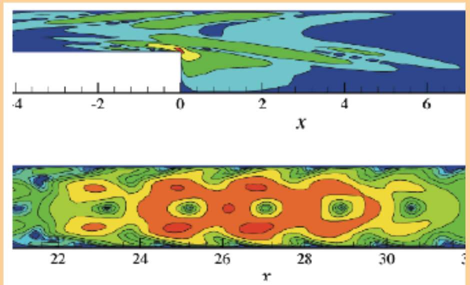



Imperial College London

# Direct Stability Analysis



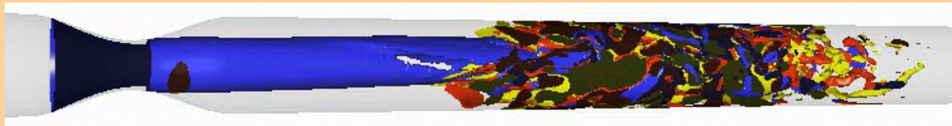
B. S. Carmo, S. J. Sherwin, P. W. Bearman, R. H. J. Willden
J. Fluid Mech. (2008), vol. 597, pp. 1–29.



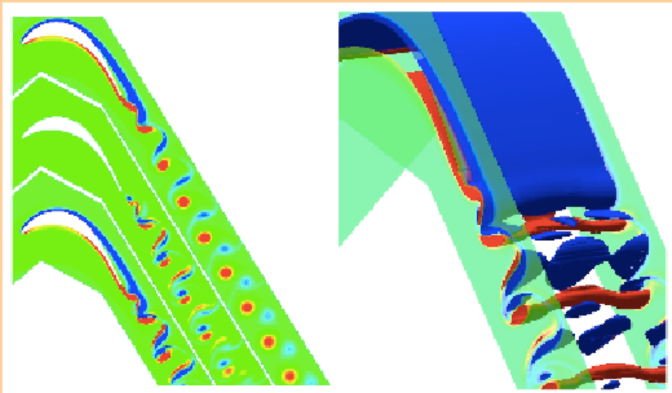

H. M. Blackburn, D. Barkley, S. J. Sherwin
J. Fluid Mech. (2008), vol. 603, pp. 271-304.

Stability of basic flows

Complex Geometry LNS & DNS



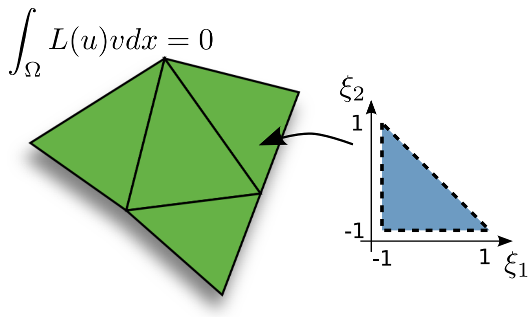
S. J. Sherwin, H. M. Blackburn
J. Fluid Mech. (2005), vol. 533, pp. 297–327.



A. S. Sharma, N. Abdessemed, S. J. Sherwin, V. Theofilis
Theor. Comput. Fluid Dyn. (2011) 25:19–30

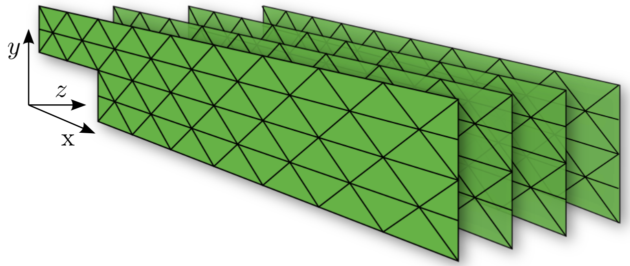
Biomedical and Engineering applications

Imperial College London




# Mathematical Construction

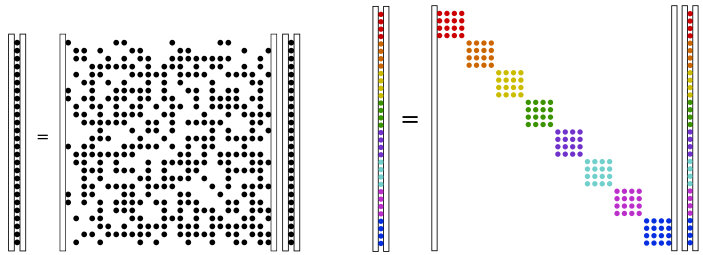
Expose different discretisations (CG, DG) by combining and reusing low-level elemental mathematical constructs.




Retain and exploit domain symmetries and embeddings (homogeneous, cylindrical, manifold)

# Computational Implementation

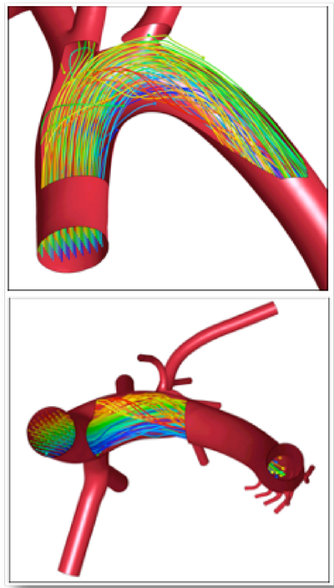
Challenge high-/low-order boundaries while maintaining efficiency.



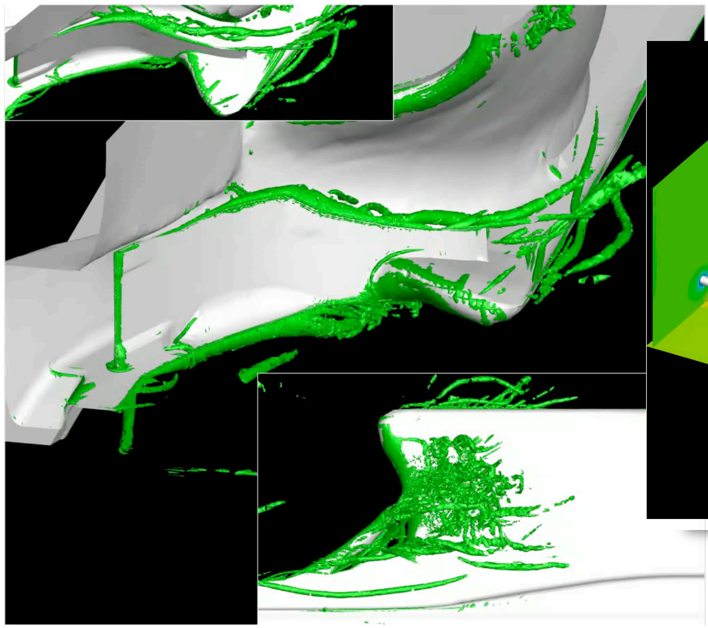
What do we understand by low and high order? When is each choice useful?

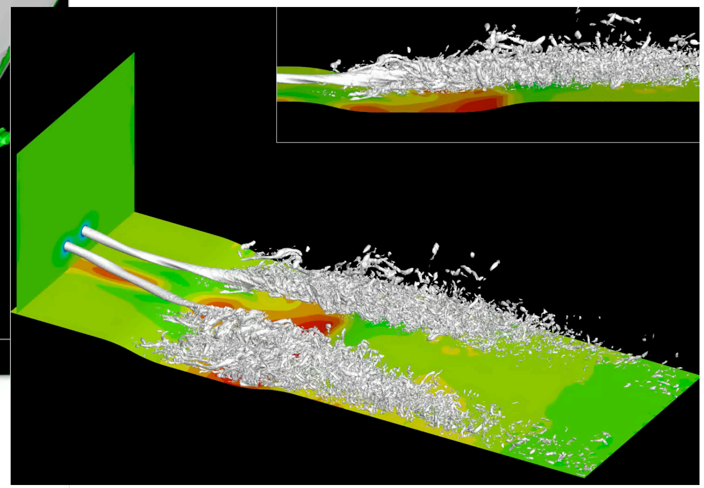
Low Order
$P = 1$

High Order
$P = 15$



P. E. Vincent, A. M. Plata, A. A. E. Hunt, P. D. Weinberg
J. R. Soc. Interface, 10.1098/rsif.2011.0116





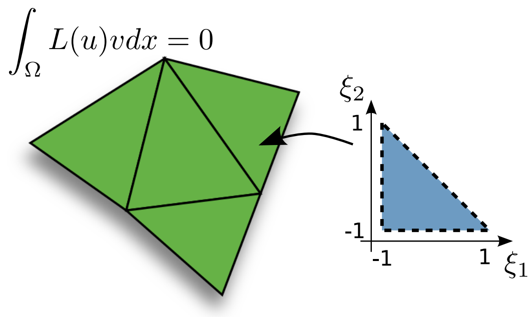
Julien Hoessler PhD Thesis 2012

High Reynolds number flows.

Low Reynolds number flows.

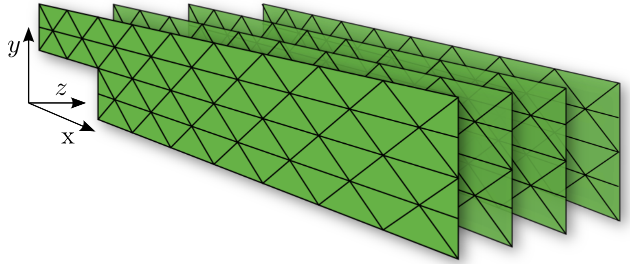
Complex geometry.

How do we implement low- and high-order techniques efficiently across current and future hardware?




## Mathematical Construction

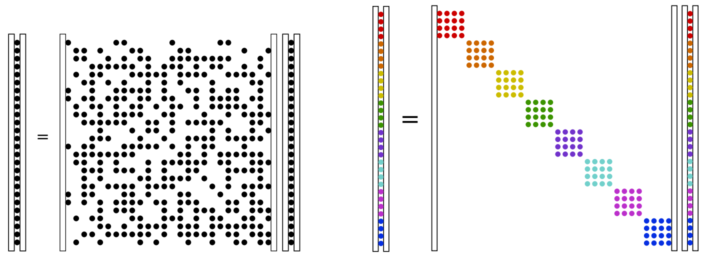
Expose different discretisations (CG, DG) by combining and reusing low-level elemental mathematical constructs.

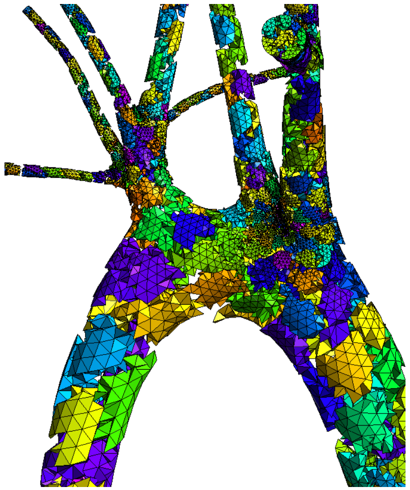



Retain and exploit domain symmetries and embeddings (homogeneous, cylindrical, manifold)

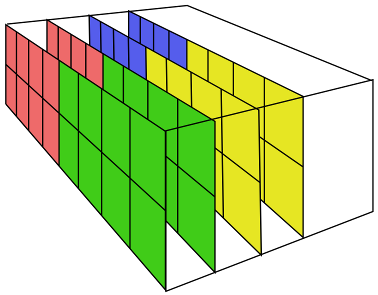
## Computational Implementation

Challenge high-/low-order boundaries while maintaining efficiency.



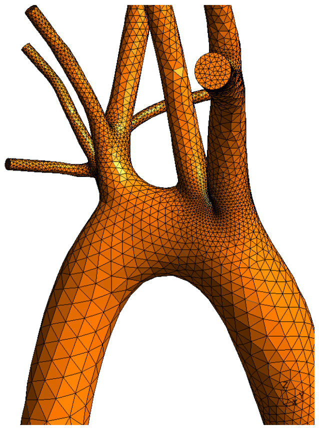



Bridge current and future hardware diversity through hybrid implementation strategies.

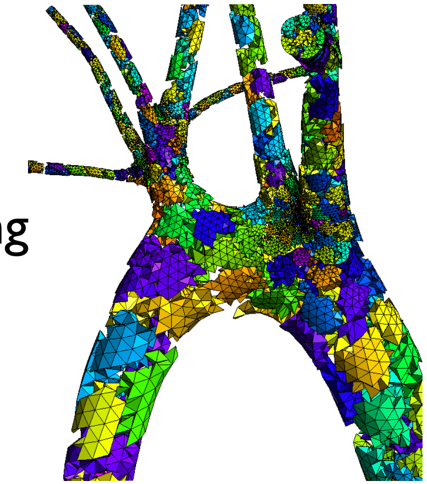


Achieve flexible HPC scalability and performance through mixed parallelism.

# Hybrid Numbering & Mixed Parallelisation

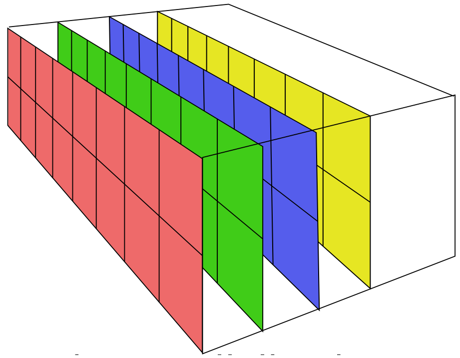


Global numbering

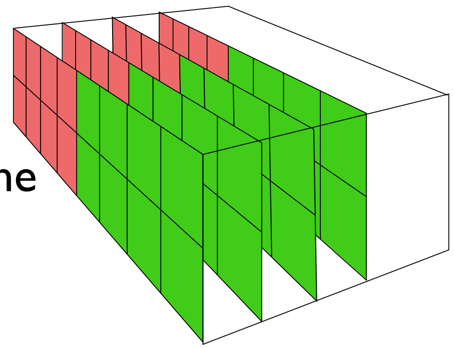


Hybrid numbering

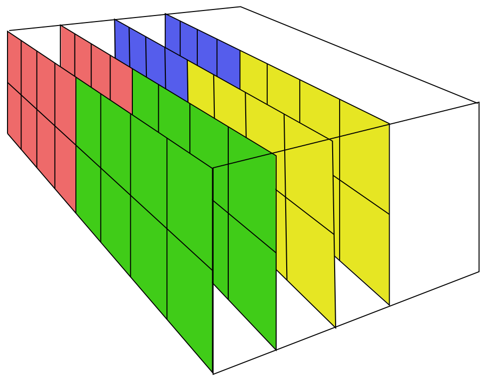
Local numbering



Fourier parallelisation



Spectral/hp plane parallelisation

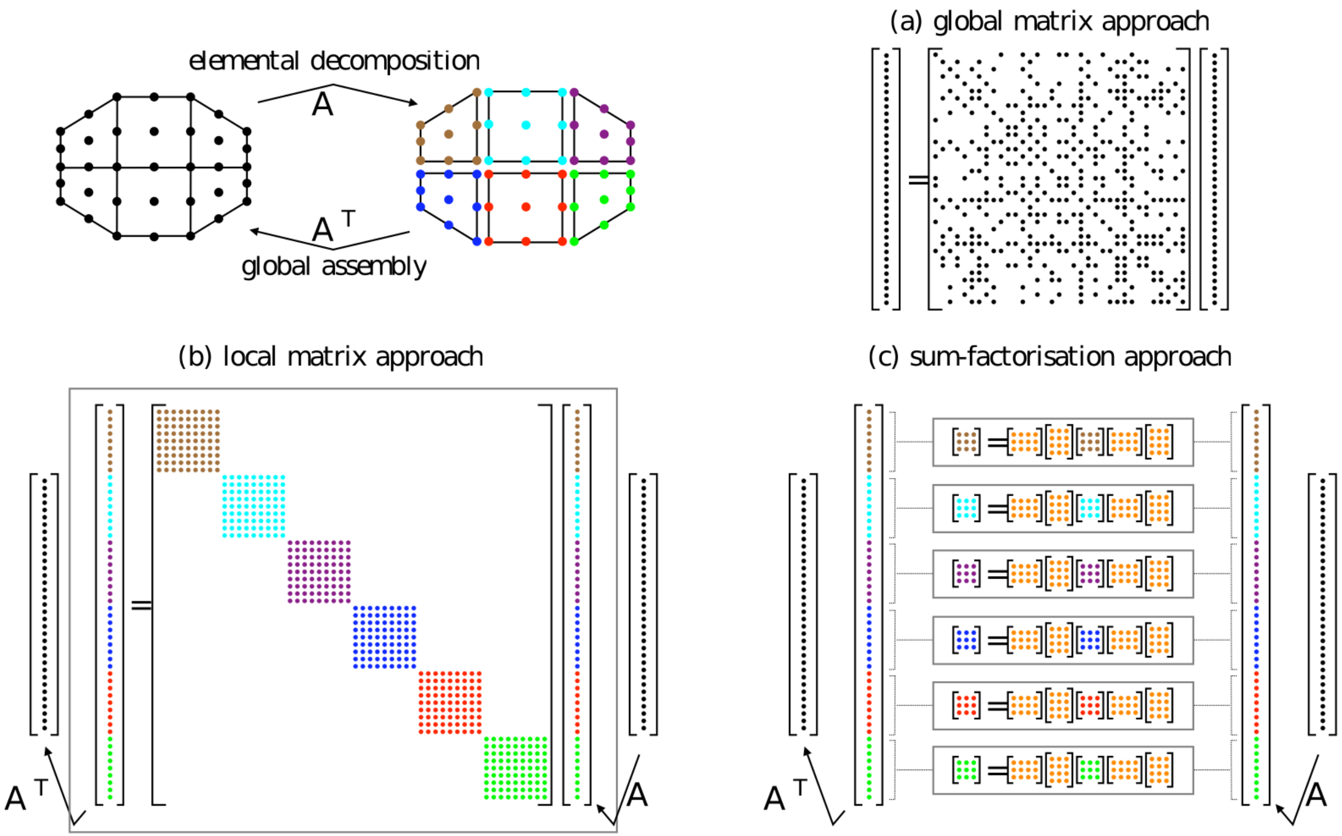


Mixed parallelisation

# Outline

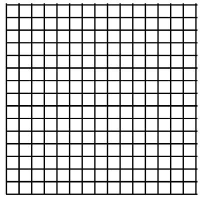
- What are we doing?
- Optimizing our implementations
  - From h to p efficiently
  - Variable p
  - Cloud Computing

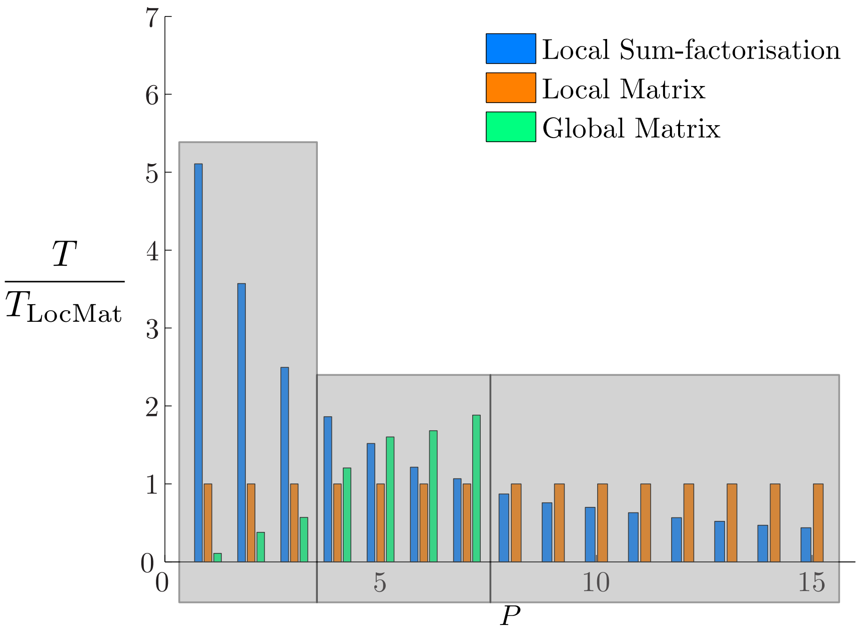
# Evaluation Strategies for iterative solvers

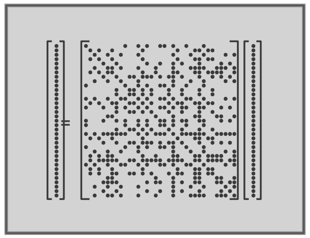


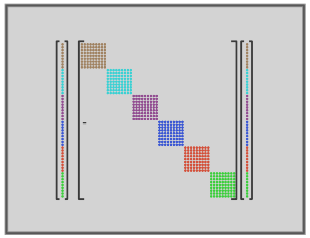
# Computational results

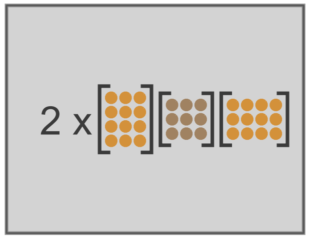
- mass matrix operator: $\hat{g}_i = \sum_j (\Phi_i, \Phi_j)_\Omega \hat{f}_j \quad \forall i$





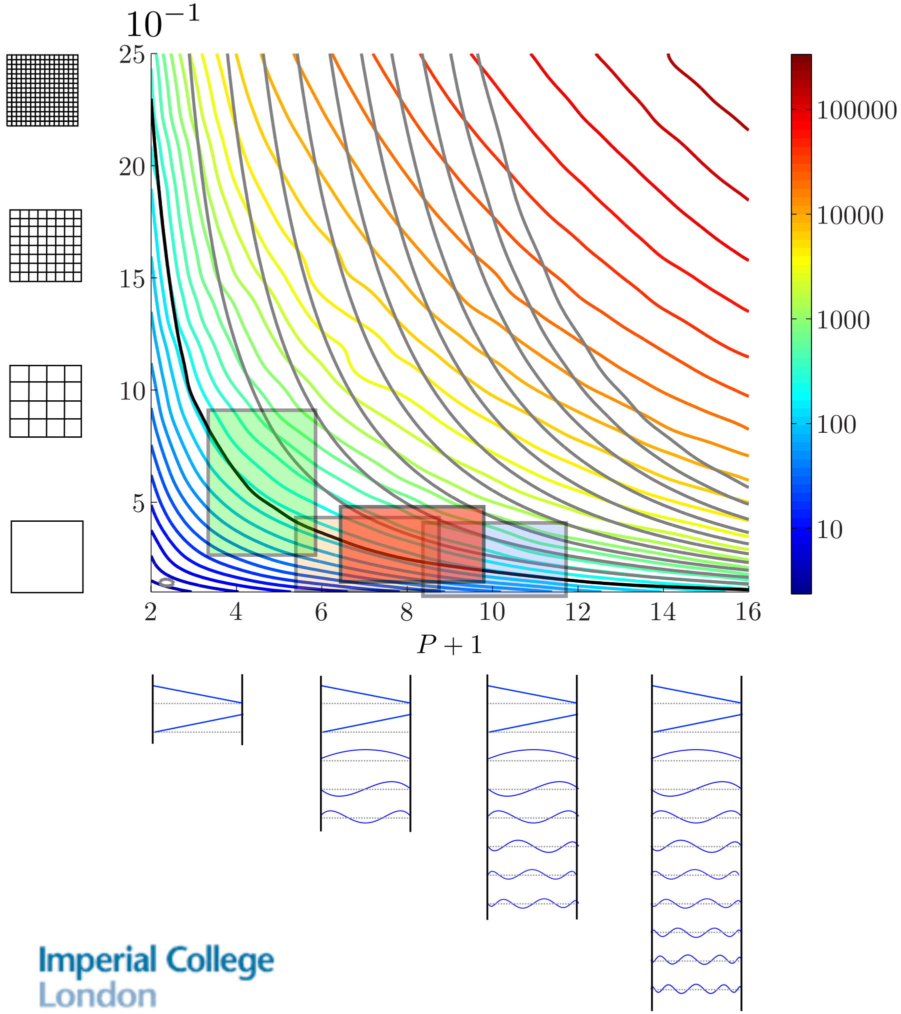


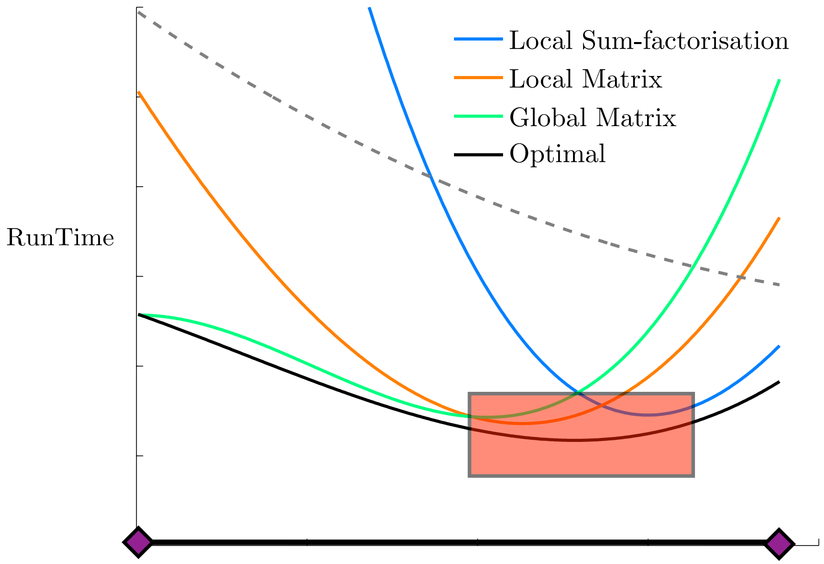



*Vos, Sherwin, Kirby, JCP, 2010*

# Error vs computational cost?

- minimal run-time



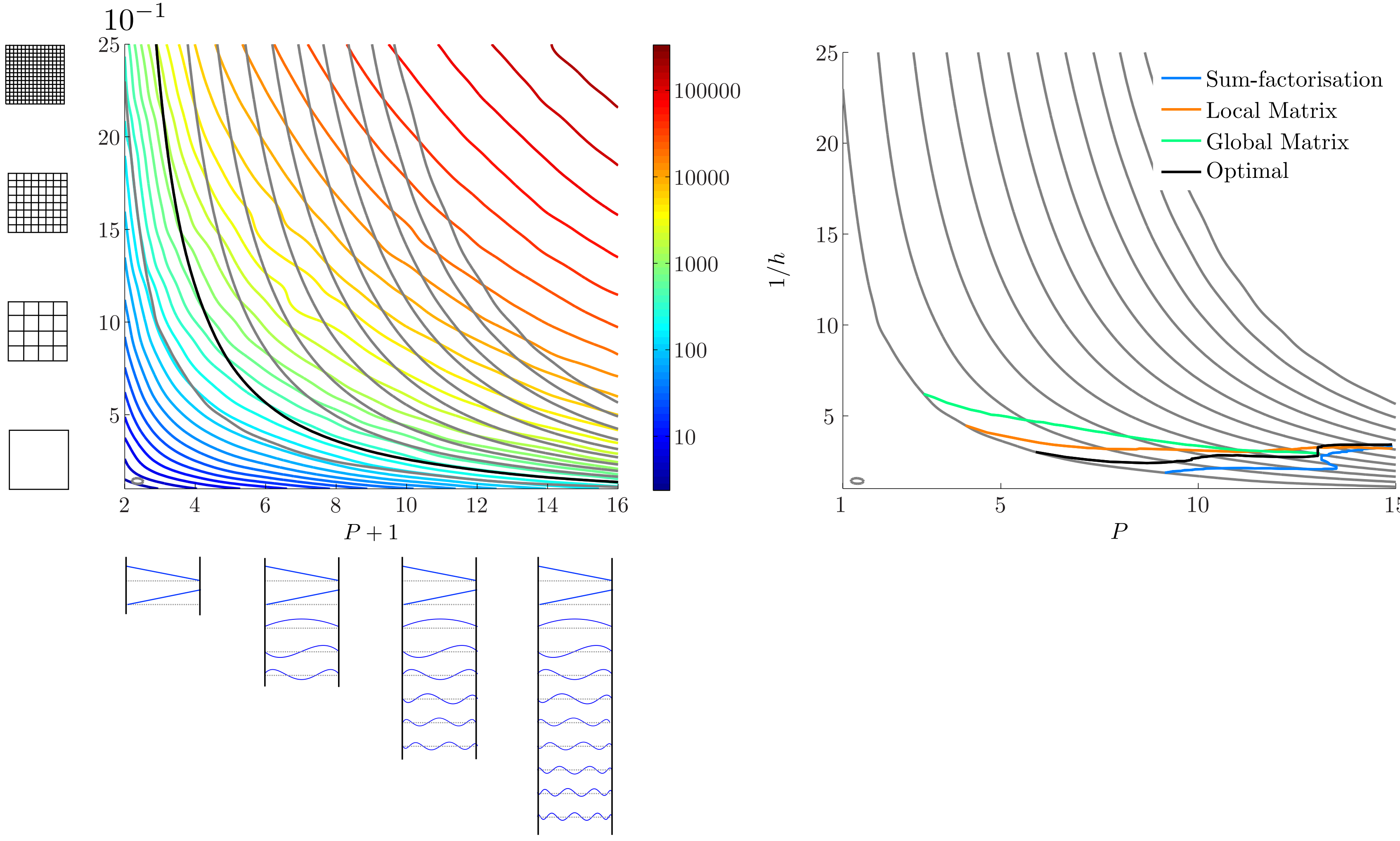




9 Elements, P=6

Imperial College
London

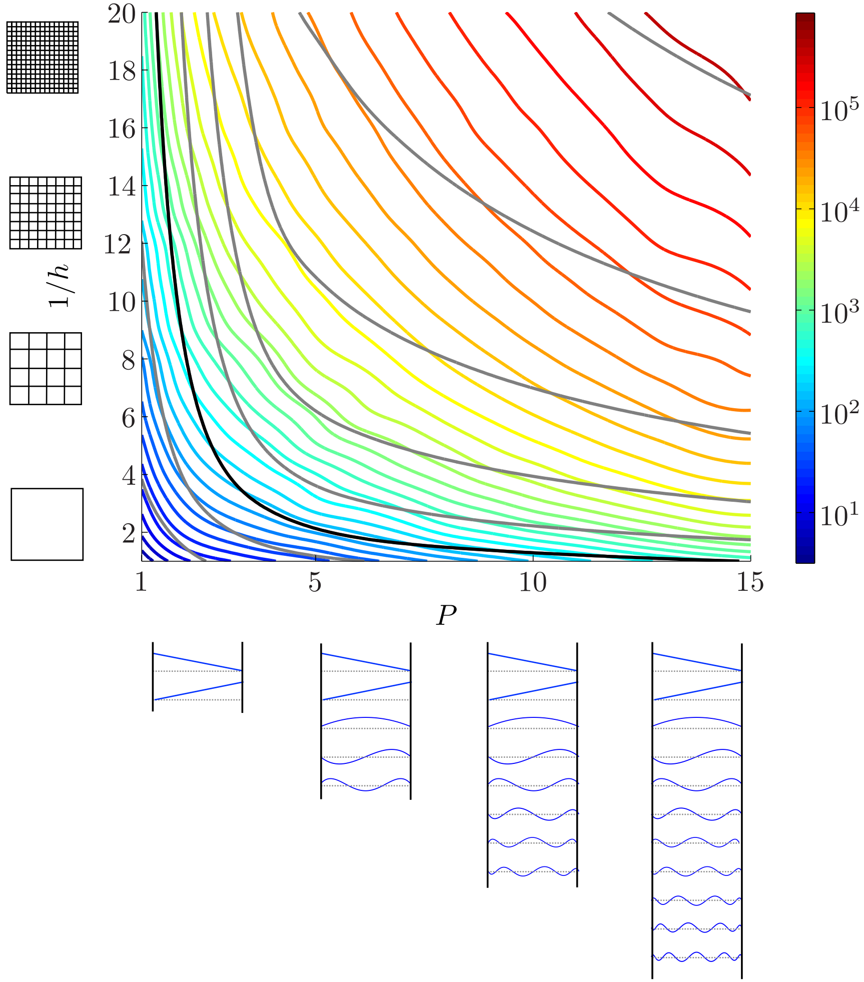
# Error vs computational cost?

- minimal run-time - *path of optimal discretisations*



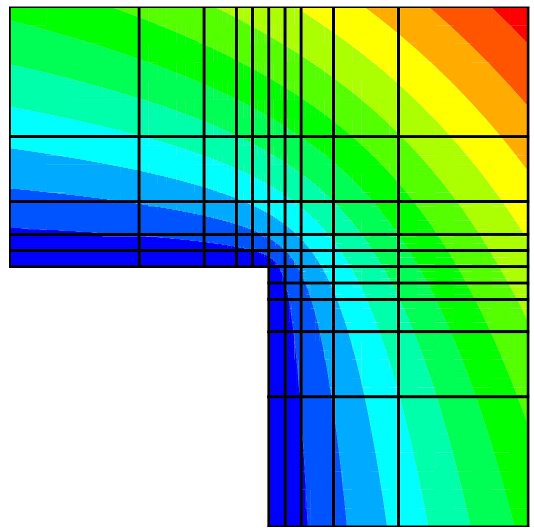
# Computational results: Non smooth solution

- minimal run-time - corner problem (error = $10^{-4}$)




Helmholtz equation

$$\nabla^2 u(\boldsymbol{x}) - \lambda u(\boldsymbol{x}) = f(\boldsymbol{x}) \qquad x_i = \left(\frac{i}{n_{el}}\right)^3$$



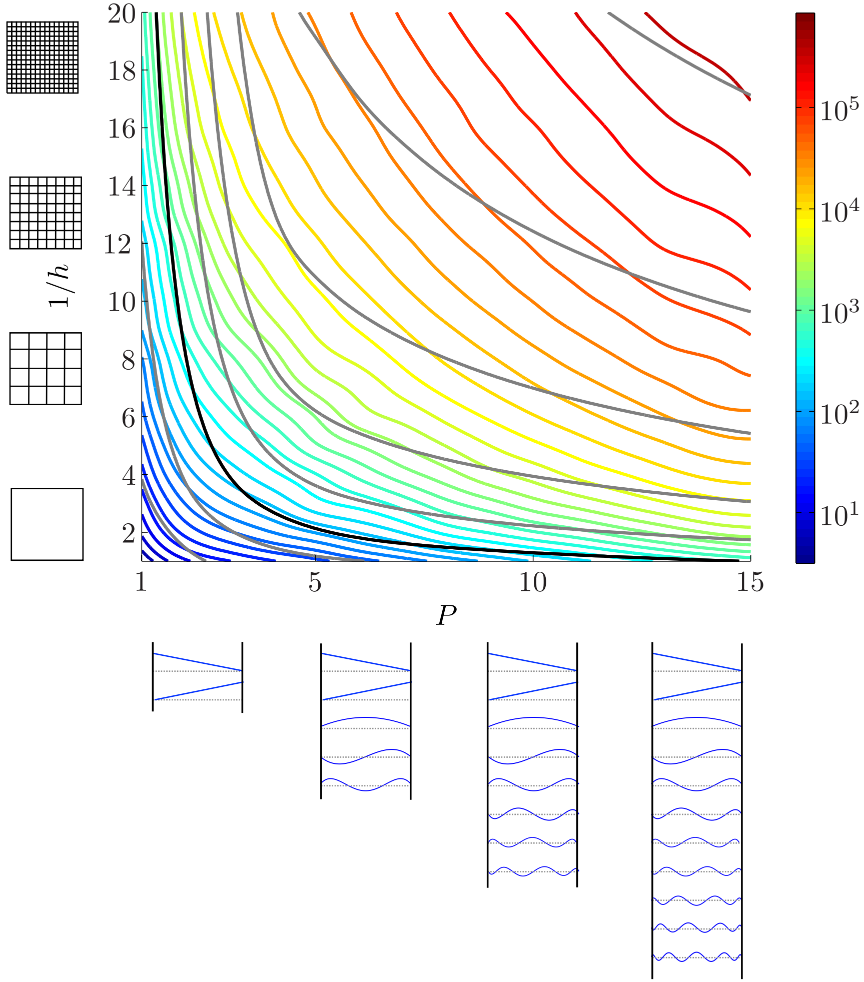
$$u(r,\theta) = r^{\frac{2}{3}} \sin(\frac{2}{3}\theta + \frac{\pi}{3})$$

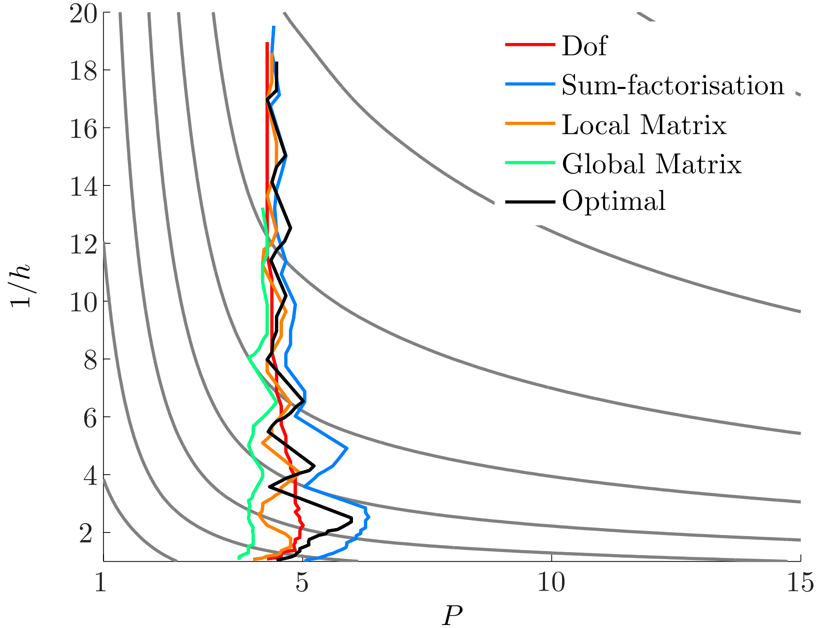
irregular solution

# Computational results

Imperial College London

- minimal run-time - path of optimal discretisations

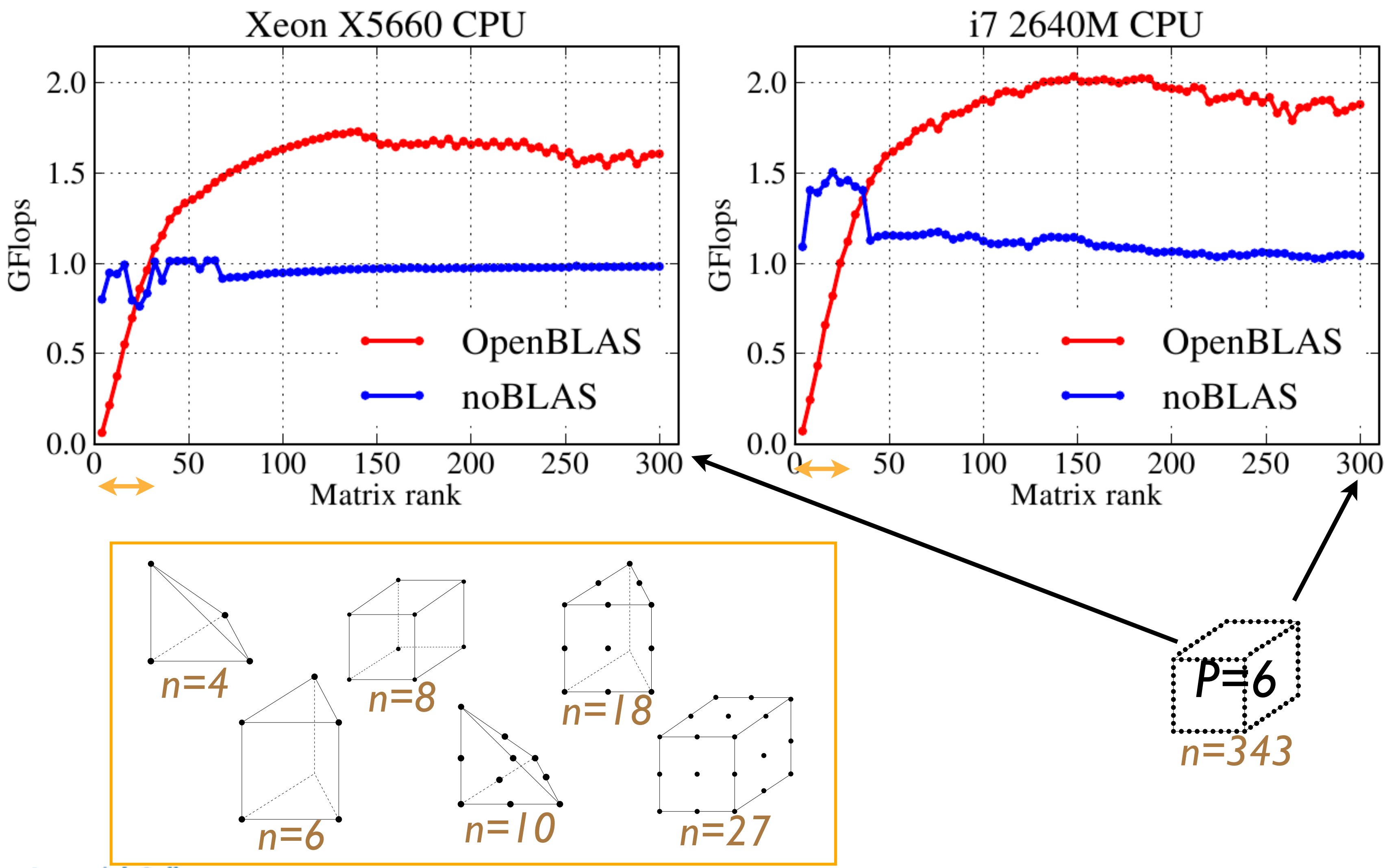


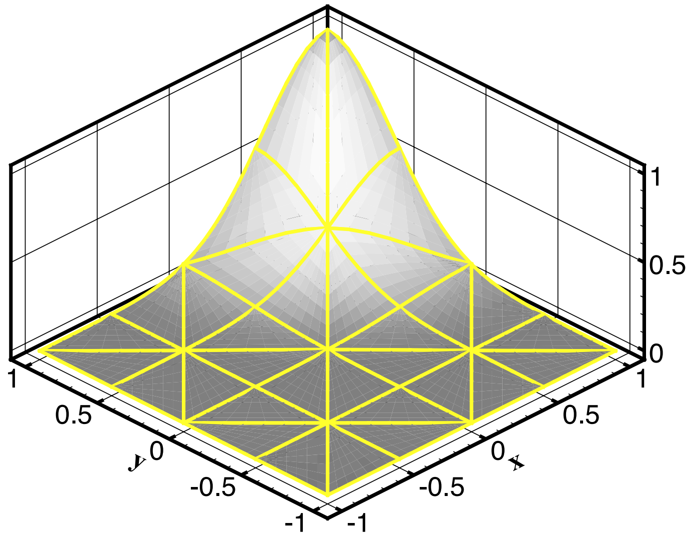
# Outline

- What are we doing?
- Optimizing our implementations
  - Variable p
  - Cloud Computing

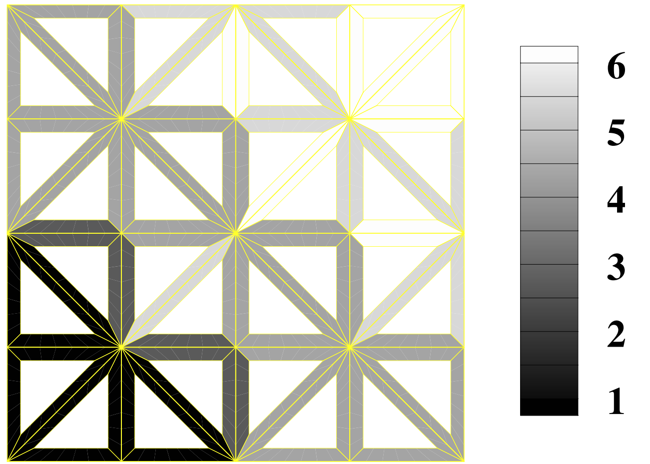
# Local Matrix: Hardware diversity



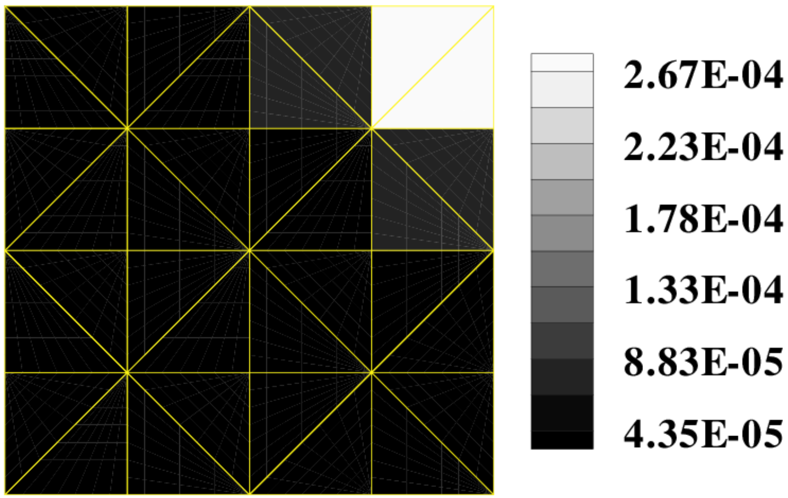
# Variable matrix size due to variable p



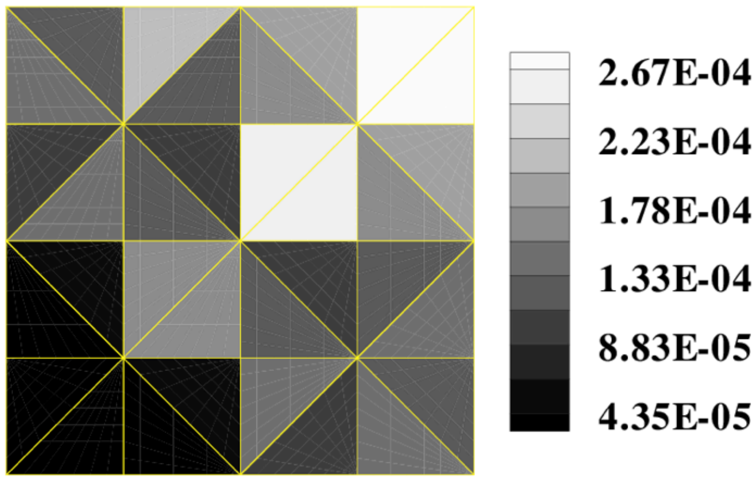

Solution




Variable P



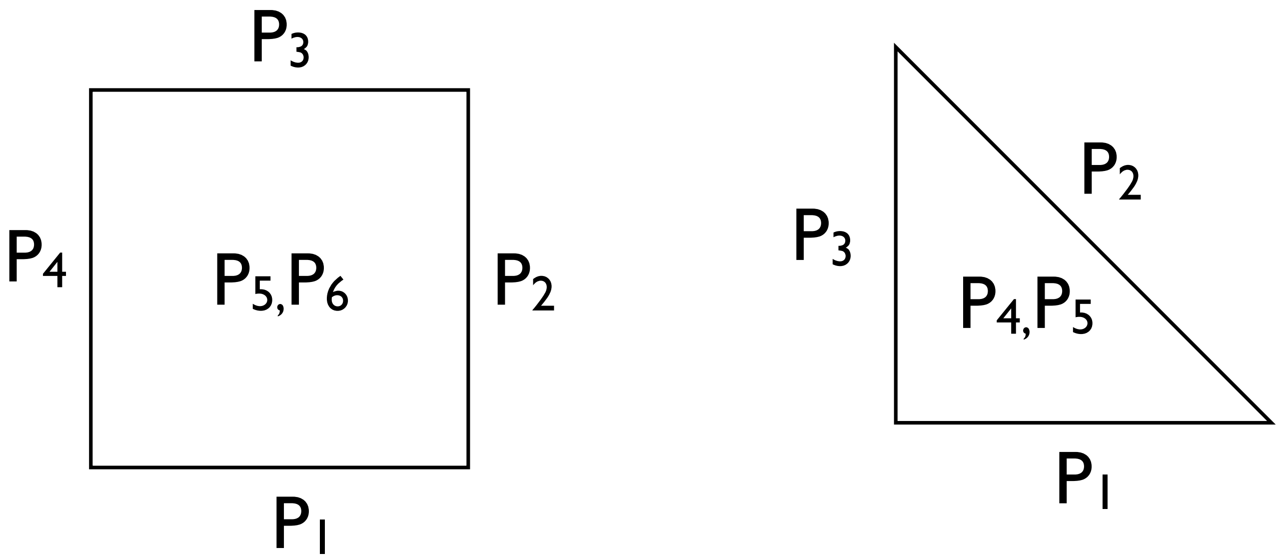

Fixed P error




Variable P Error

Imperial College London

# How many parameters?



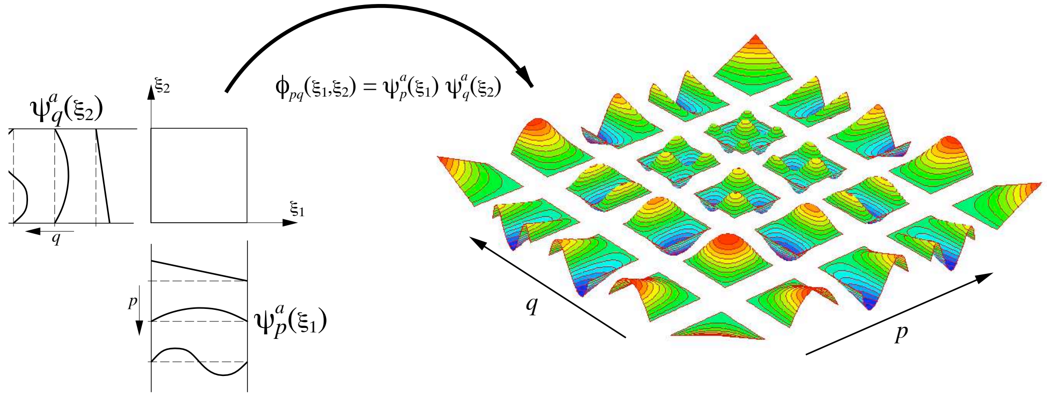

6 parameters

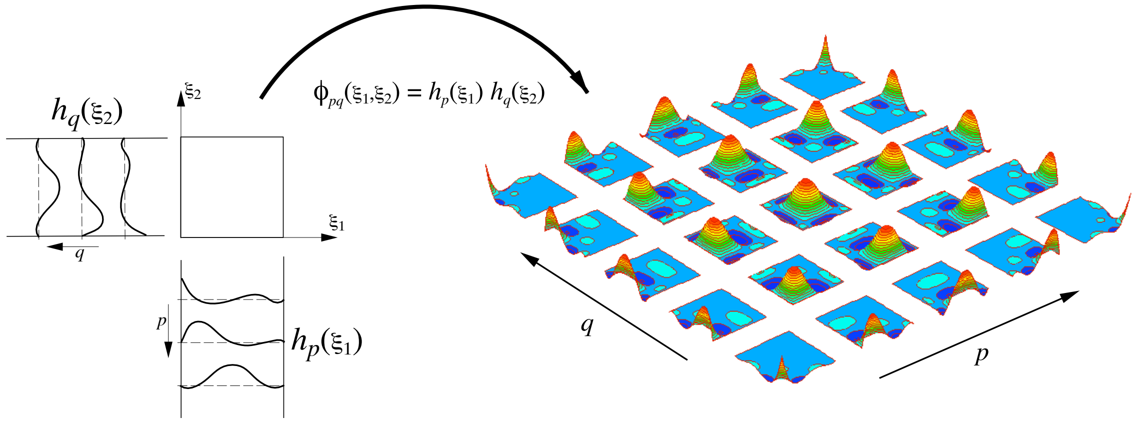
5 parameters

Hexahedral: 12 edges, 8 faces, 1 interior = 31 parameters

Tetrahedral: 6 edges, 4 faces, 1 interior = 17 parameters

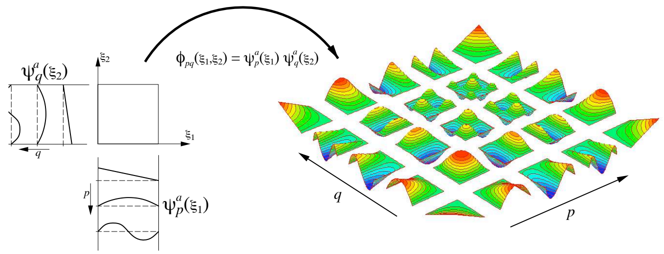
# Tensor product design

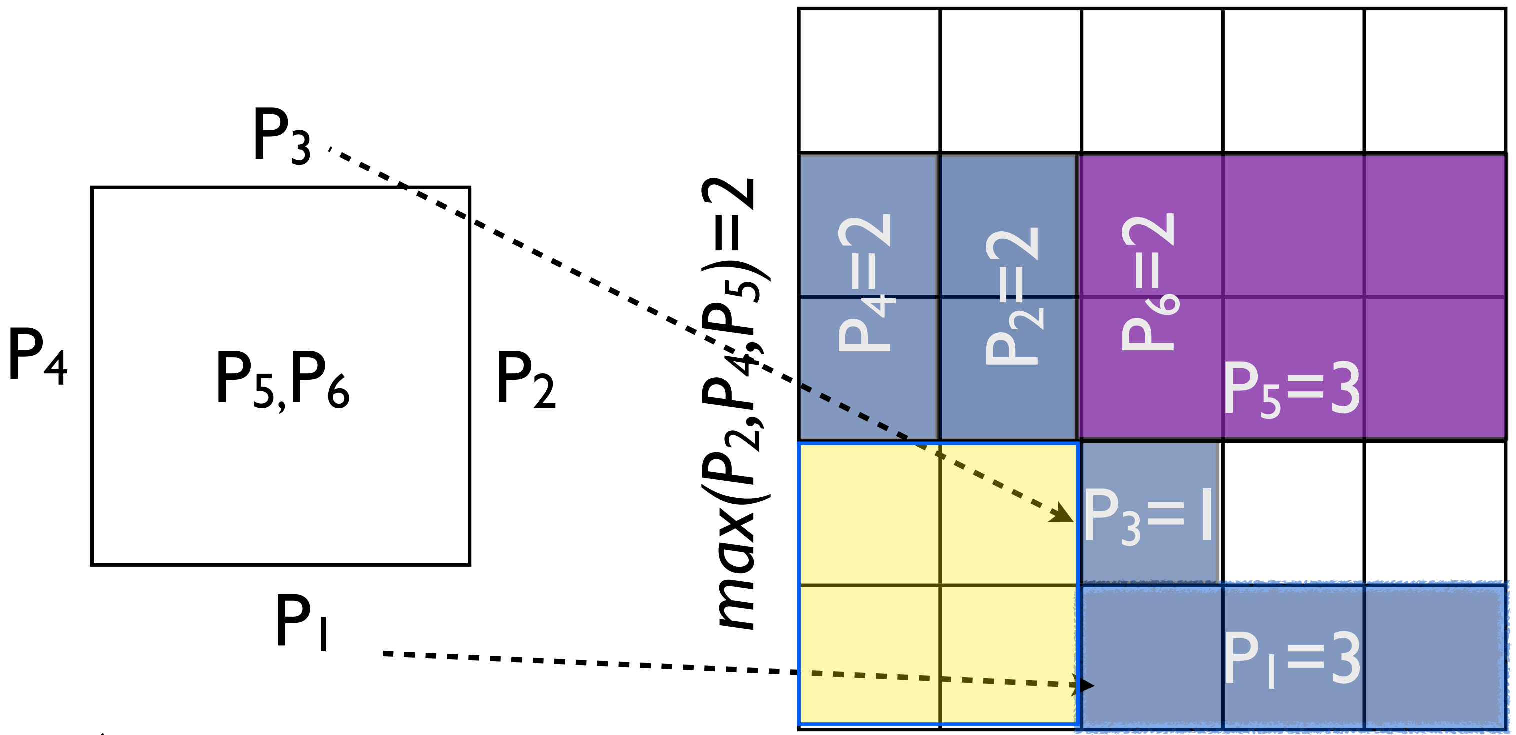



Imperial College London

# Zero packing



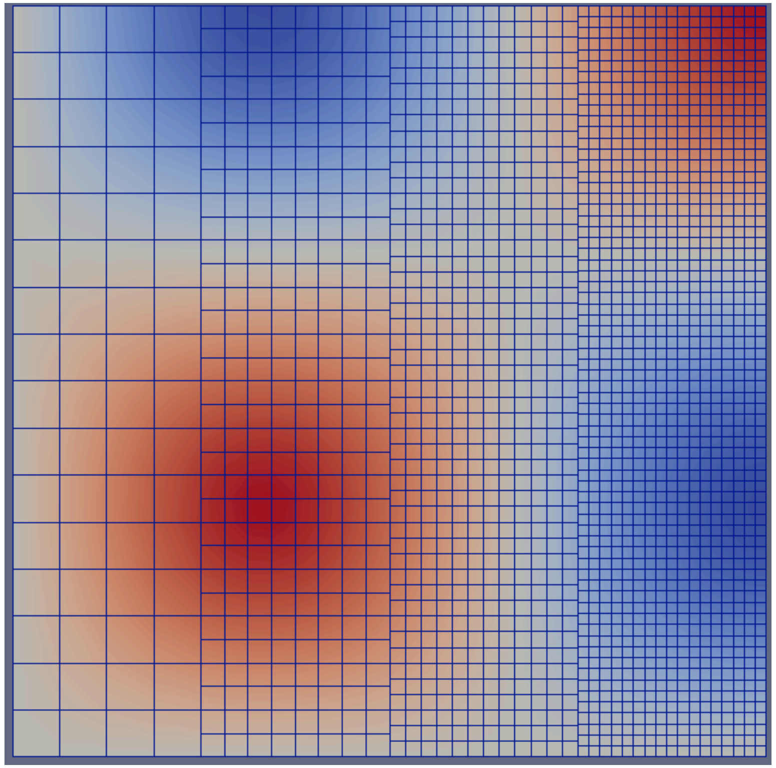




6 parameters

$max(P_1,P_3,P_5)=3$

2 parameters

# Helmholtz example

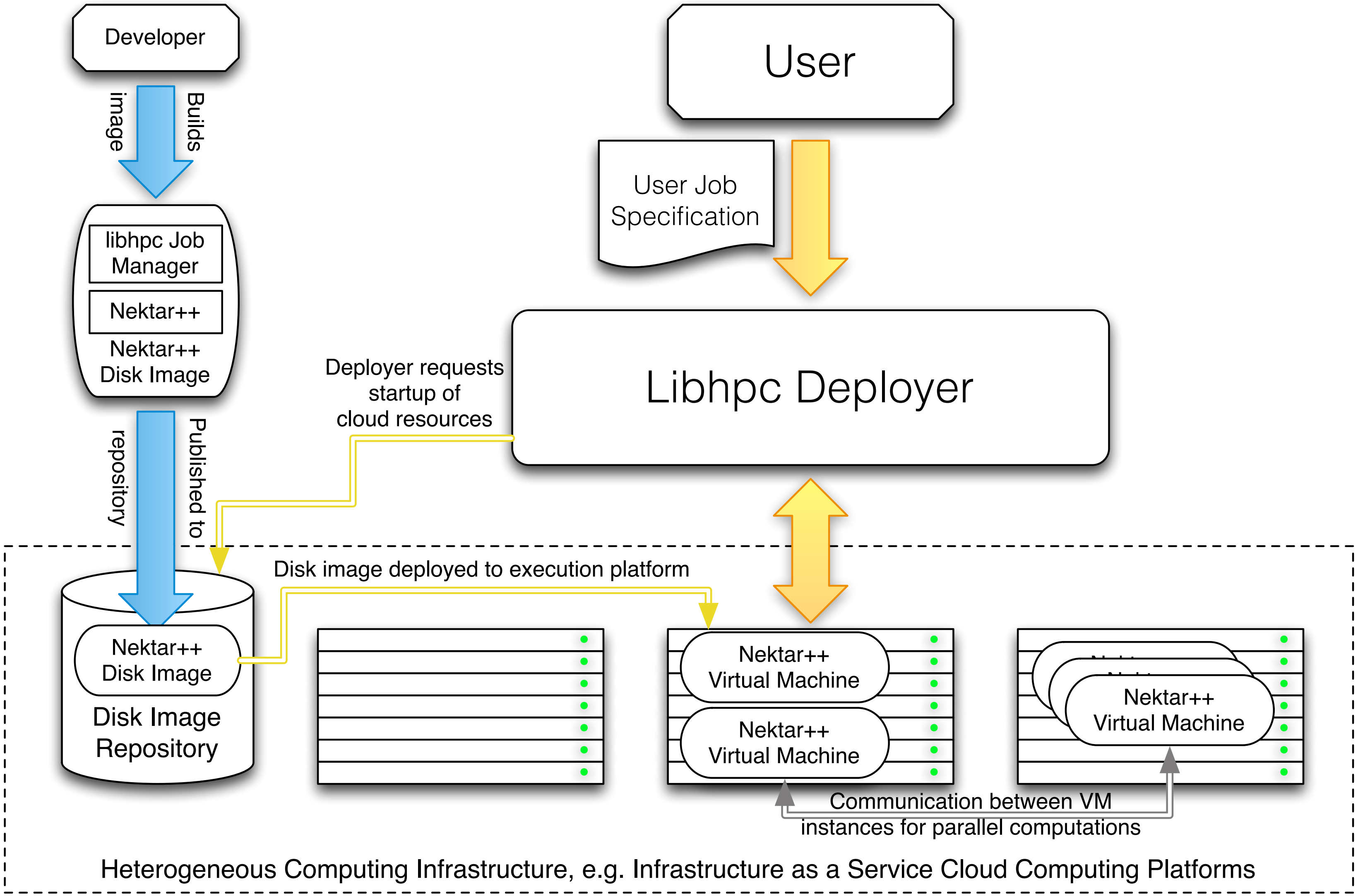


P=4 P=8 P=12 P=17

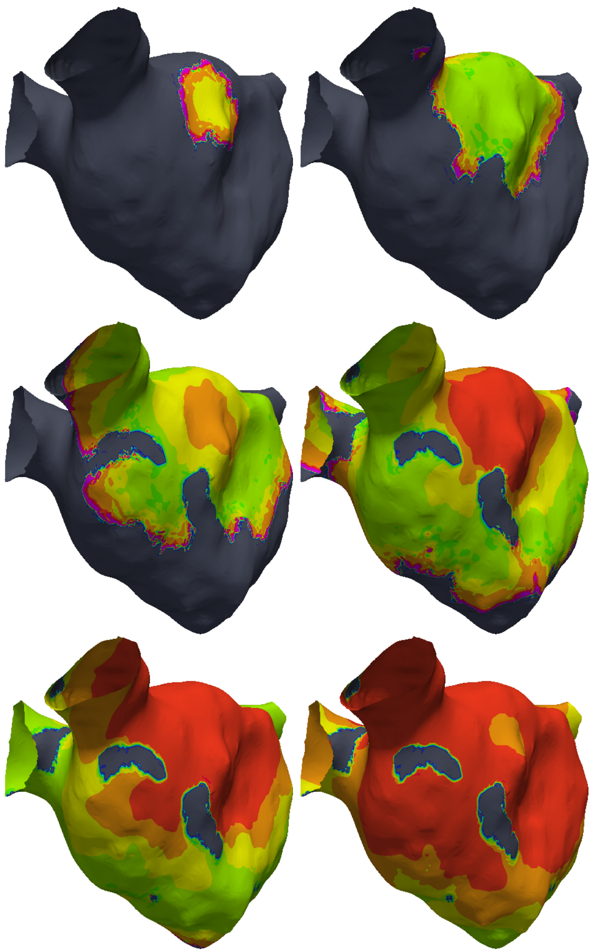
# LibHPC

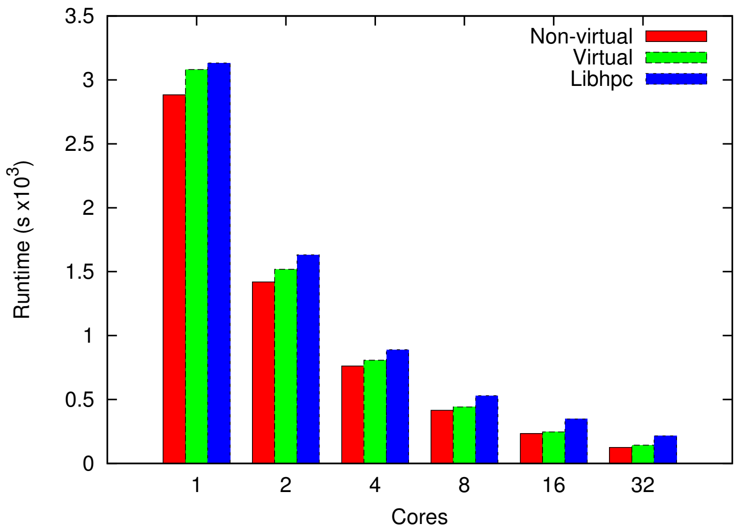
*J. Cohen, P. Burovskiy, J. Darlington*

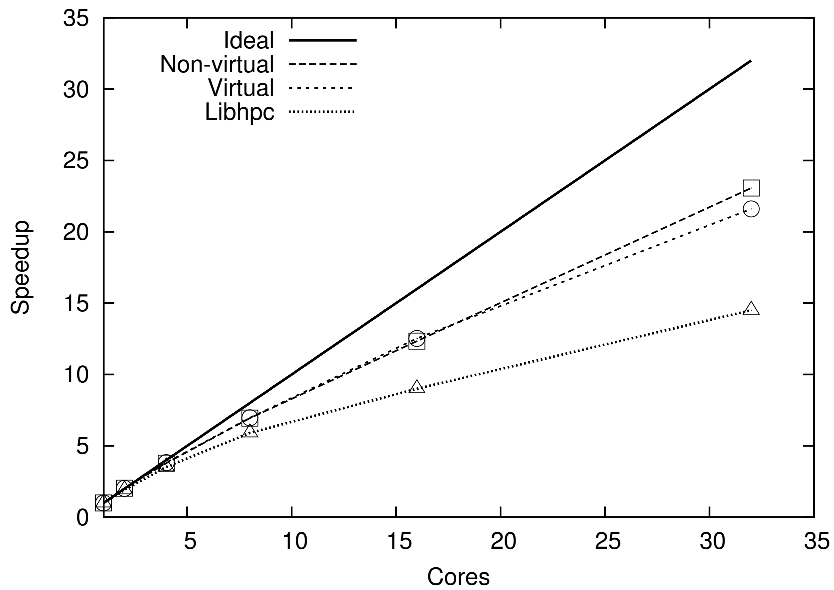
- Target software on multi-core, distributed & hetrogeneous platforms
- Run on Infrastructure-as-a-service (IaaS) clouds
- Why?
  - Intermittent running makes access to HPC difficult
  - Scale resource beyond local capacity




Imperial College London






Imperial College
London

# Summary

- Presented implementation optimisations to blend high and low order polynomial order
  - Mixed implementation of basic operators
  - Mixed Fourier discretisations.
- Does cloud computing offer possibilities?

Imperial College London

EPSRC
Engineering and Physical Sciences Research Council

THE UNIVERSITY OF UTAH