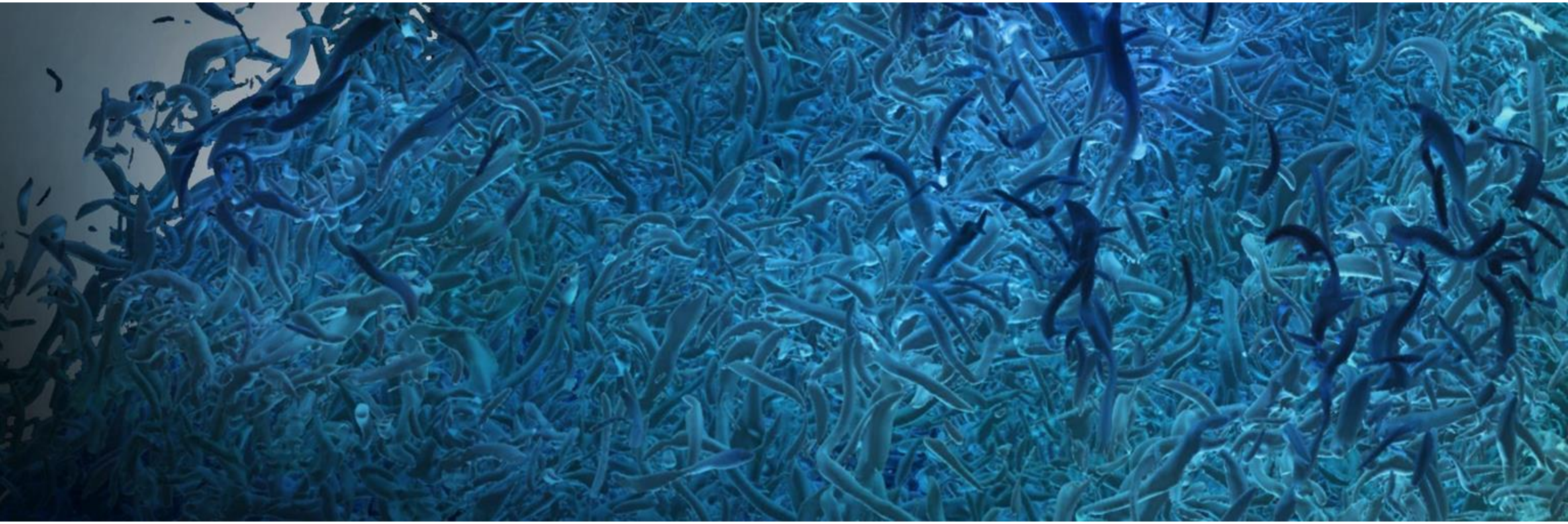
ExaFLOW – Towards Exascale in High-Order Computational Fluid Dynamics

Philipp Schlatter, KTH Mechanics, Royal Institute of Technology,
Stockholm, Sweden

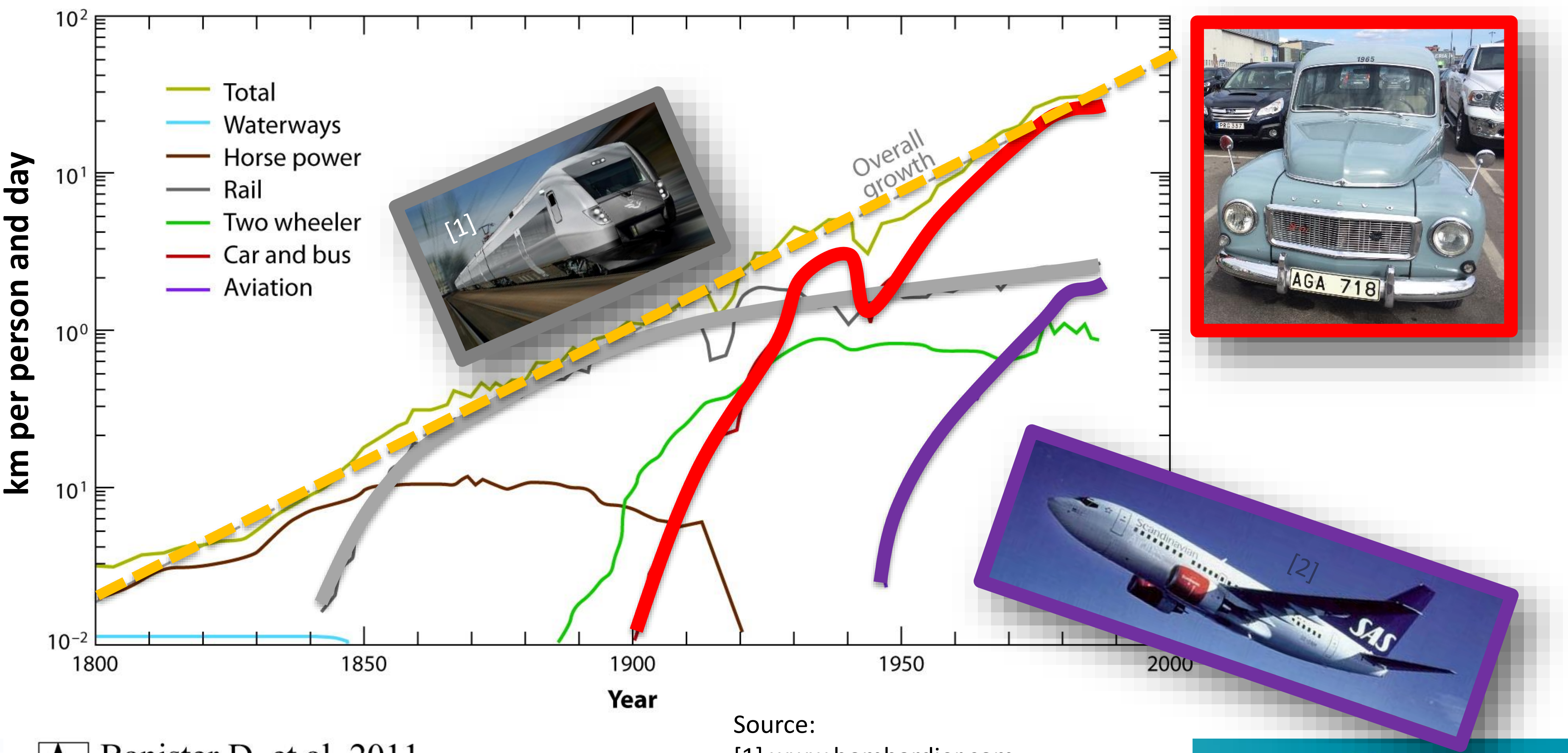


ExaFLOW

→ Address current **algorithmic** bottlenecks for **exascale** to enable the use of **accurate** CFD codes for problems of **practical engineering** interest

Why CFD?

Skin friction/drag reduction is the key for economically and ecologically more efficient transport

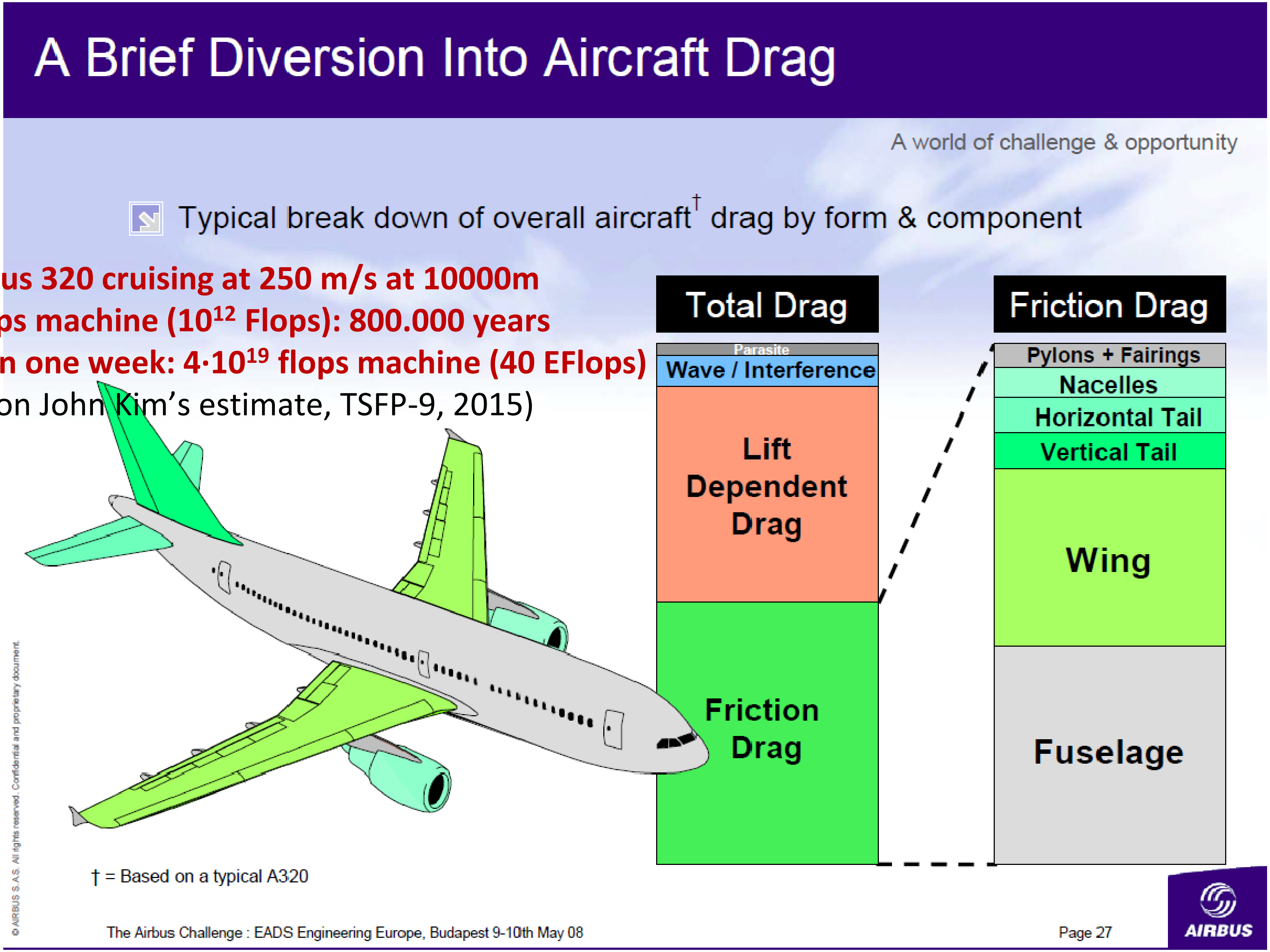


Banister D, et al. 2011.
Annu Rev. Environ. Resour. 36:247–70

Source:
[1] www.bombardier.com
[2] www.flysas.de

SERC
Swedish e-Science Research Centre

The Airbus Challenge, David Hills, 2008



A Brief Diversion Into Aircraft Drag

A world of challenge & opportunity

- Typical break down of overall aircraft[†] drag by form & component

An Airbus 320 cruising at 250 m/s at 10000m
Teraflops machine (10^{12} Flops): 800.000 years
Result in one week: $4\cdot10^{19}$ flops machine (40 EFlops)
(based on John Kim’s estimate, TSFP-9, 2015)

Total Drag

Parasite
Wave / Interference
Lift Dependent Drag
Friction Drag

Friction Drag

Pylons + Fairings
Nacelles
Horizontal Tail
Vertical Tail
Wing
Fuselage

† = Based on a typical A320

APS-DFD Gallery of Fluid Motion 2015

Entry #: V0078

APS Gallery of Fluid Motion 2015

Turbulent flow around a wing profile, a direct numerical simulation

Mohammad Hosseini, Ricardo Vinuesa, Ardeshir Hanifi
Dan Henningson, and Philipp Schlatter

Linné FLOW Centre
and
Swedish e-Science Research Centre (SeRC)
KTH Mechanics, Stockholm, Sweden

https://www.youtube.com/watch?v=hz7UjN_vYuw

Swedish e-Science Research Centre

Navier – Stokes equations

Data from Mira (2013), million core hours

Engineering/CFD	525	19%
Subsurface flow & reactive transport	80	3%
Combustion	100	4%
Climate	280	10%
Astrophysics	28	1%
Supernovae	105	4%
	1118	**40%**

(fraction of Navier-Stokes based simulation on current supercomputers)

ExaFLOW Overview (Oct. 2015-Oct. 2018)

The main goal of the project is to address current algorithmic bottlenecks to enable the use of accurate CFD codes for problems of practical engineering interest. The focus will be on different simulation aspects including:

- Accurate **error control** and **adaptive mesh refinement** in complex computational domains
- **Solver efficiency** via mixed discontinuous and continuous Galerkin methods and appropriate optimized preconditioners
- **Heterogenous modeling** to allow for different solution algorithms in different zones of a domain
- Strategies to ensure **fault tolerance** and resilience
- Parallel **input/output** for extreme data, employing novel data reduction algorithms (feature-based in-situ analysis)
- **Energy awareness** of high-order methods
- **Academic** and **industrial** uses cases to drive development

ExaFLOW Partners

- KTH Stockholm, PDC and Mechanics (Coordinator, SWEDEN)
- Imperial College, London, Aeronautics (UK)
- University of Southampton, Aerodynamics (UK)
- University of Edinburgh, EPCC (UK)
- University of Stuttgart, HLRS and Aerodynamics (GERMANY)
- EPF Lausanne, Mathematics (SWITZERLAND)

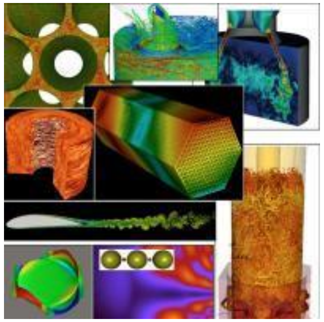
- McLaren Racing (UK)
- Automotive Simulation Center Stuttgart (GERMANY)

Regular Meetings

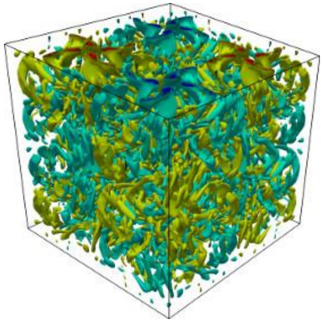
ExaFLOW Overview

The algorithms developed by the project are prototyped in open-source **high-order CFD codes**

- **Nek5000**
  - Spectral element
  - Fortran 77 + C
  - Hexahedra or Quadrilateral elements
- **Nektar++**
  - Spectral/HP element
  - C++
  - Various elements (mixed)
- **OpenSBLI**
  - Finite difference
  - Python
  - Structured Multi-block



NEKTAR++
SPECTRAL/HP ELEMENT FRAMEWORK



Why is exascale (CFD) hard?

- 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

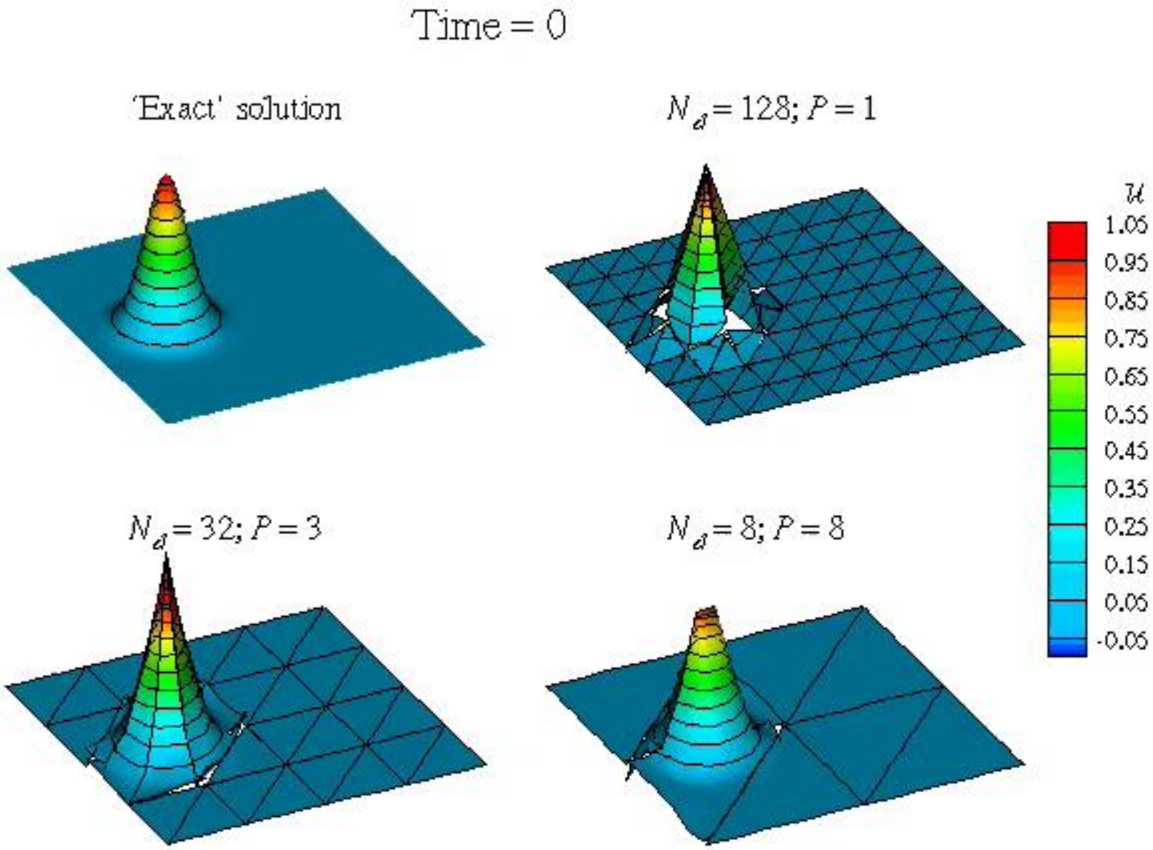
Main problem (asides from communication):

- Complicated memory hierarchies
- Very limited memory bandwidth

Therefore need algorithms with **high arithmetic intensities** that can actually use FLOPS available

Why high order?

- Higher p (vs h) means more work per core:



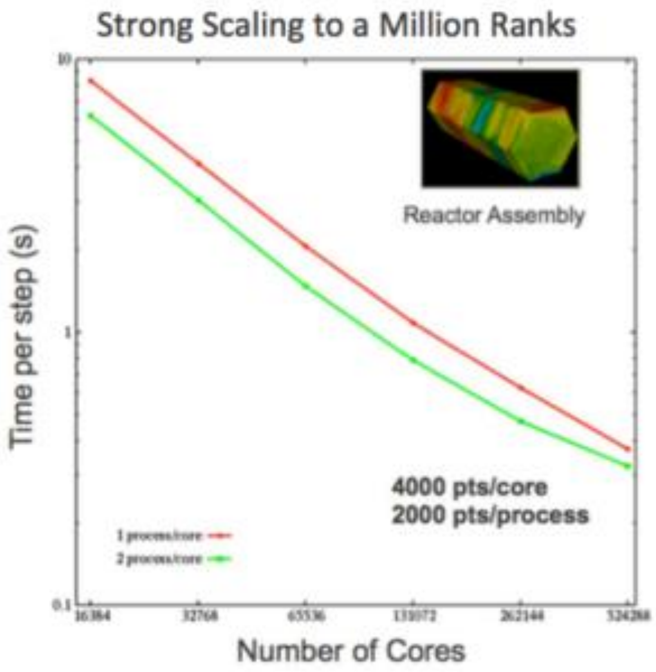
From David Moxey, Univ. Exeter

SEM (and Nek5000)

- High-order accuracy at low cost
  - rapid (exponential) convergence in space
  - Low numerical viscosity
- Highly scalable
  - Fast scalable coarse-grid solvers (AMG, XX^T)
  - Scales up to 1,000,000 processes on BGQ
- “Issues”
  - Meshing potentially difficult
  - Absence of numerical viscosity, some type of stabilization is generally required (filtering)

- Domain partitioned into high-order quadrilateral elements
- Loosely coupled elements (C0 continuity between elements)
- Iterative solvers imply local work with dense operators, followed by data exchanges to update interface

- 2.95M spectral elements with polynomial order N = 9;
- 524,288 cores on ALCF BG/Q Mira;
- Full scale parallel efficiency 0.6
  - two processes/core
  - 2000 points/rank



Can we go to exa-scale with SEM?

- Number of grid points N per processor important, local work has to outweigh cost for communication

- For Nek5000 on BG/P: $(N/P) \sim 1000 - 10{,}000$ sufficient

➔ **$\sim 10^{12}$ = minimum number of points to scale to $P = 10^{8}$**

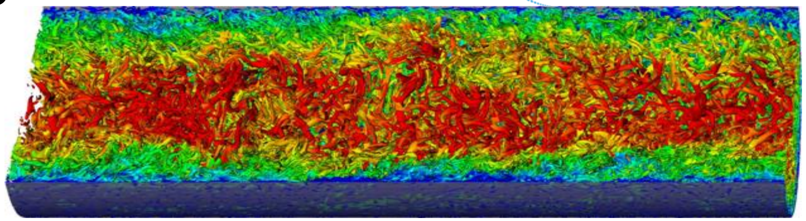
- We must increase problem size for efficient usage of exa-scale, no problem for higher Reynolds numbers

- More work per grid point advantage
  - HOM (Higher Order Methods) such as SEM
  - Multi-physics (magneto-hydrodynamics, combustion, heat transfer)
  - Accelerators (GPU) require more points per processor

- Major bottleneck: (global) pressure calculation!

ExaFLOW Overview

- Five computationally-demanding use cases:
  - NACA4412 (compressible) - Soton
  - NACA4412 (incompressible) - KTH
  - Jet in crossflow - KTH/Stuttgart
  - Automotive use case – Opel/ASCS
  - Front wing – Imperial/McLaren
- Baseline performance
- Feedback to algorithm development



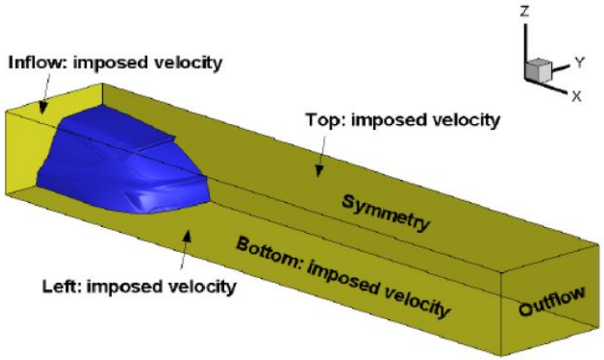
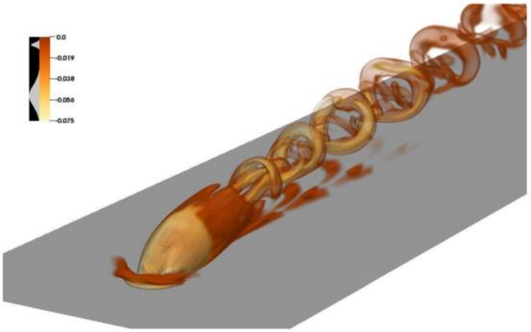
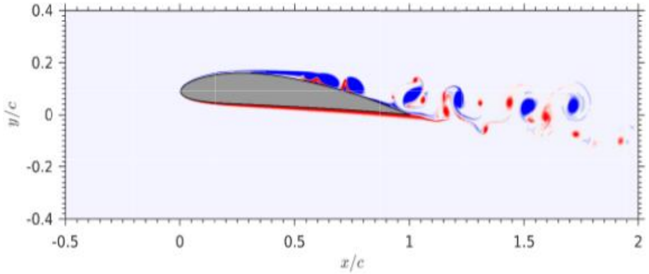
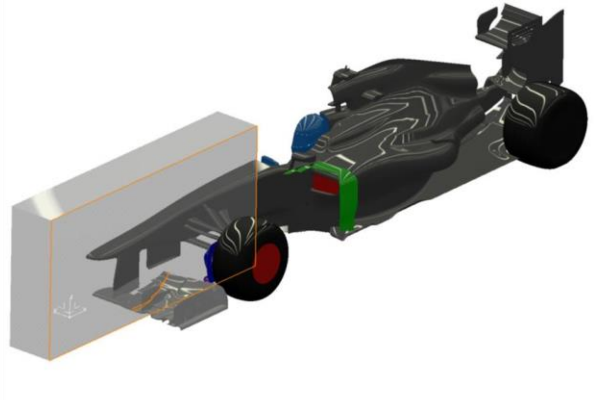


Figure 6: Computational domain of submodel and its boundary conditions.







ExaFLOW Overview (Oct. 2015-Oct. 2018)

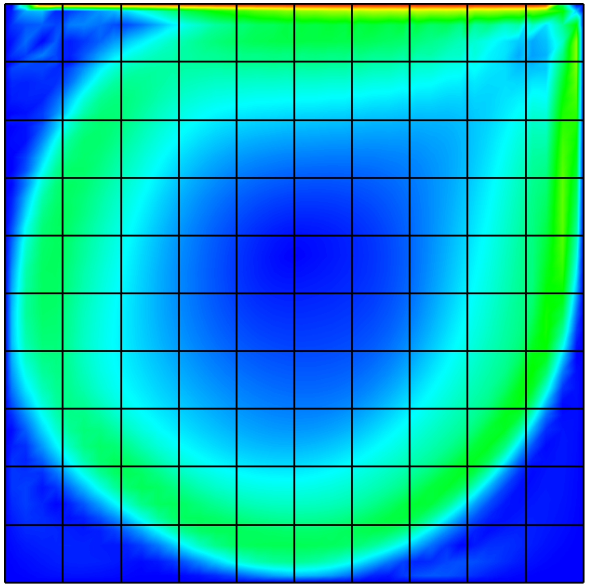
The main goal of the project is to address current algorithmic bottlenecks to enable the use of accurate CFD codes for problems of practical engineering interest. The focus will be on different simulation aspects including:

- Accurate **error control** and **adaptive mesh refinement** in complex computational domains
- **Solver efficiency** via mixed discontinuous and continuous Galerkin methods and appropriate optimized preconditioners
- **Heterogenous modeling** to allow for different solution algorithms in different zones of a domain
- Strategies to ensure **fault tolerance** and resilience
- Parallel **input/output** for extreme data, employing novel data reduction algorithms (feature-based in-situ analysis)
- **Energy awareness** of high-order methods
- **Academic** and **industrial** uses cases to drive development

ExaFLOW’s Strategy towards Exascale

Mesh Quality and Mesh Adaptivity to resolve regions of the flow without *a priori* knowledge of physics

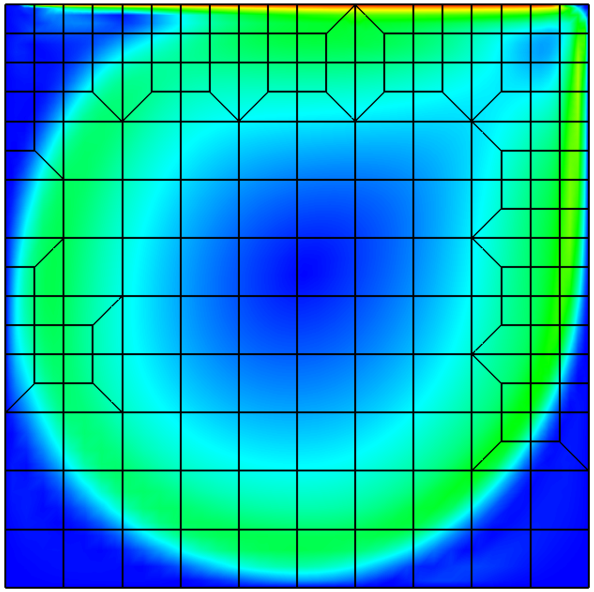
- Adaption in terms of *h* (element size), *p* (polynomial order) and *r* (moving mesh)



original mesh

error indicators

adaption

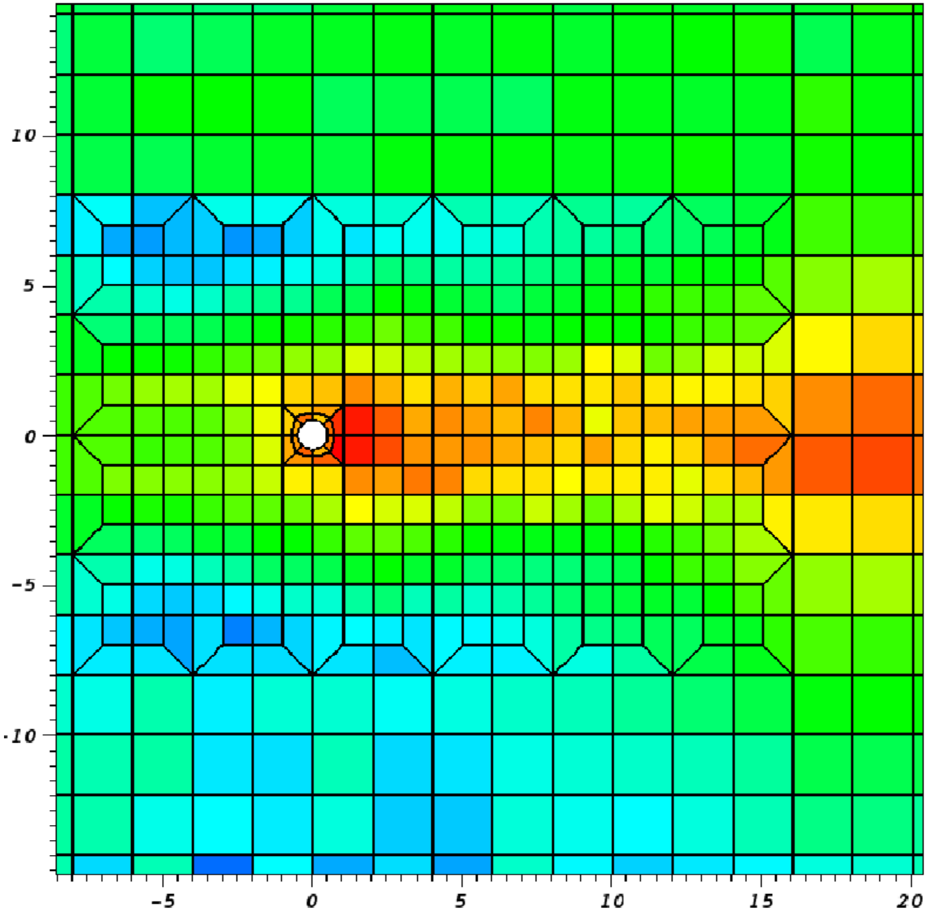


h-refined (conformal) mesh

Example: lid-driven cavity flow

Error indicators (KTH / Nek5000)

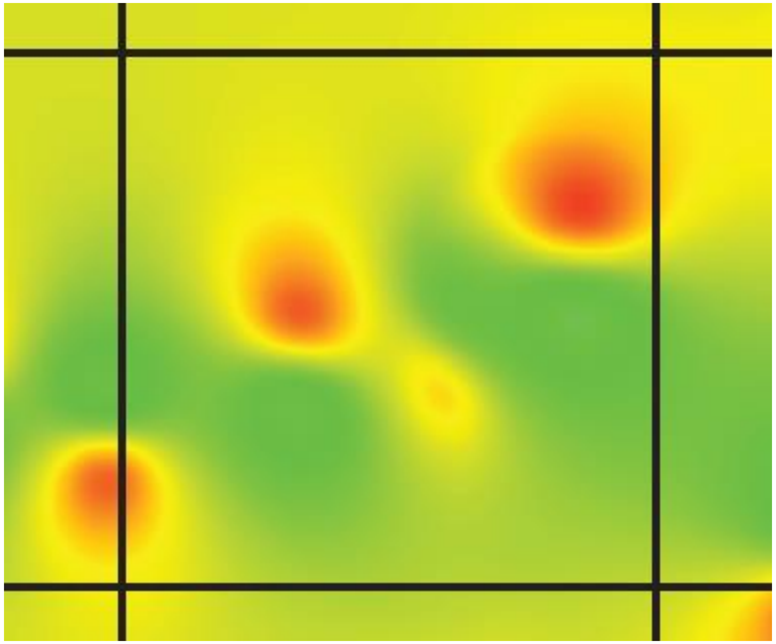
- Classical problem: vortex shedding behind a cylinder
- High-order polynomial inside each element
- Colours indicate error levels from **low** (blue/green) to **high** (yellow/red)
- Error indicator able to detect local regions of inaccuracy, particularly in the wake of the flow
- Can then use this error distribution to drive adaption



Spectral transform

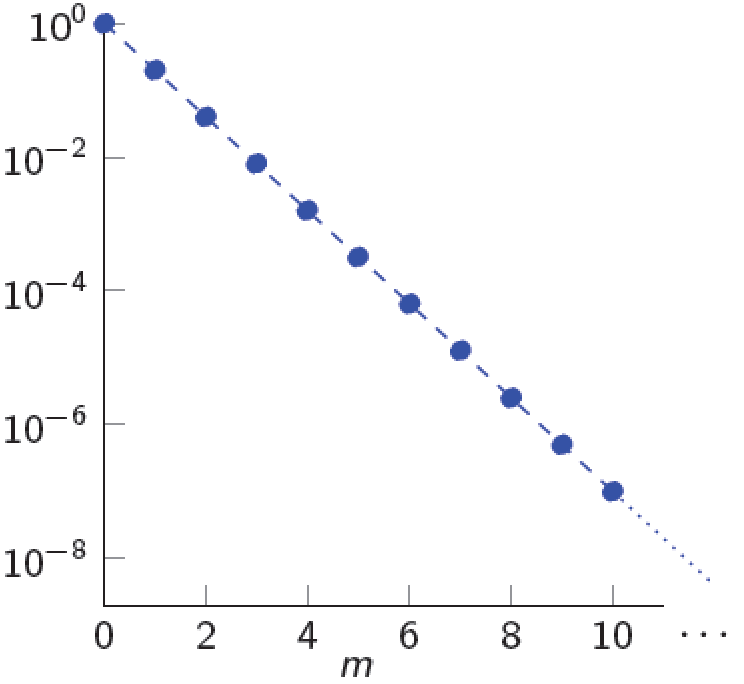
Spatial domain

$$u(x) := \sum_{m=0}^{\infty} \hat{u}_m p_m(x)$$



Spectral domain

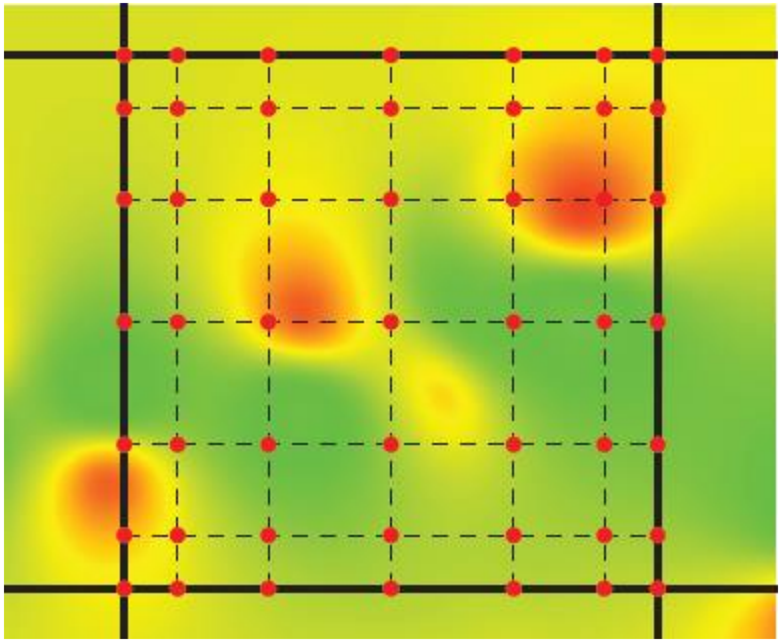
$$\hat{u}_m := \frac{1}{\gamma_m} \int_{-1}^{1} w(x) u(x) p_m(x) dx$$



Truncation error

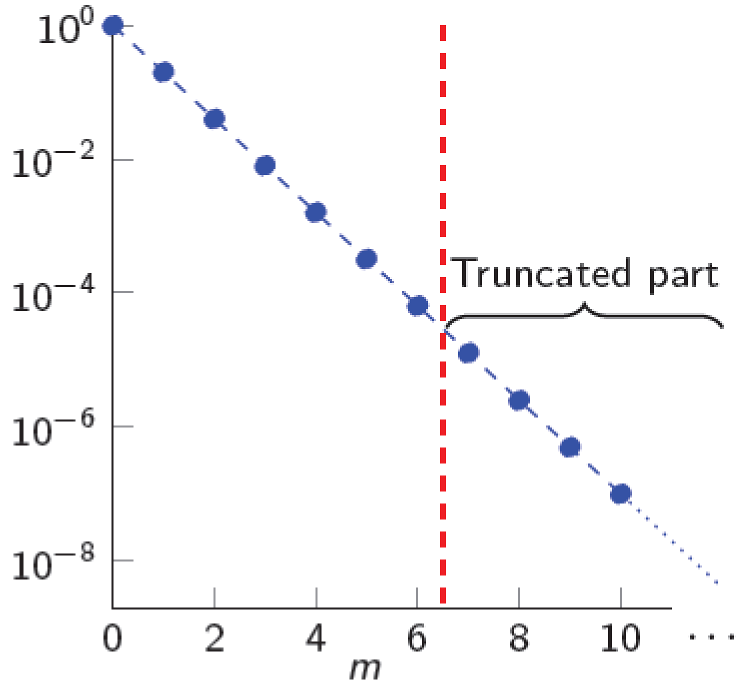
Spatial domain

$$\color{red}{P_N}\color{black}{u(x) := \sum_{m=0}^{\color{red}{N}} \hat{u}_m p_m(x)}$$



Spectral domain

$$\hat{u}_m := \frac{1}{\gamma_m}\int_{-1}^{1} w(x)u(x)p_m(x)dx$$



Estimate of the truncation error

Assumed exponential decay

$$\hat{u}_m \sim c\, e^{-\sigma m}$$

Interpolation for $m = 0, \ldots, N$.

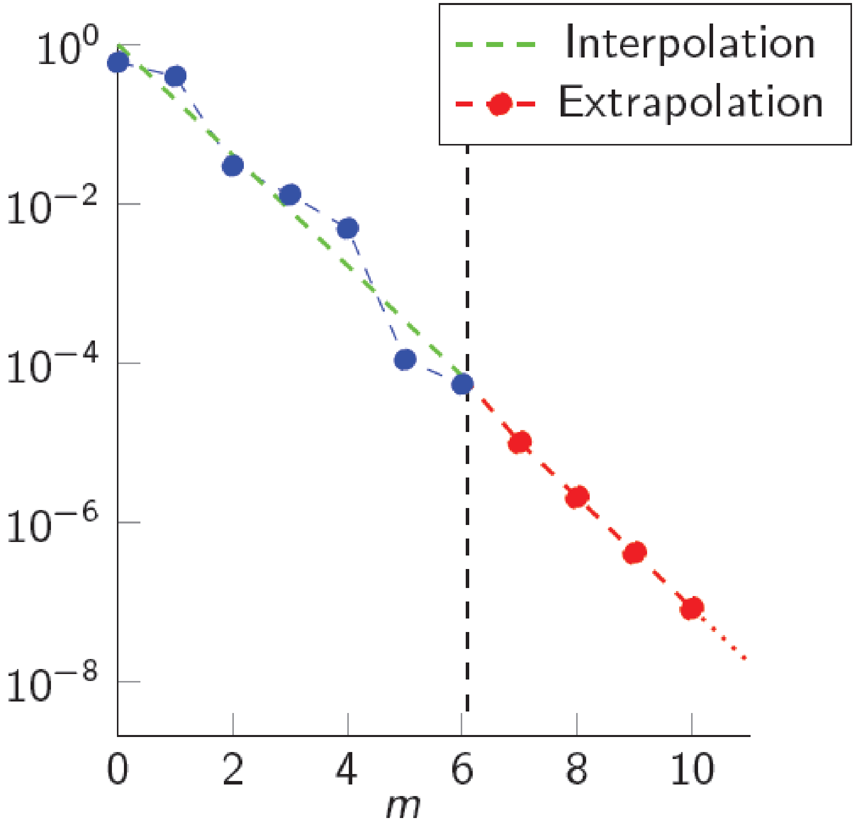
Best fit for c and σ.

Extrapolation for $m = N+1, \ldots, \infty$.

Integral under the « tail »

$$\epsilon_t = \left(\sum_{m=N+1}^{\infty} \frac{\hat{u}_m^2}{\frac{2m+1}{2}} \right)^{1/2}$$

$$\approx \left(\int_{N+1}^{\infty} \frac{c^2 e^{-2\sigma m}}{\frac{2m+1}{2}}\, dm \right)^{1/2}$$



Quadrature error

True spectral coefficients

$$\hat{u}_m = \frac{1}{\gamma_m} \int_{-1}^{1} w(x) u(x) p_m(x) dx$$

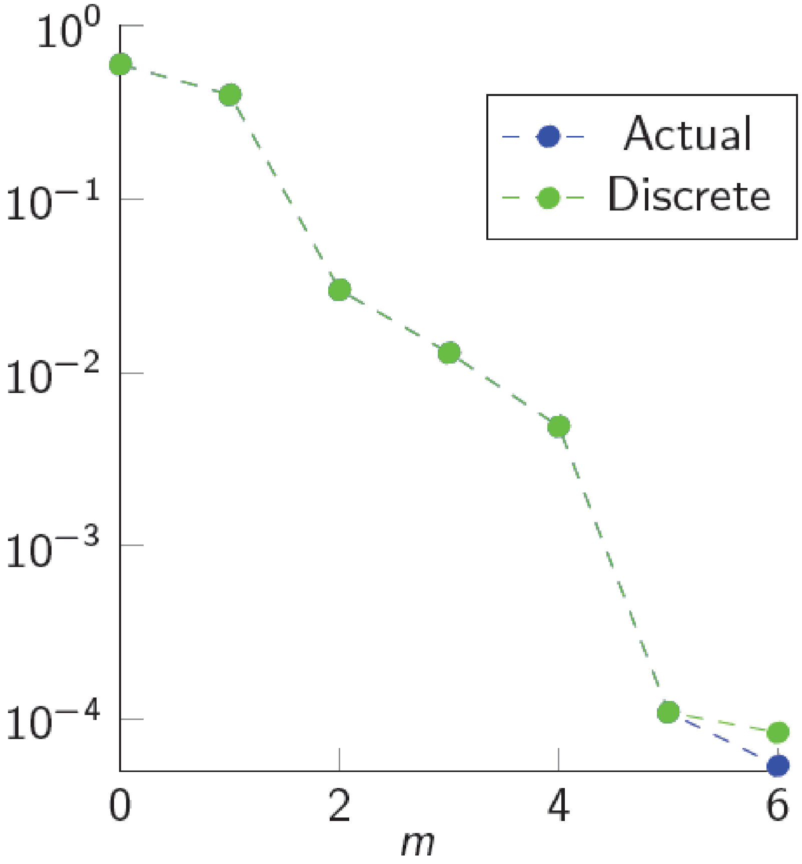
Discrete spectral coefficients

$$\tilde{u}_m = \frac{1}{\gamma_m} \sum_{i=0}^{N} \rho_i u(\xi_i) p_m(\xi_i)$$

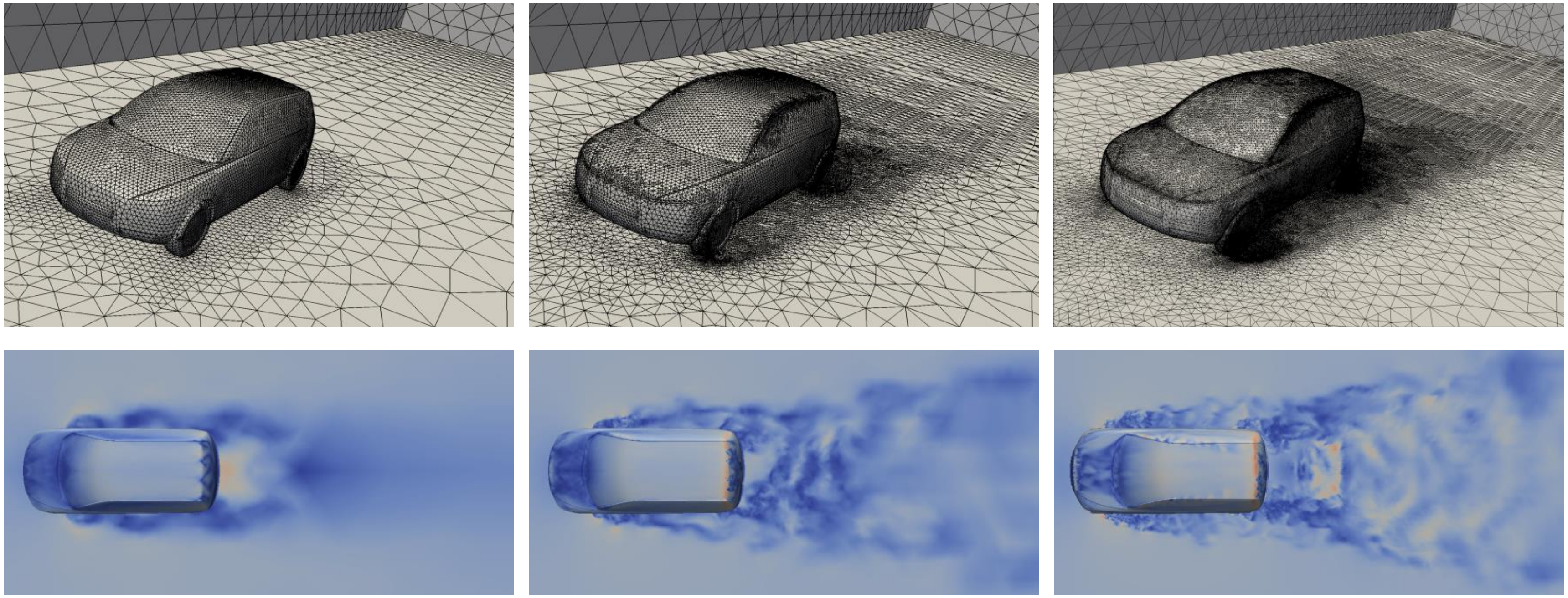
Estimated quadrature error

$$\epsilon_q = \left(\sum_{m=0}^{N} \frac{(\hat{u}_m - \tilde{u}_m)^2}{\frac{2m+1}{2}} \right)^{1/2}$$

$$\approx \left(\frac{\tilde{u}_N{}^2}{\frac{2N+1}{2}} \right)^{1/2}$$



Adjoint based goal driven adaptivity



$$|M(\hat{u}) - M(\hat{U})| \le \sum_{n=1}^{N} \Big[\int_{I_n} \sum_{K\in\mathcal{T}_n} |R_1(\hat{U})|_K \cdot \omega_1 \, dt + \int_{I_n} \sum_{K\in\mathcal{T}_n} |R_2(U)|_K \, \omega_2 \, dt \Big]$$

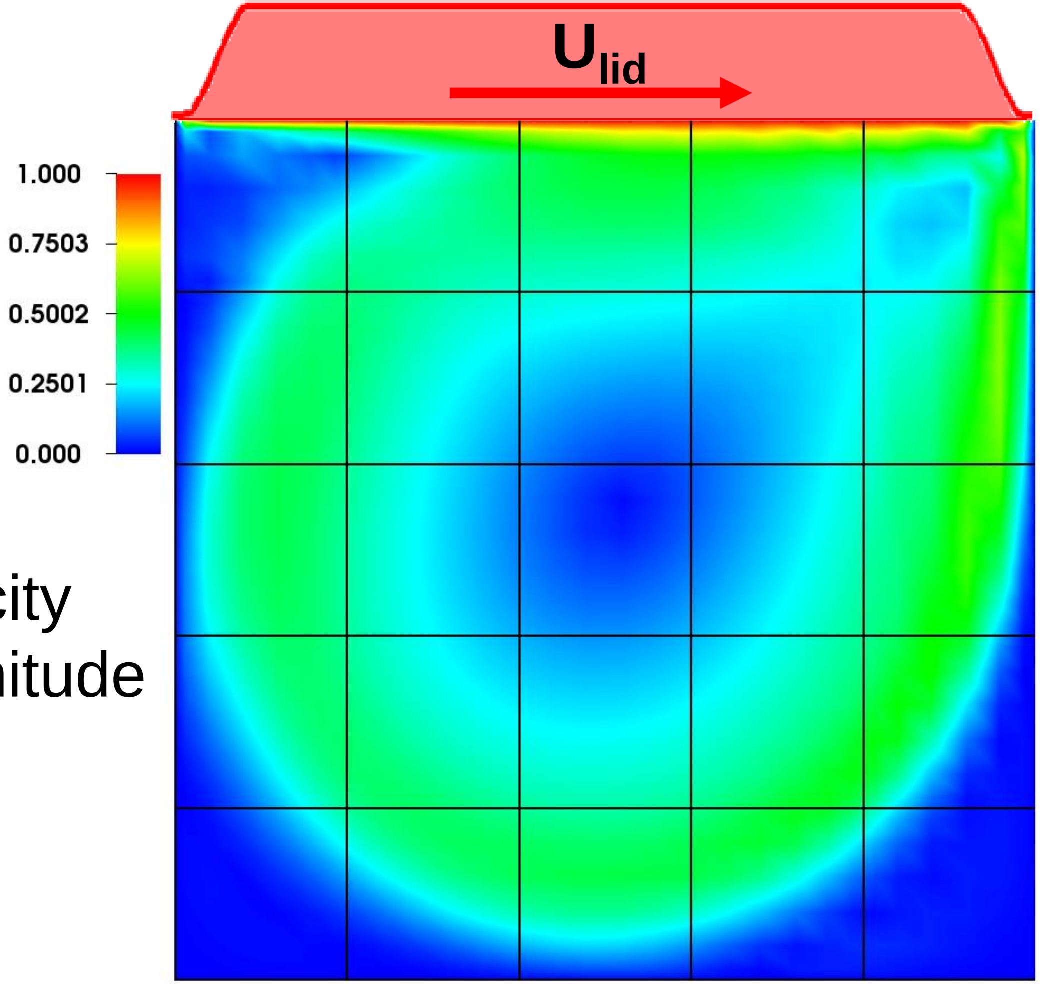
$$R_1(\hat{U}) = \dot{U} + (U\cdot\nabla)U + \nabla P - \nu\Delta U - f$$

$$R_2(U) = \nabla \cdot U$$

Hoffman et al., FeNICS

First AMR test case

- 2D lid-driven cavity (Re = 2000, steady)

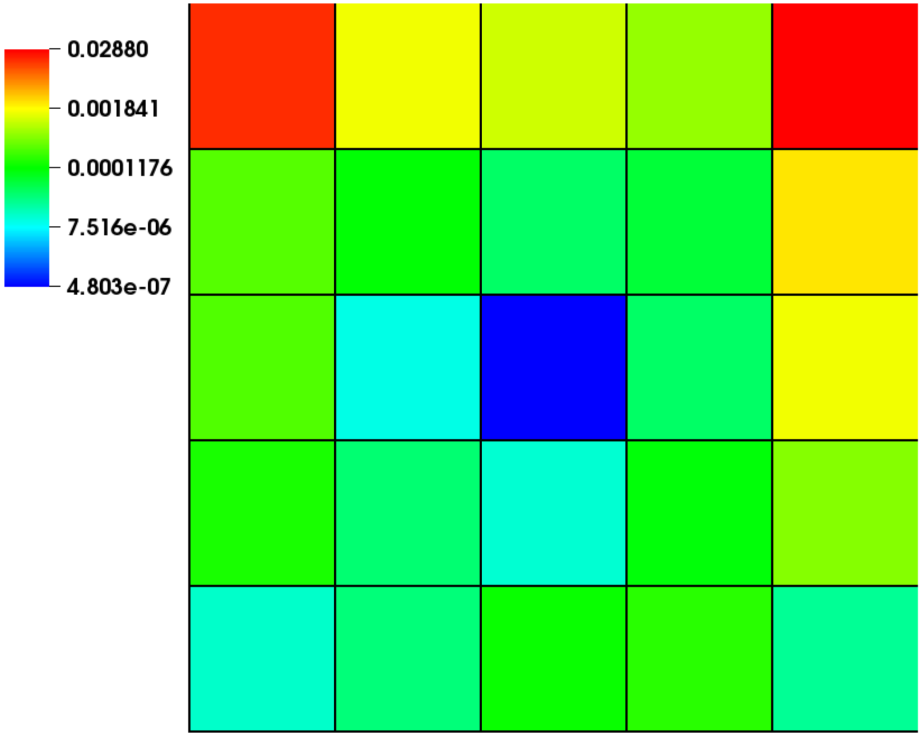


Quantity
of interest:
Drag on the lid

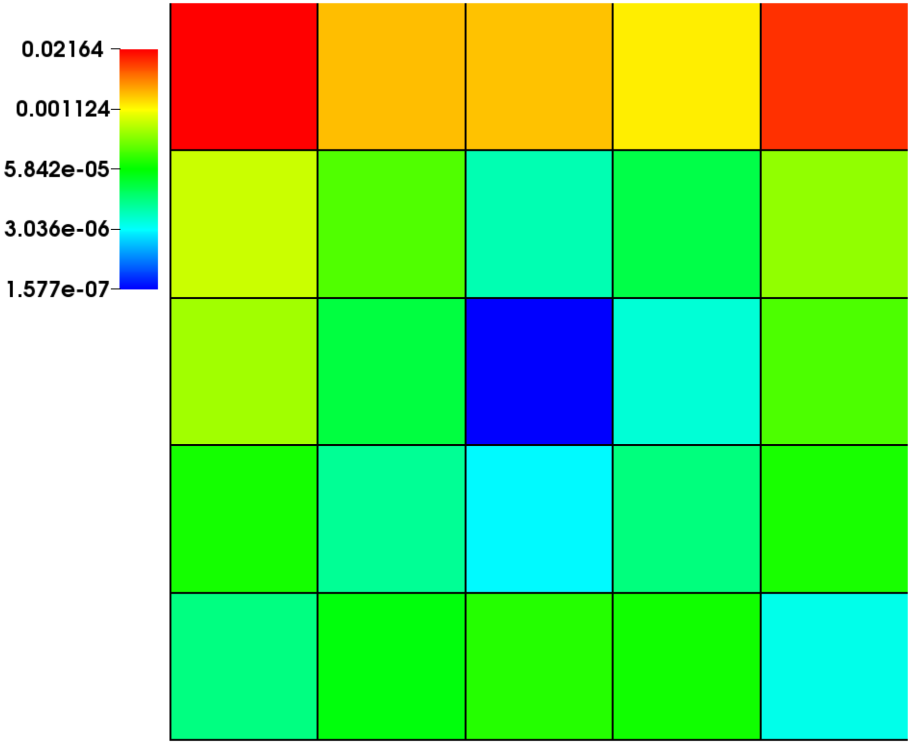
Velocity
magnitude

Initial estimators

Spectral err. ind.



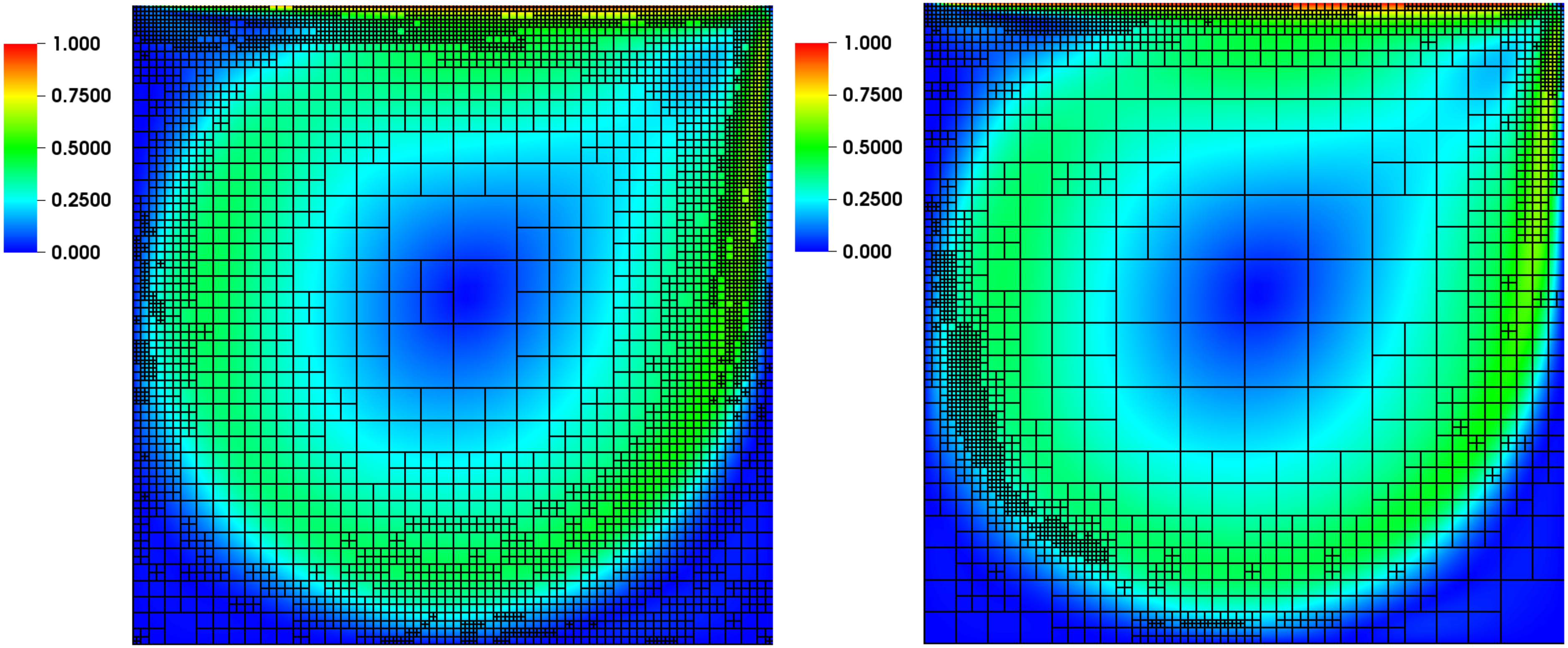
Adjoint err. est.



Final meshes

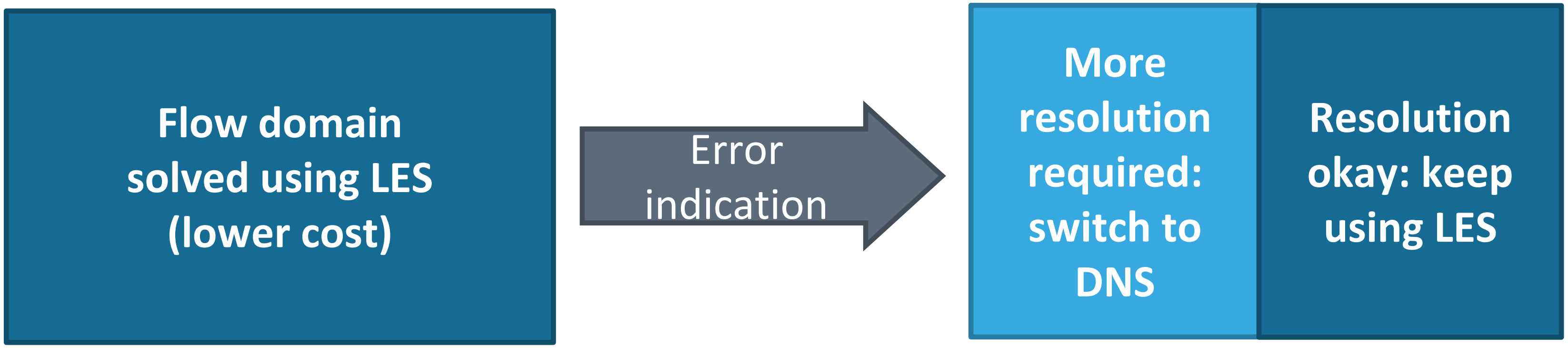
Spectral err. ind.

Adjoint err. est.



ExaFLOW’s Strategy towards Exascale

- **Heterogeneous modelling** will allow efficient use of the **heterogeneous resources** expected in an exascale machine by modelling regions of the flow with different models



Flow domain solved using LES (lower cost)

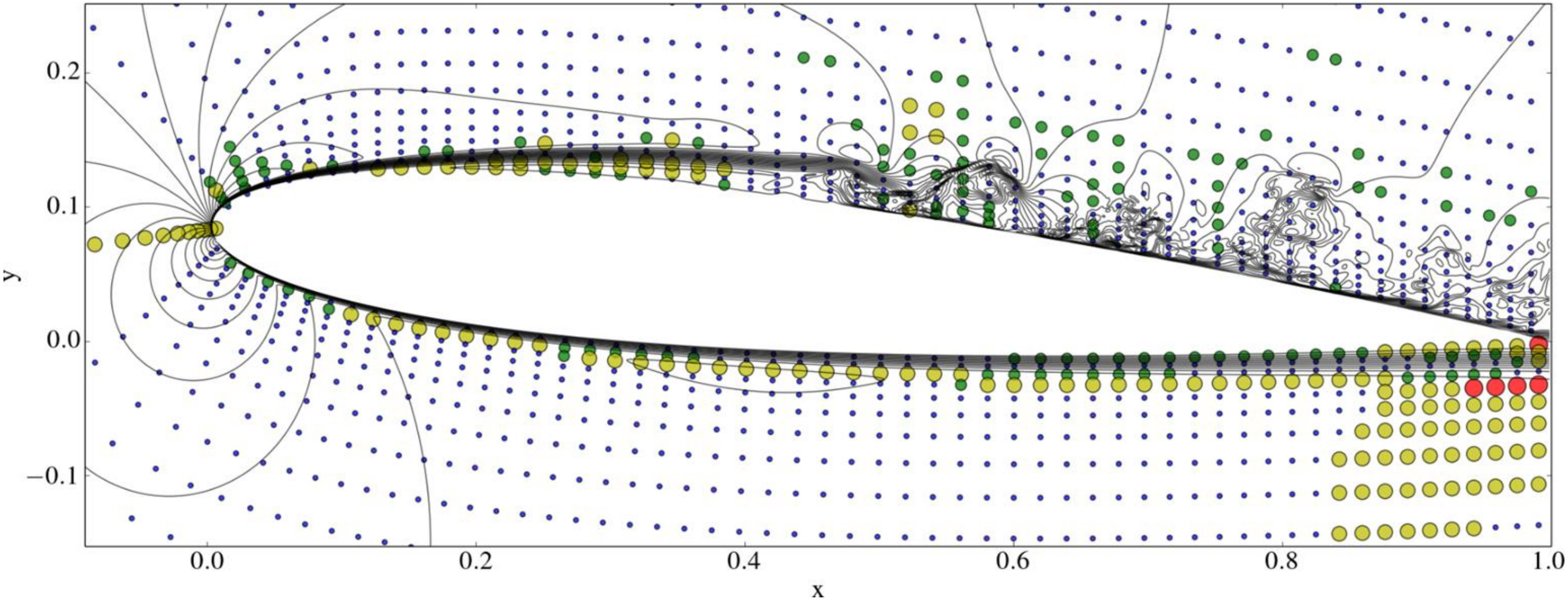
Error indication

More resolution required: switch to DNS

Resolution okay: keep using LES

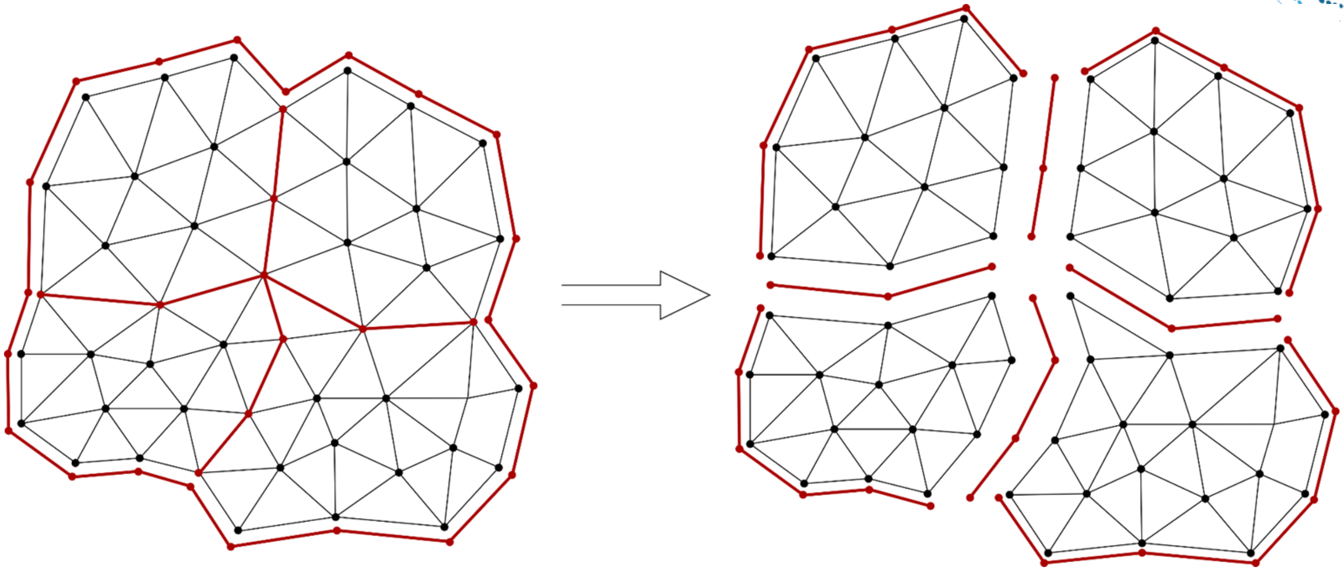
- Most work to date has focused on error control aspect

Heterogeneous modelling & error control



- Uniform flow away from the aerofoil is well-resolved. High error severity along the boundary layers either side of the aerofoil. Also near the trailing edge where eddy shedding occurs.

ExaFLOW's Strategy towards Exascale

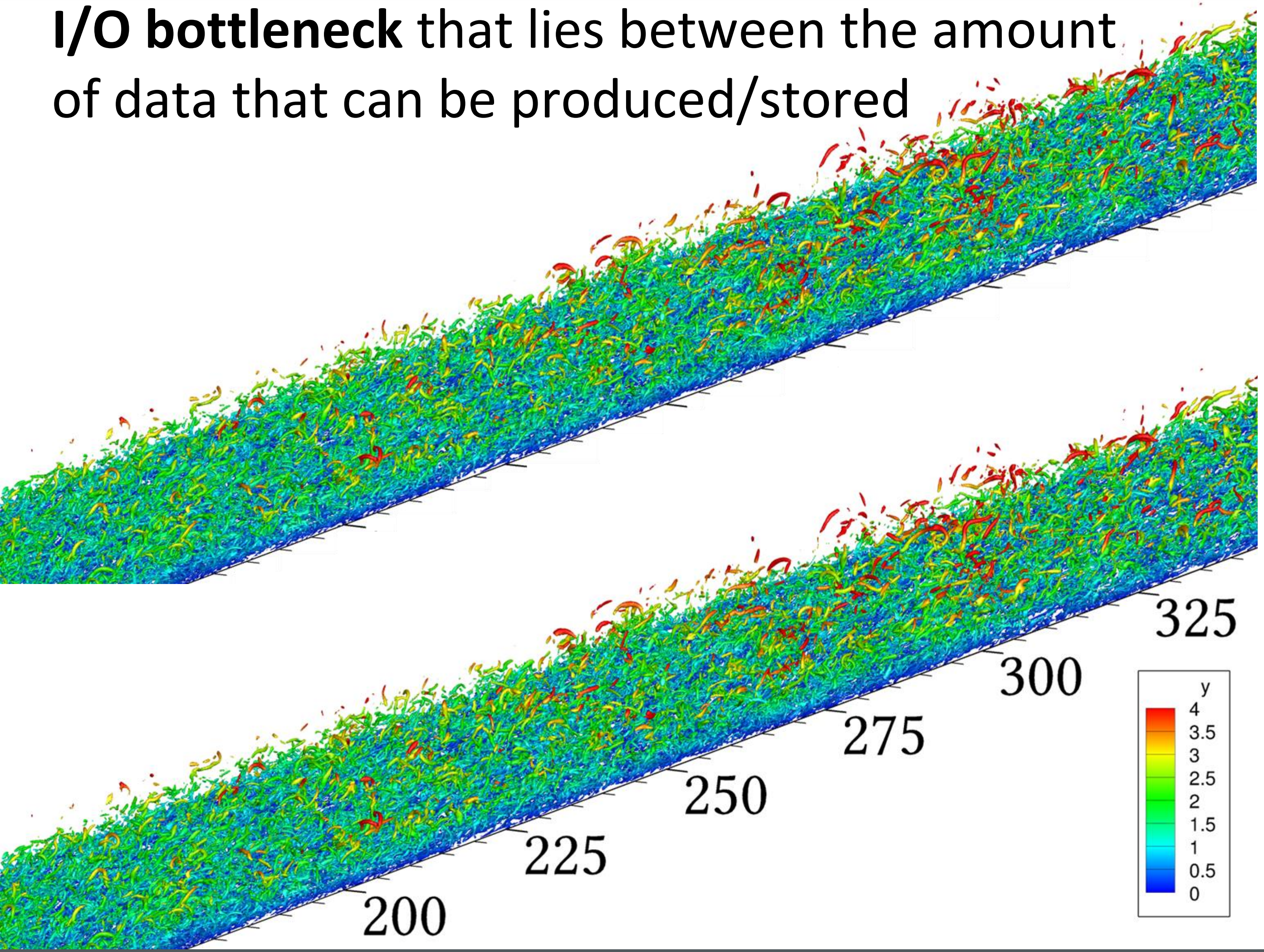


Combine the advantages of **CG** and **HDG**:

- Each node uses **CG** on the interior (efficient)
- Inter-node coupling through **HDG** (good comms)
- Using **spectral element method**, with a polynomial order *p* inside each element

ExaFLOW's Strategy towards Exascale

Data compression techniques will widen the **I/O bottleneck** that lies between the amount of data that can be produced/stored



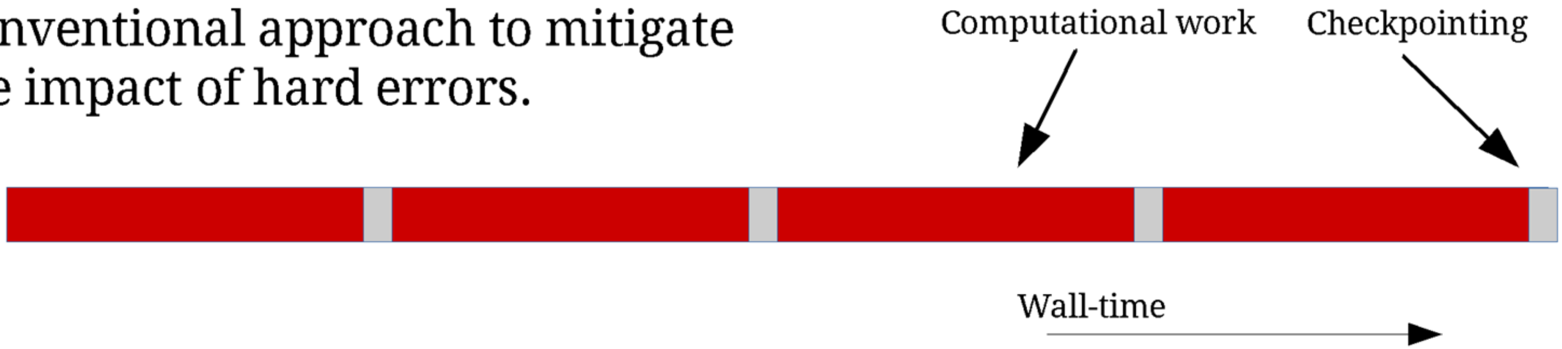
Original
Contours of
λ_2-criterion

Compressed
Compression ratio
17:1

ExaFLOW's Strategy towards Exascale

Fault tolerance strategies address the issue of **resilience**

Conventional approach to mitigate the impact of hard errors.

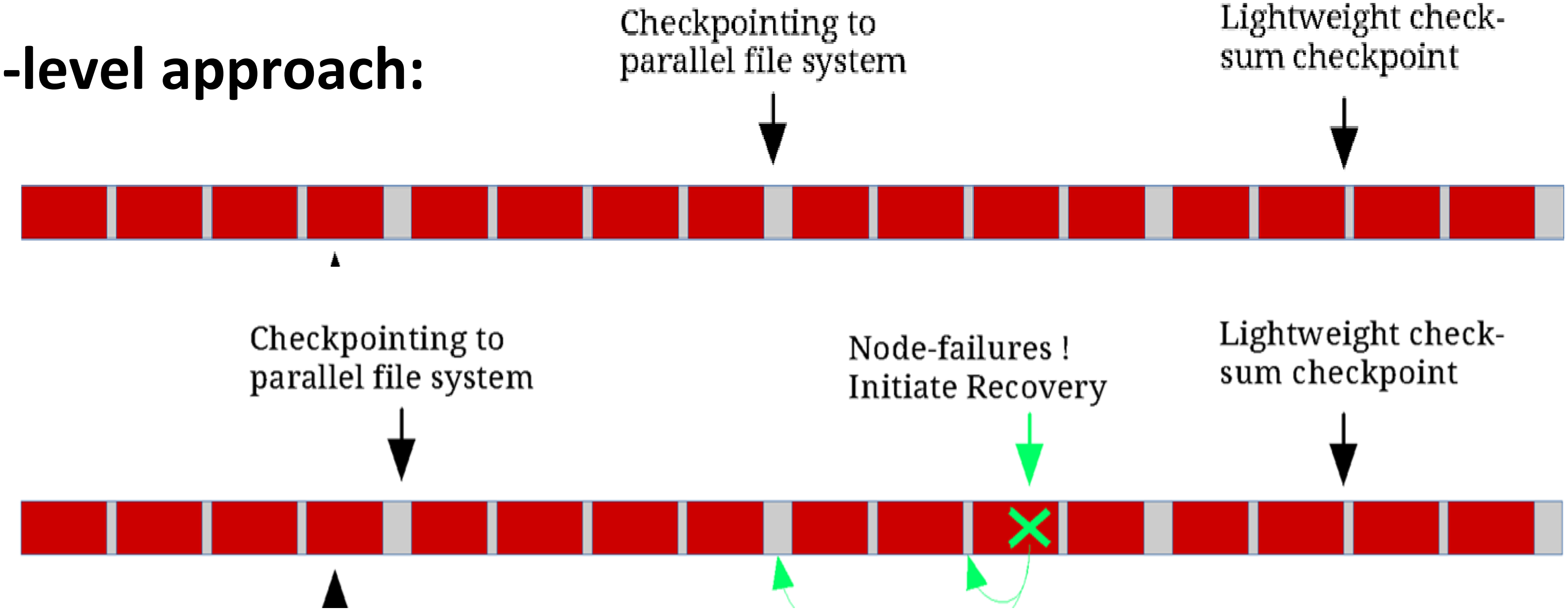


Computational work

Checkpointing

Wall-time

Multi-level approach:



Checkpointing to parallel file system

Lightweight check-sum checkpoint

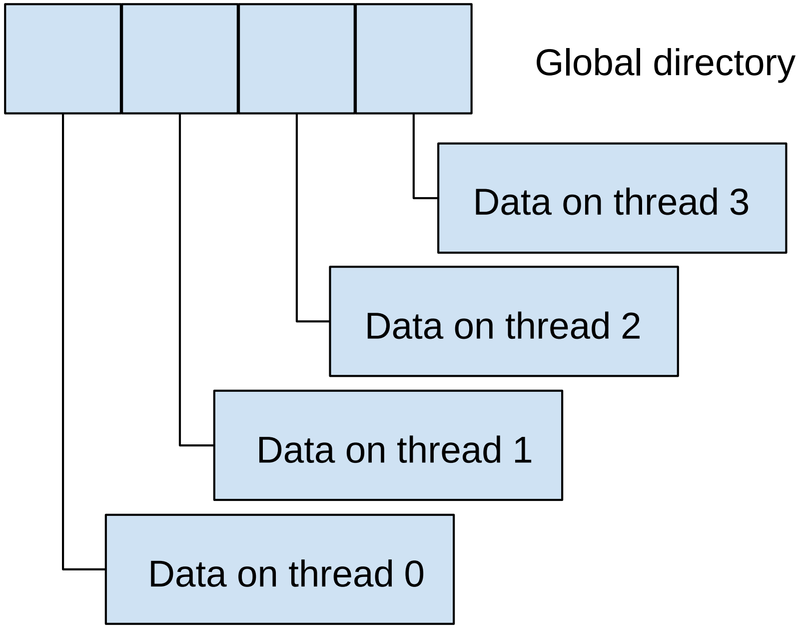
Checkpointing to parallel file system

Node-failures ! Initiate Recovery

Lightweight check-sum checkpoint

ExaFLOW's Strategy towards Exascale

- **Efficient implementations** (including right choice of programming model) are needed on exascale systems



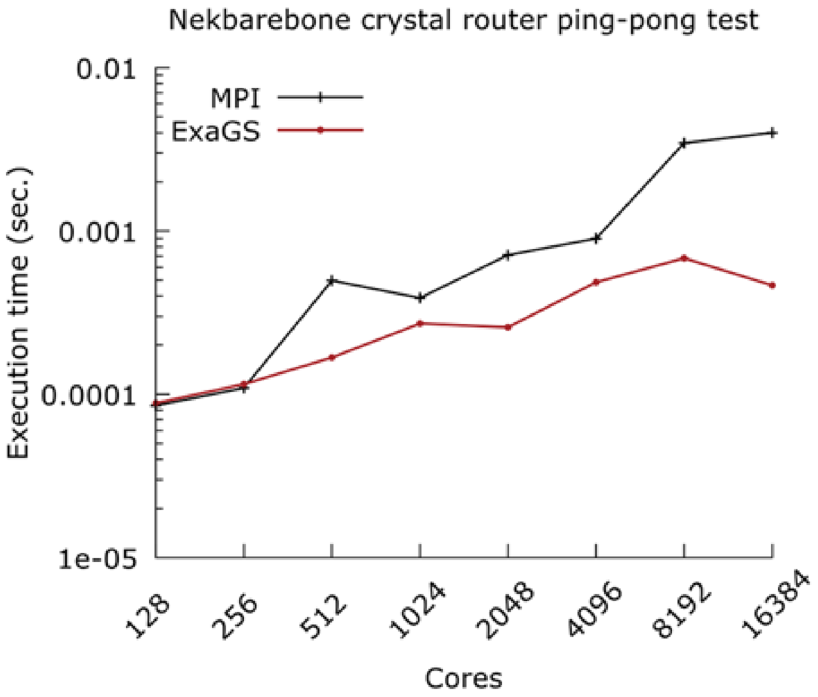
Global directory

Data on thread 3

Data on thread 2

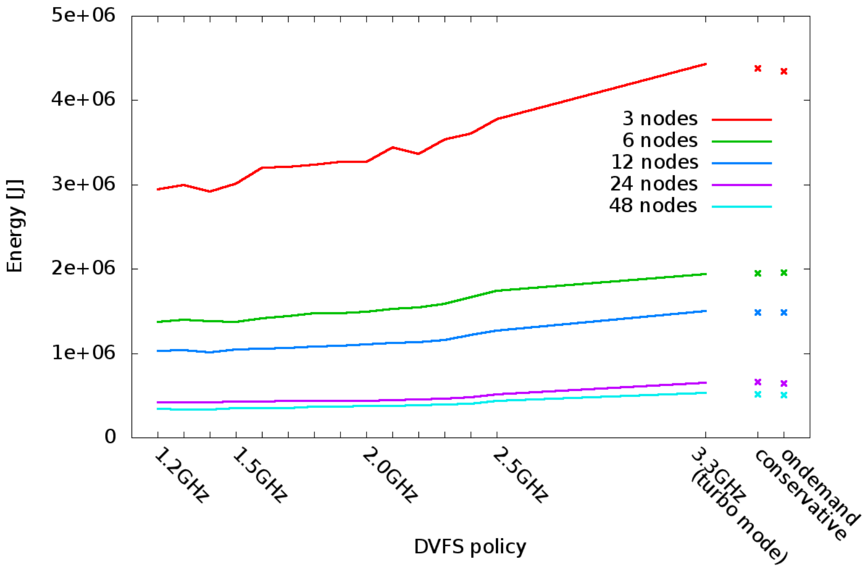
Data on thread 1

Data on thread 0

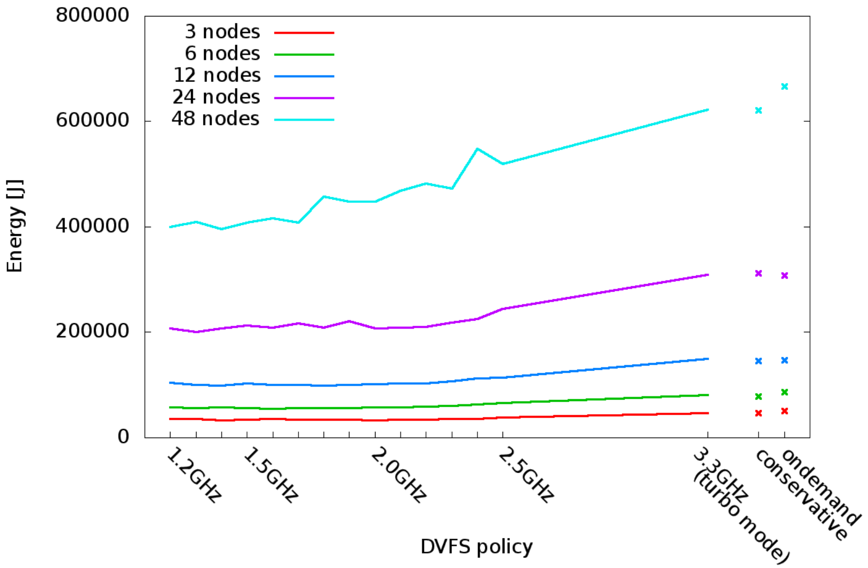


ExaFLOW's Strategy towards Exascale

- Understanding **energy consumption** will be essential for such large machines



Computational phase



I/O phase (HDF5)

Selected „Mini Projects“

- **AMG**
  - Investigate different strategies for AMG setup, and usability for Nek5000 and Nektar++
- **ExaGS**
  - Develop a new low-latency gather-scatter library for Nek5000 and Nektar++, investigating different programming models & techniques
- **ExaArch**
  - Feasibility study and technology watch on exascale architectures (e.g. accelerators) and their impact on the ExaFLOW project
- **Performance comparison Nek5000 – Nektar++**
  - Study on the performance characteristics of Nek5000 and Nektar++, in particular communication patterns, preconditioners

Conclusions

- Fluid mechanics is a **prime example** for exascale
- ExaFLOW will address some of the issues when it comes to **practical applications**
  - error control and adaptive meshing for larger and more complex simulation domains; capable of dynamic remeshing if necessary.
  - Solver efficiency, numerical methods and modelling
  - Heterogeneous modelling
  - Resilience & fault tolerance
  - data handling & compression.
- Open-source development enhancing **community codes** (Nek5000, Nektar++, OpenSBLI)

http://exaflow-project.eu