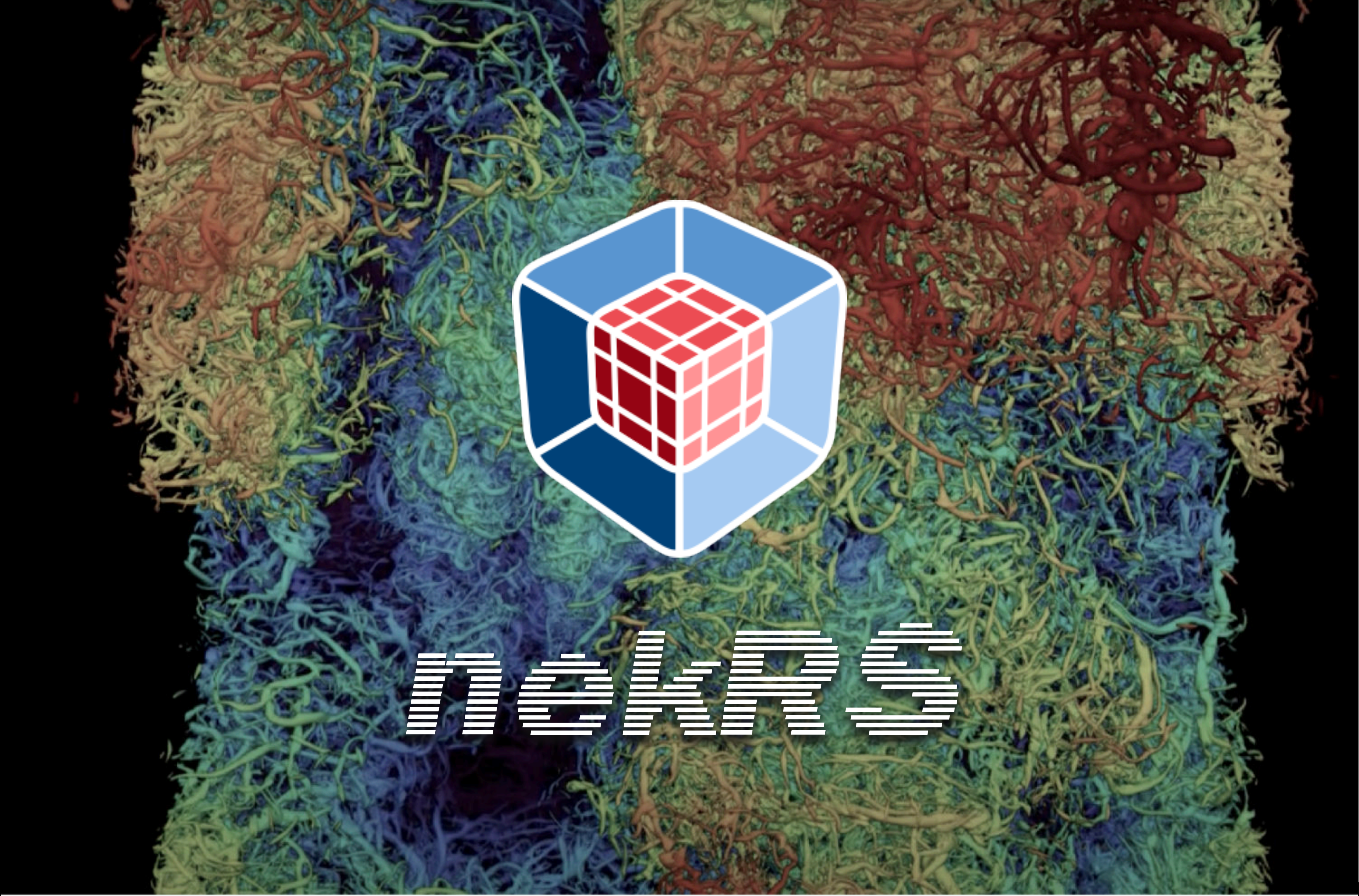


An Exascale CFD Solver

Stefan Kerkemeier (ANL), Mathis Bode (JSC)

Castiel2, Webinar Code of the Month Vol. 3, 2023

START

Mid 1980s at M.I.T (Paul Fischer, Lee Ho and Einar Rønquist) based on the spectral element method (Patera 1984)

MISSION

Pushing the boundaries of fast and accurate CFD to improve predictive modeling

COMMUNITY

500+ users worldwide
~40 publications per year

APPLICATIONS

Thermal hydraulics, renewables, combustion, urban, environmental, aeronautics and many more

TEAM

Cross-functional team of 10 about people

RESEARCH IMPACT

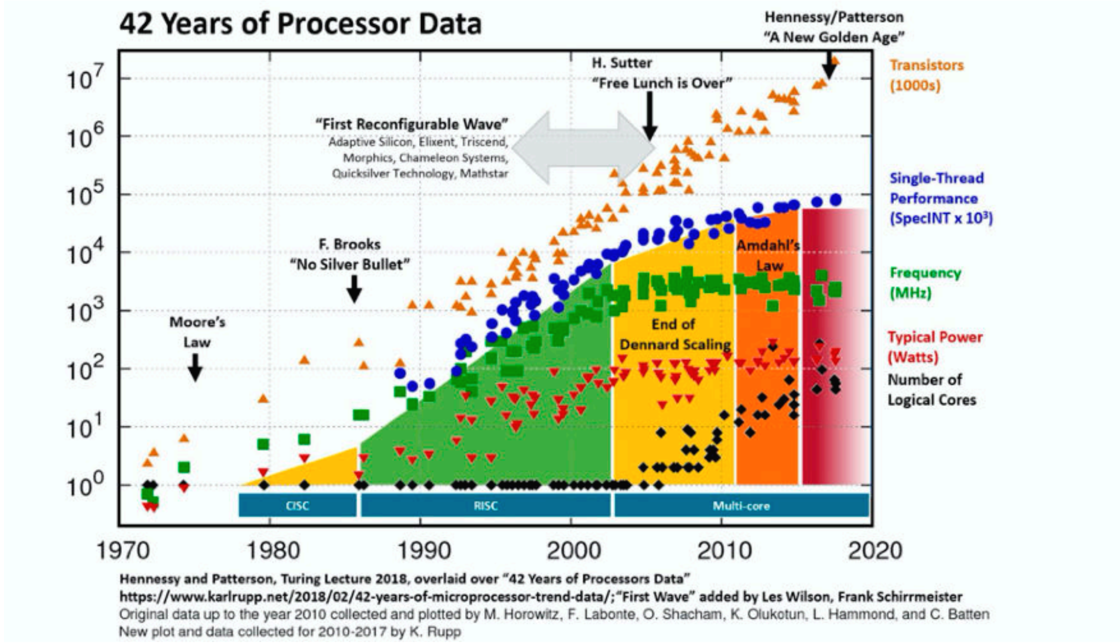
ECP, CEEC, CoEC, EXCELERAAT, PSAAP II, NEAMS, ...

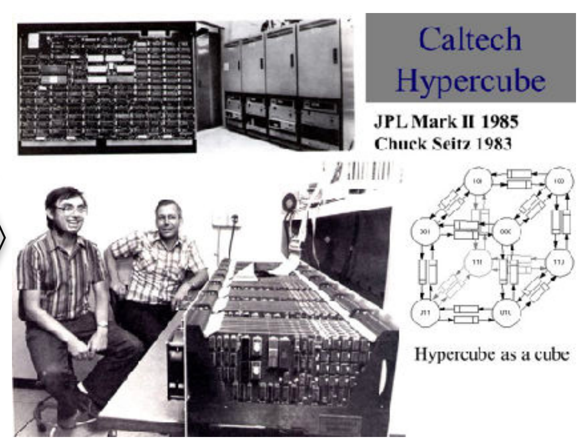
OPEN SOURCE

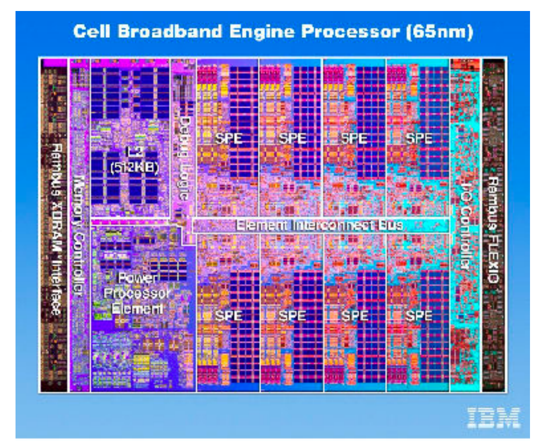
funded by DOE

INDUSTRY COLLABORATION

Kairos, Terrapower, Nuscale, GE, ExxonMobil, ...





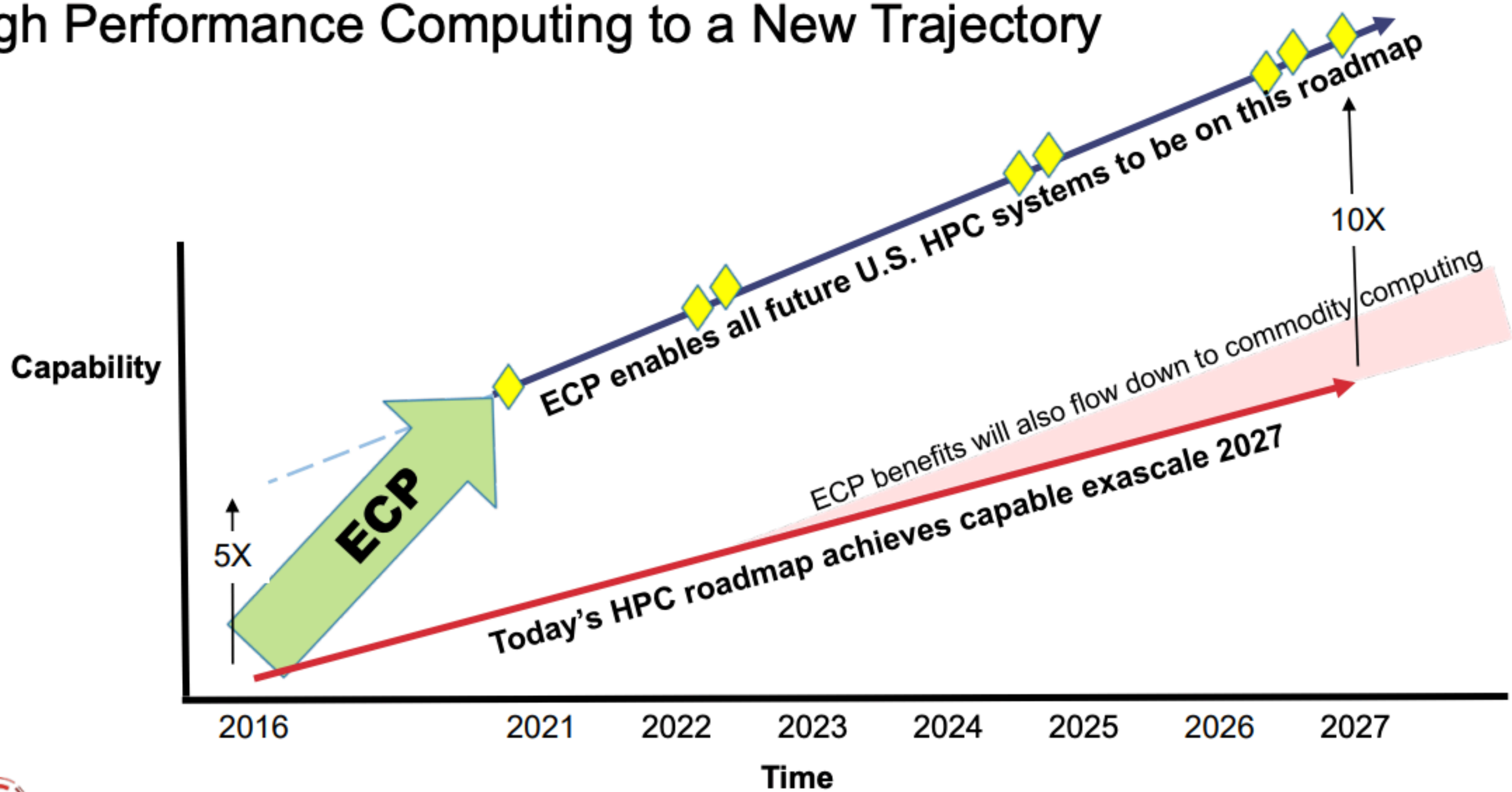


vector

distributed memory

specialization (accelerators)

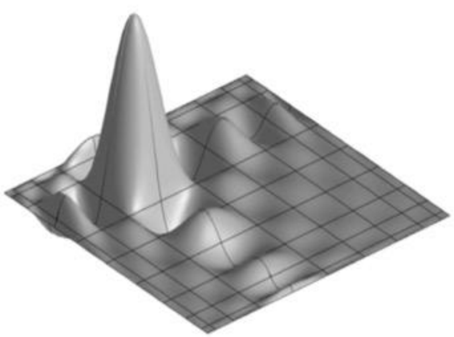
Vision: Exascale Computing Project (ECP) Lifts all U.S. High Performance Computing to a New Trajectory



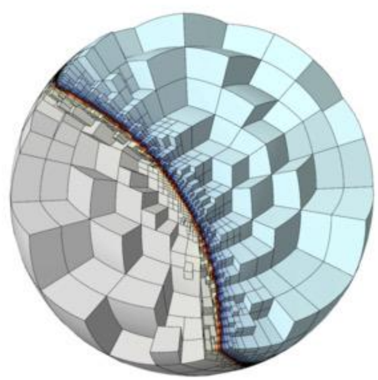
ECP EXASCALE COMPUTING PROJECT

- PDE-based simulations on **unstructured grids**
- **high-order** and **spectral** finite elements

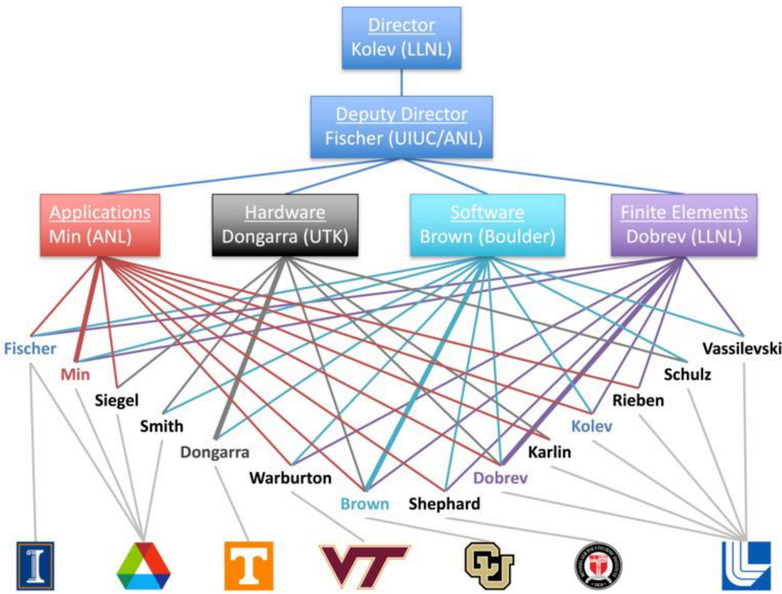
  ✓ *any order space on any order mesh* ✓ *curved meshes,*
  ✓ *unstructured AMR* ✓ *optimized low-order support*



10th order basis function



non-conforming AMR, 2nd order mesh



2 Labs, 5 Universities, 30+ researchers

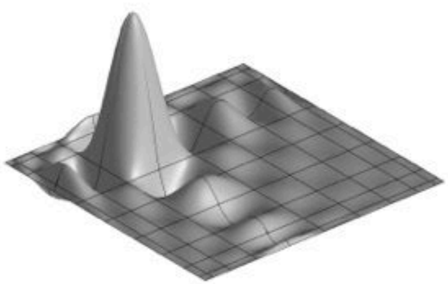
nekRS

- started as an early fork of *libParanumal*
- incompressible + low Mach Navier-Stokes solver
- MPI + (X=OCCA) using CPUs/GPUs
- autotuning across entire solver stack
- high-order spectral elements in space
- overset grids
- moving and deforming meshes
- conjugate heat transfer
- Lagrangian phase model
- LES and RANS models
- available under github.com/Nek5000/nekRS

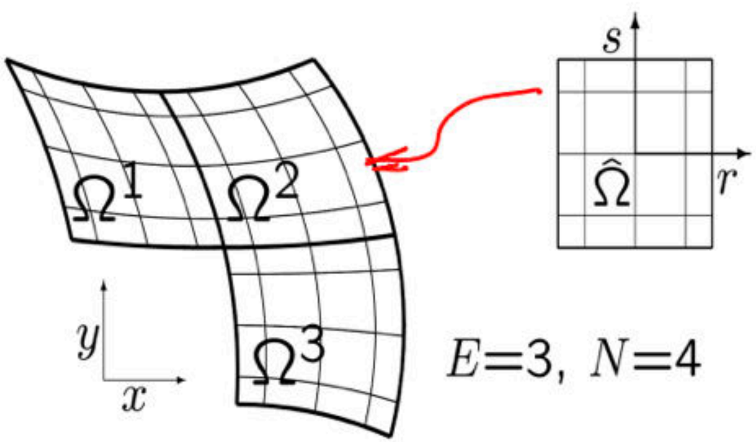
https://www.sciencedirect.com/science/article/abs/pii/S0167819122000710

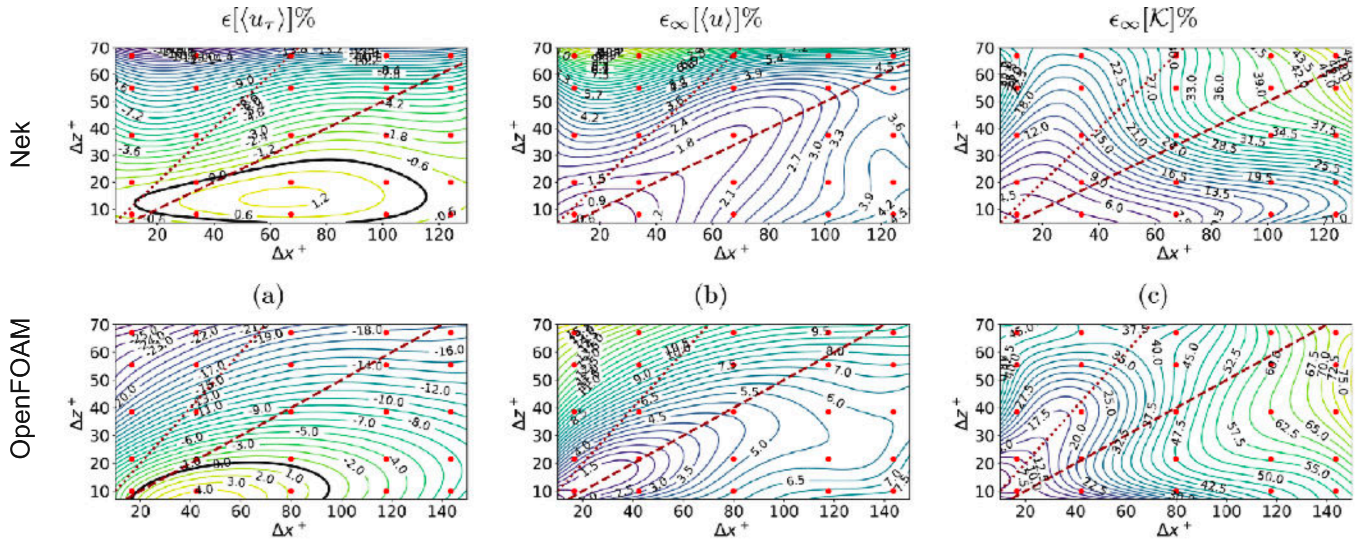
Spectral Element Method

- Variational method, similar to FEM, using Gauss-Lobatto-Legrende points
- Domain partitioned into E (unstructured) high-order elements
- Trial and test function represented an Nth order tensor-product polynomials within each element (typically N=5-9)
- EN^3 grid points in 3D
- Low dissipation and small dispersion errors
- Fast (matrix-free) operator evaluation: o(n) storage, o(nN) work

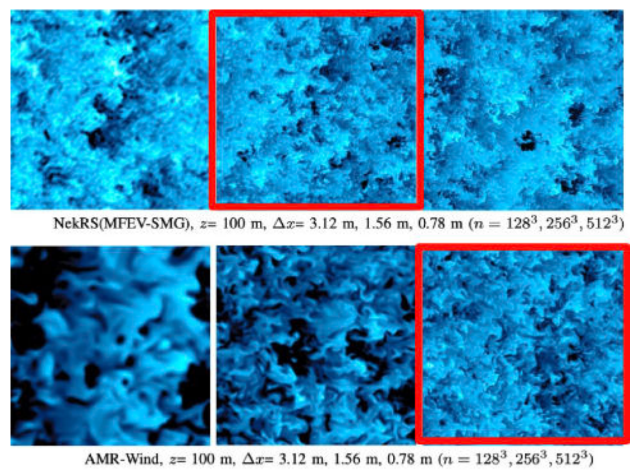


2D basis function, N=10



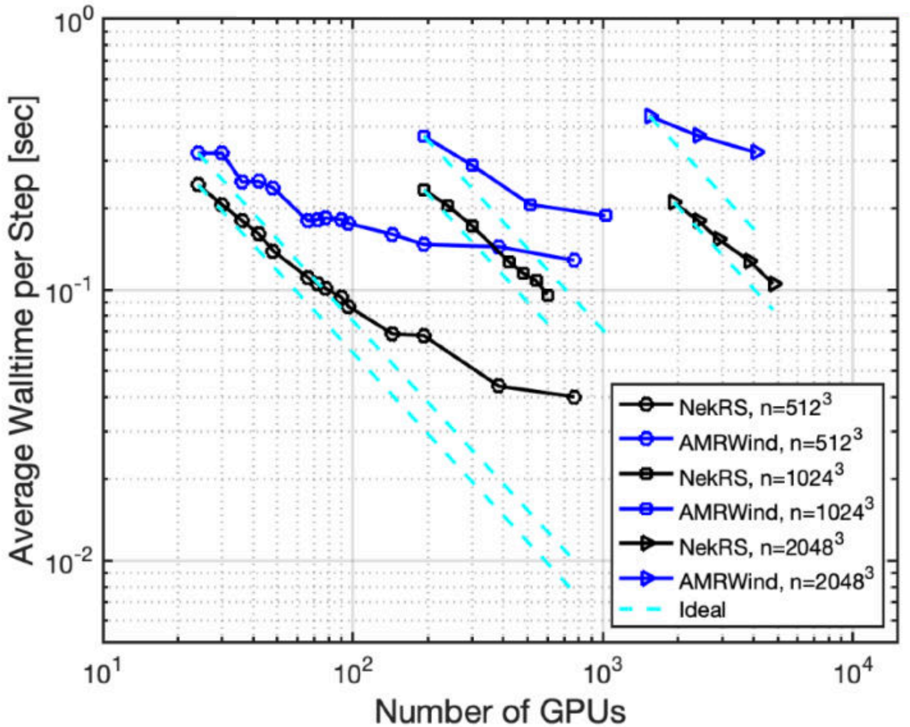


(d) (e) *Saleh Rezaeiravesh et al., C&F 2021* (f)



Misun Min et al., arXiv preprint arXiv:2210.00904

Nek requires (e.g. ~8x) less grid points for the <u>same</u> accuracy



Cost per grid point competitive to low-order methods

Towards Exascale for Wind Energy Simulations, Misun Min et al., arXiv preprint arXiv:2210.00904

High-order mesh generation	Steady state solvers	Robustness to under resolution	Shock capturing	Turbulence wall models

MPI +

CUDA

HIP

Raja

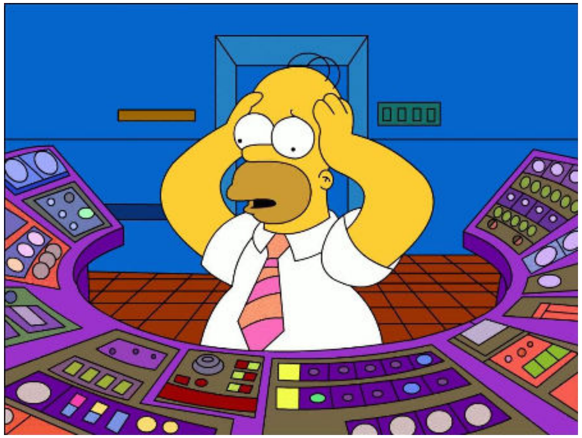
SYCL

openMP

Kokkos

openACC

=



WHAT IS OCCA?

Open
Concurrent
Compute
Architecture

https://github.com/libocca/occa

API

- Unified models for device, memory, etc.
- Backend selection at runtime:
  - Serial, OpenMP
  - CUDA, HIP, OpenCL, Metal
- Lightweight wrappers around backend APIs
- C/C++, Fortran

Kernel Language

- Directive based extension to C/C++
- Transparent translation to backend code
- JIT compilation + caching
- Alternatively, write kernels directly with backend specific code

Command Line Tool

- Hardware info
- Available backends
- Environment variables
- Translate/compile kernels

Kris Rowe, ANL

Periodic Hill at strong scaling limit

E=3.14M

N=7

1.08B gridpoints

ALCF Mira

No. 3 TOP500 2012

16384 out of 49152 nodes (33%)

16384 CPUs (16 cores each 2 hw-threads)

n/P ~ 2k (4k per core)

code: nek5000

ORNL Summit

No. 1 TOP500 2018

88 out of 4600 nodes (2%)

528 GPUs

n/P ~ 2M (25k per SM)

code: nekRS

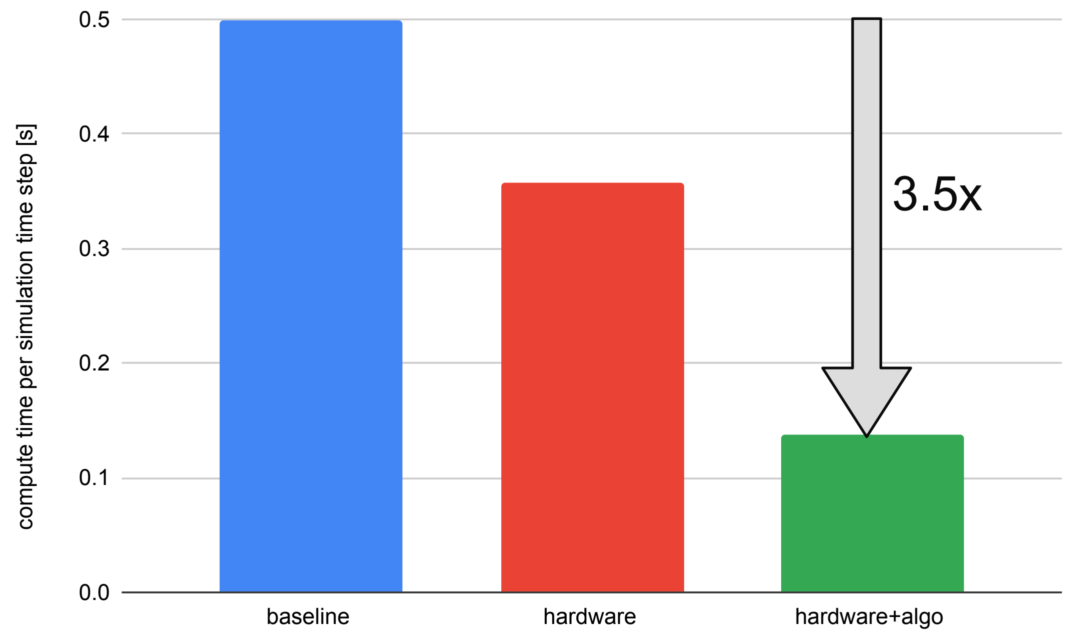
Summary

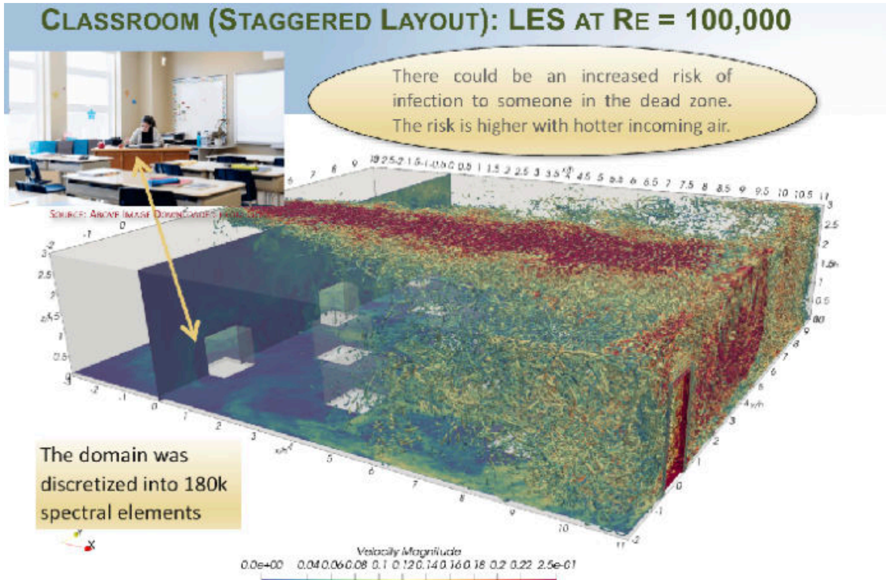
~ From 33% (hero) to 2% (toy) machine usage

~ 3.5x faster

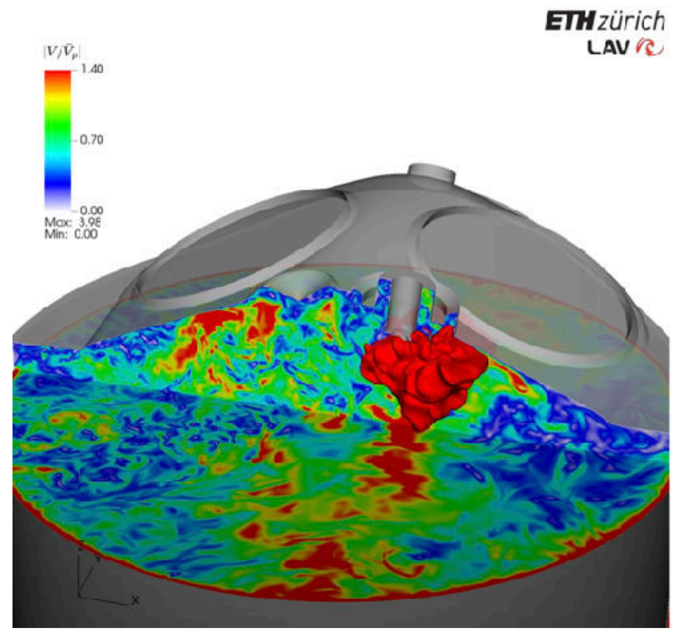
~16x lower energy consumption

Ramesh Balakrishnan, ANL





Ramesh Balakrishnan, ANL



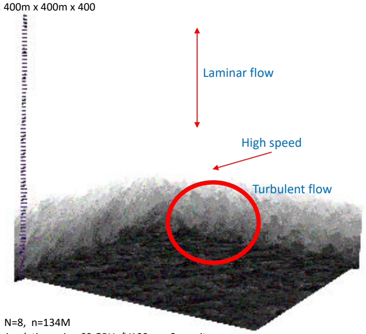
George Giannakopoulos, ETH Zurich

Wind Energy Application

Atmospheric Boundary Layer Flows

GABLS benchmark problem

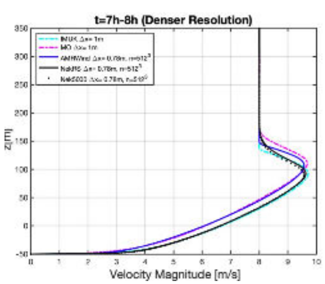
Domain: 400m x 400m x 400



E=64³, N=8, n=134M

NekRS simulation using 60 GPUs/V100s on Summit

Neil Lindquist (UTK)



Flow analysis with spatial and time-averaged velocity

(Ananias Tomboulides, Misun Min)

Nuclear Energy Application

Pebble Bed Simulations on Full Summit

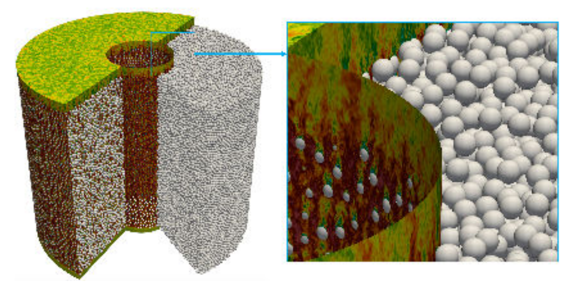


Figure 8: Turbulent flow in an annular packed bed with $\mathcal{N} = 352625$ spheres meshed with $E = 98,782,067$ spectral elements of order $N = 8$ ($n = 50$ billion gridpoints). This NekRS simulation requires 0.233 seconds per step using 27648 V100s on Summit. The average number of pressure iterations per step is 6.

- *352,625 spherical pebbles*
- *E=99 M elements*
- *N=51 B gridpoints*
- *1.4 TB per snapshot (FP32)*
- *P=27648 V100s (all of Summit)*
- *High quality all-hex mesh generated by tessellation of Voronoi facets that are projected onto the sphere or domain boundaries to yield hexahedral elements*
- *~300 elements / sphere*
- *Turbulent flow in the interstitial region between the randomly-packed spheres.*

MS121 Mon 1:50pm Paul Fsicher

MS220 Wed 5:20pm YuHsiang Lan

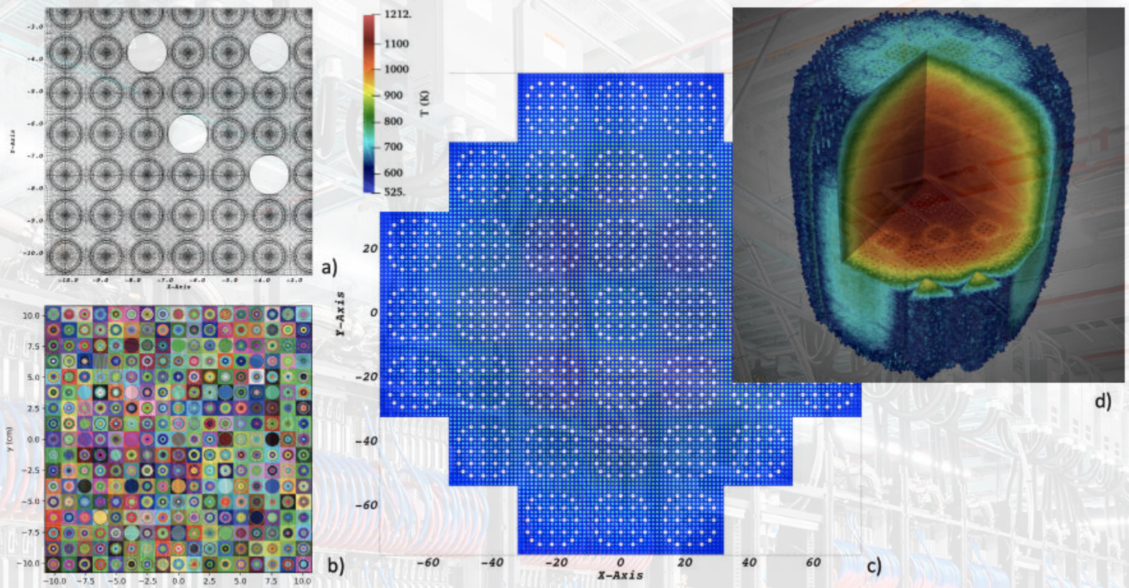
Y. Lan, P. Fischer, E. Merzari, M. Min: All hex meshing strategies for densely-packed spheres. Int. Meshing Roundtable, 2021.

GB Finalist 2023

Exascale Multiphysics Nuclear Reactor Simulations for Advanced Designs

Elia Merzari[a,*], Steven Hamilton[b], Thomas Evans[b], Paul Romano[c], Paul Fischer[d,e,f], Misun Min[d], Stefan Kerkemeier[d], Yu-Hsiang Lan[d], Jun Fang[g], Malachi Phillips[e], Thilina Rathnayake[e], Elliott Biondo[b], Katherine Royston[b], Noel Chalmers[h], Tim Warburton[i]

- Largest reactor fluid flow simulation to date
- Coupled physics (neutron transport)
- ~350B gridpoints
- 9000 nodes (90%) of ORNL's Frontier



Assembly and full core simulations with ENRiCO. a) Detail of CFD model in NekRS including interior of each pin. b) Assembly Monte Carlo model. c) Temperature distribution in a cross section of the core. d) Total neutron interaction rate in the core computed by Shift.

U.S. DEPARTMENT OF ENERGY | Office of Science

Argonne NATIONAL LABORATORY

Ongoing Work

- Documentation & training
- Tune performance for ANL's supercomputer Aurora
- Extend mixed precision capabilities
- Wall models for RANS and LES
- Extended physics: magnetohydrodynamics, chemical reactive flows, …
- In-situ visualization
- Coupling to ML frameworks

Relevant Trends & Implications

- Hardware vendors focus on AI/ML workloads
- High demand for complex (multiphysics) simulations

↓

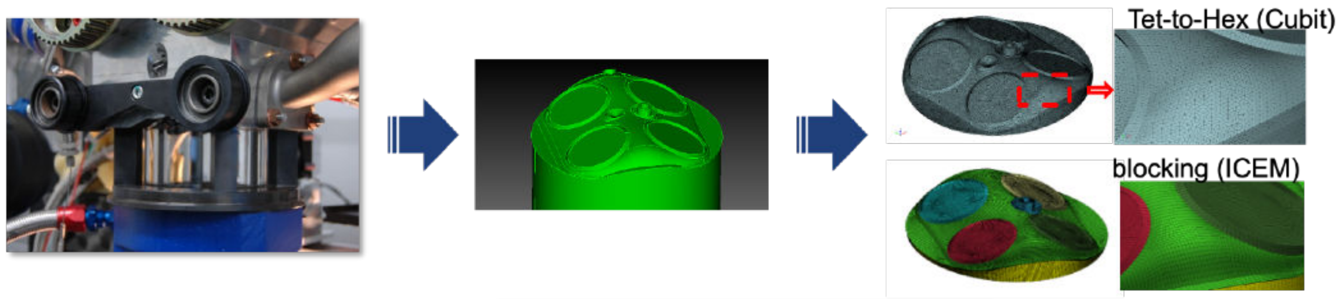
- Moderate performance improvements in particular when it comes to *time-to-solution*
- Fast changing (unstable) environment & 3rd-party integration increases cost and complexity
- Meshing and post-processing capabilities need to keep up

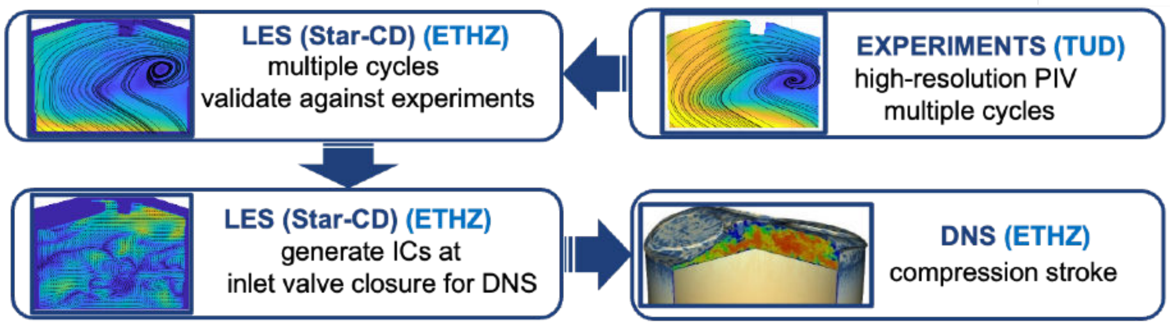
nekRS@JSC

Mathis Bode (JSC)

Application example at JSC: TU Darmstadt Engine

Overview





I. Multiple grids during compression
II. Initial conditions (interpolated U, T fields from LES)
III. Perform DNS using spectral grid-to-grid interpolation

Application example at JSC: TU Darmstadt Engine

Non-reactive nekRS full engine simulations

DNS of the compression-expansion stroke @2500 rpm (12 cycles)

4 moving meshes with up to

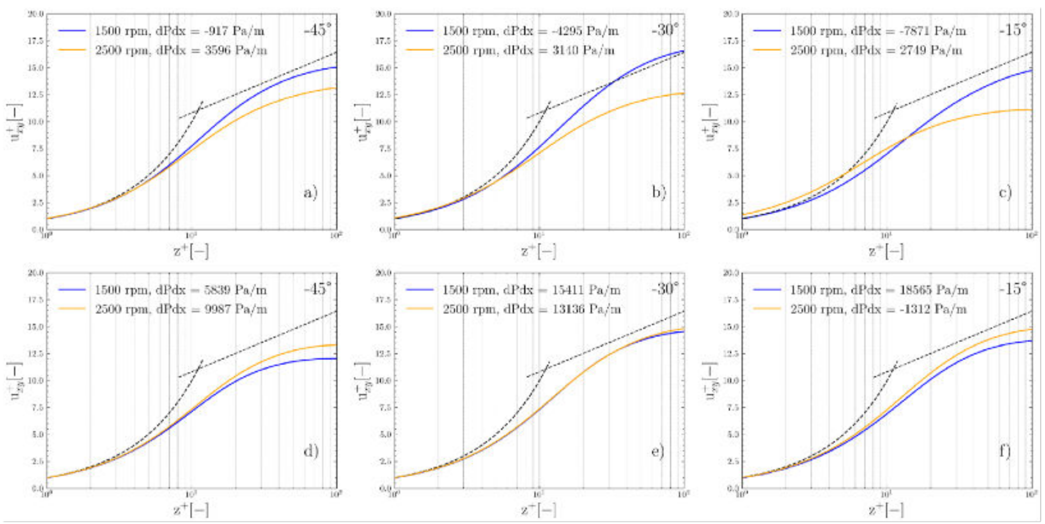
E = 9.3 million spectral elements

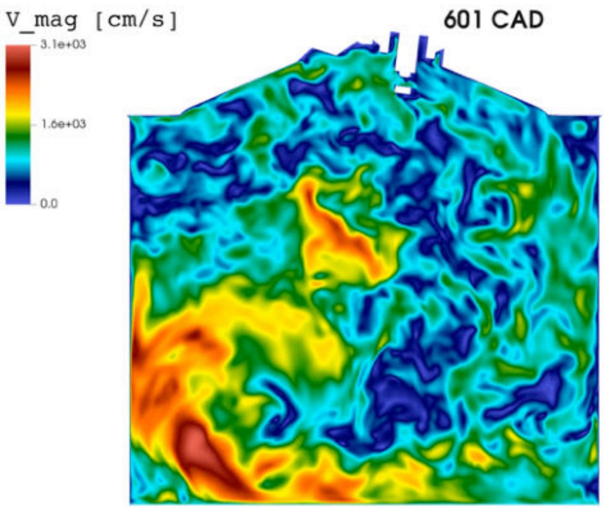
N = 9 -> 6.8 billion grid points

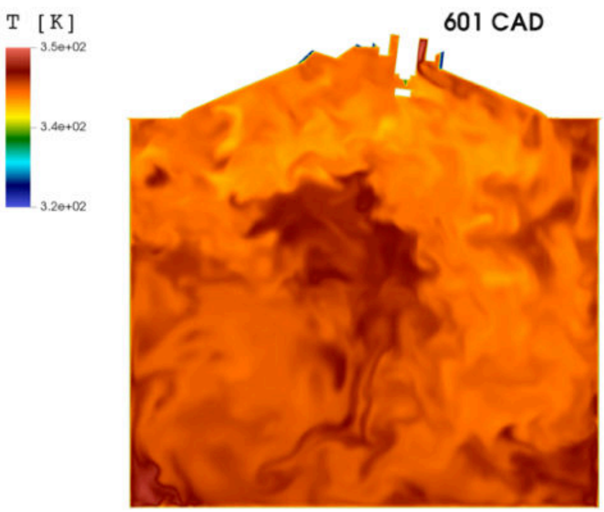
44 Tb / cycle - CAD resolved

0.27 GPU-Mcore-h / cycle (based on #cores/ CPU)

Analysis of boundary layers, turbulence, …

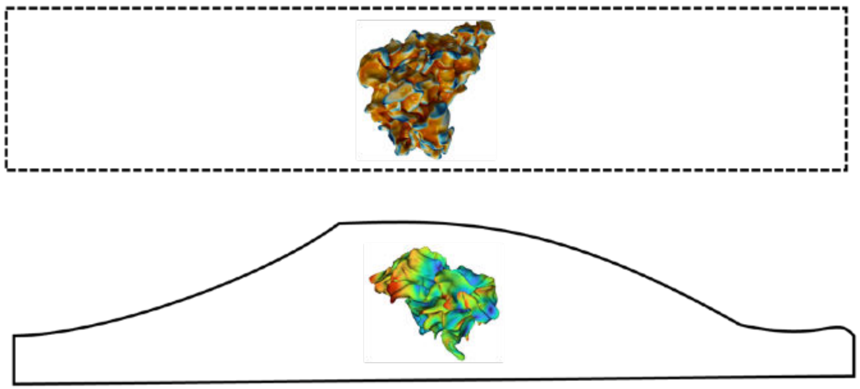




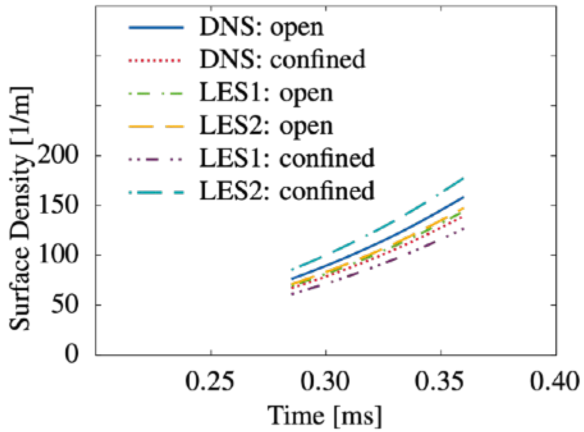


Christos Frouzakis, ETHZ

Application example at JSC: TU Darmstadt Engine



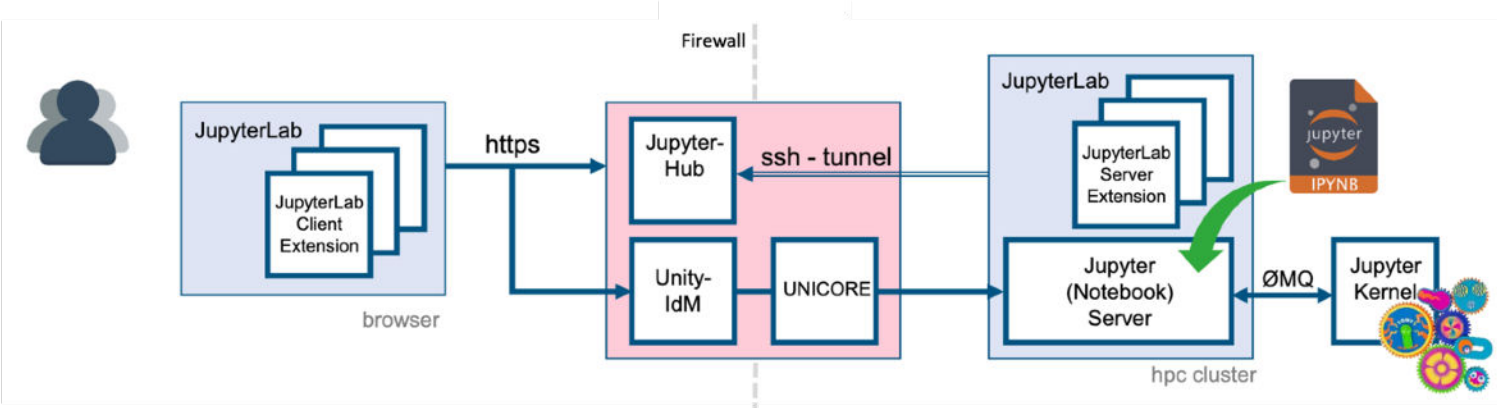
Sketch of model evaluation simulations



Evaluation of multiple LES models with OpenFOAM@JSC

Combining nekRS with other codes such as OpenFOAM to have optimal time-to-solution and wide functionality!

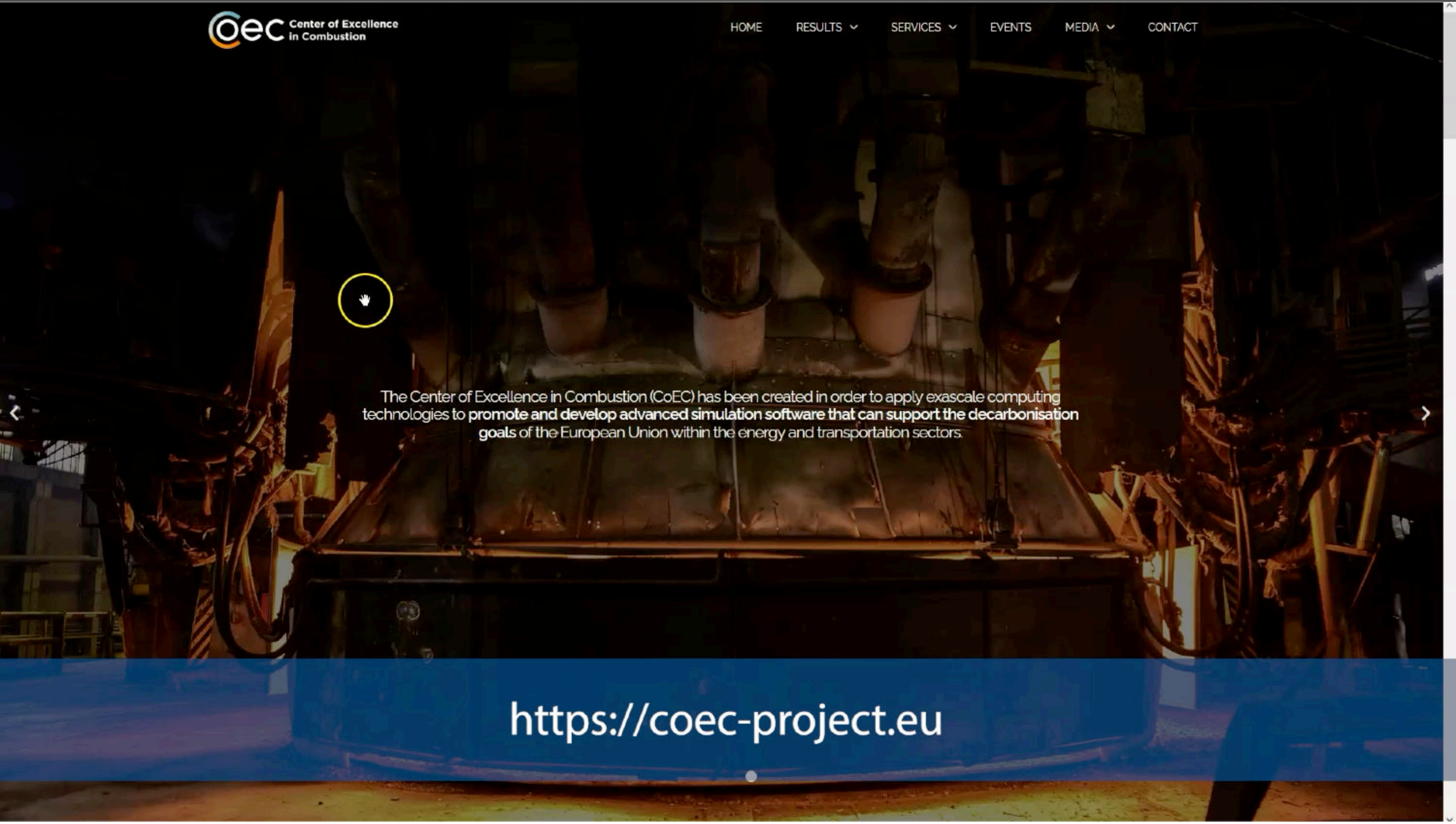
Jupyter-CoEC: Simplify the usage of nekRS@JSC



HPC resources	Functionality
JUPITER	Run Simulations
JUWELS CPU/GPU/Booster	Pre- & Post-Processing
JURECA-DC CPU/GPU	Client-Server Visualization
...	...

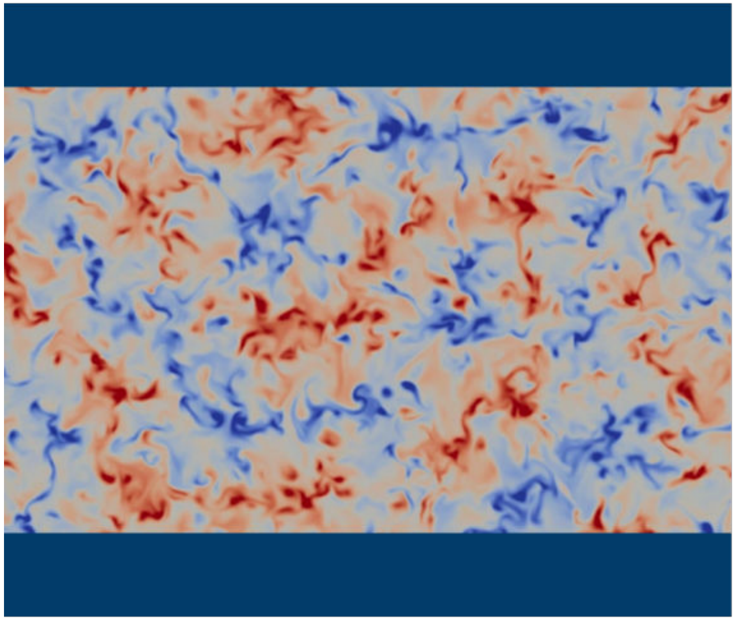
Jupyter-CoEC is simple-to-use, simple-to-access in any browser!

Jupyter-CoEC: Simplify the usage of nekRS@JSC



- ➢ https://jupyter-coec.jsc.fz-juelich.de/

Why nekRS?



nekRS is a true exascale code

Good performance at all scales

Wide applicability: From academia to industry. From boundary layers and convection processes to combustion and reactors.

Simple to use: Easy access via Jupyter-CoEC!

nekRS support @JSC:

General support and help

Powerful workflows

Assistance with computing time, good performance and scaling

Target code for exascale supercomputer JUPITER!

Contact: m.bode@fz-juelich.de

nek5000.mcs.anl.gov