Numerical Analysis group

SCAN ME

Department of Applied Mathematics (DIAM)
Faculty of Electrical Engineering,
Mathematics and Computer Science

TU Delft

Head – Kees Vuik

- Professor of Numerical Analysis
- Director of the TU Delft Institute of Computational Science and Engineering (DCSE)
- Director of Delft High Performance Computing Center (DHPC)
- Scientific Director of 4TU.AMI Applied Mathematics Institute

Staff

- Prof.dr.ir. C. Vuik
- Prof.dr.ir. C.W. Oosterlee (TUD, CWI)

- Dr. N.V. and Dr.ir. M.B. van Gijzen
- Dr. K. Cools, Dr. D. Toshniwal, Dr. M. Möller
- Dr.ir. J.E. Romate (TUD, Shell)

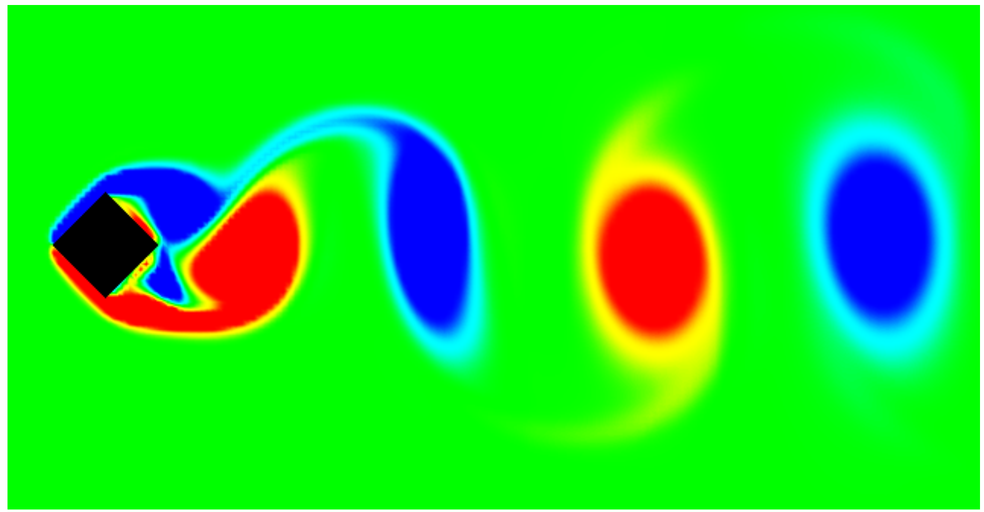
- Drs. I. Goddijn, Ir. P. van Nieuwenhuizen
- Dr.ir. D. den Ouden, Drs. S. Aerts

- 25-30 PhD students/postdocs, 20 Master/Bachelor students

Research

- **Numerical Methods**
  - BEM, FEM, IGA, meshless methods (SPH), hybrid methods (MPM)
- **Fast Solvers**
  - Krylov methods, multigrid, preconditioners, domain decomposition
- **Scientific Computing**
  - High-performance computing, GPU, FPGA, quantum computing
- **Applications**
  - CFD, CSM, multi-phase flows, inverse problems, medical imaging, computational finance, radar applications, scientific machine learning

Numerical Simulations

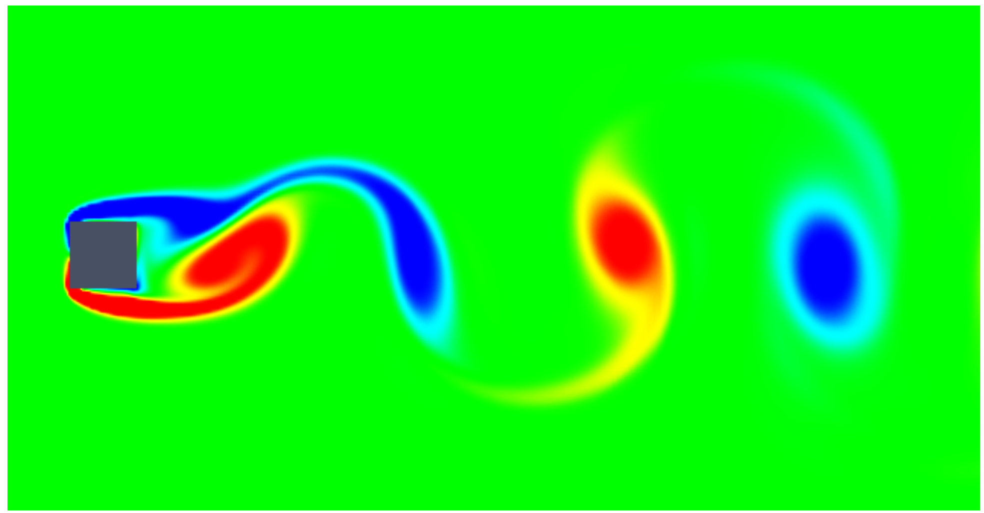


$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) - \nabla\cdot\sigma = \boldsymbol{f}$$

$$\nabla\cdot\boldsymbol{u} = 0$$

Karman vortex street animation by V. Fuka (https://artax.karlin.mff.cuni.cz/~fukav1am/sqcyl.html)

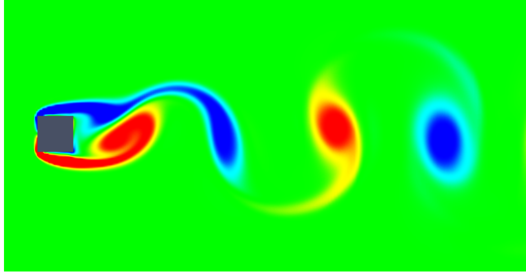
Numerical Simulations



$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) - \nabla \cdot \sigma = \boldsymbol{f}$$
$$\nabla \cdot \boldsymbol{u} = 0$$

Karman vortex street animation by V. Fuka (https://artax.karlin.mff.cuni.cz/~fukav1am/sqcyl.html)

Numerical Simulations



- Study the behavior of fluids, solids, stock market … by **virtual experiments** under changes to external parameters (geometry, material, …) in order to
  - **predict** future behaviour (weather, stock market, …)
  - **optimize** the shape of designs (aircraft wing, nozzles, …)
  - **find back** hidden parameters (inverse problems)
  - **analyse** what-if scenarios

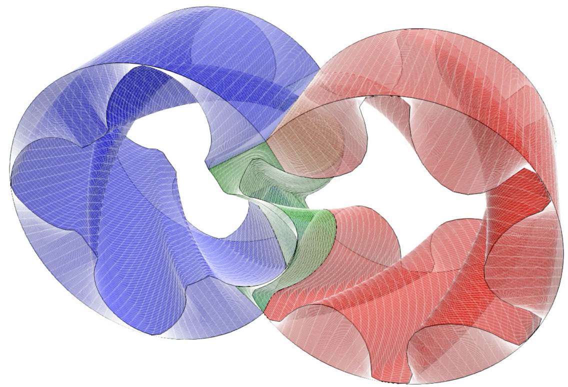
Our interests

- Development, analysis and implementation of new numerical methods
- Use of methods in applications (in collaboration with domain experts)

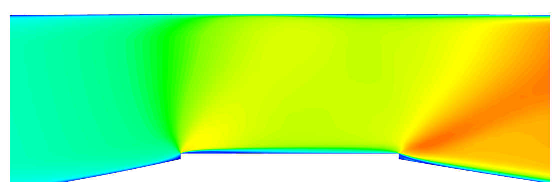
Karman vortex street animation by V. Fuka (https://artax.karlin.mff.cuni.cz/~fukav1am/sqcyl.html)

Design optimization of rotor profiles

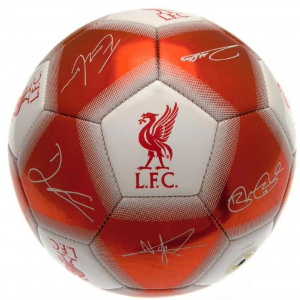
technische universität
dortmund

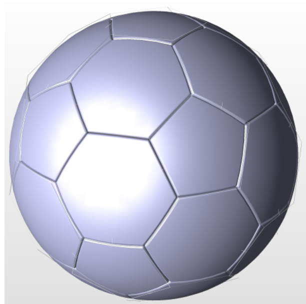


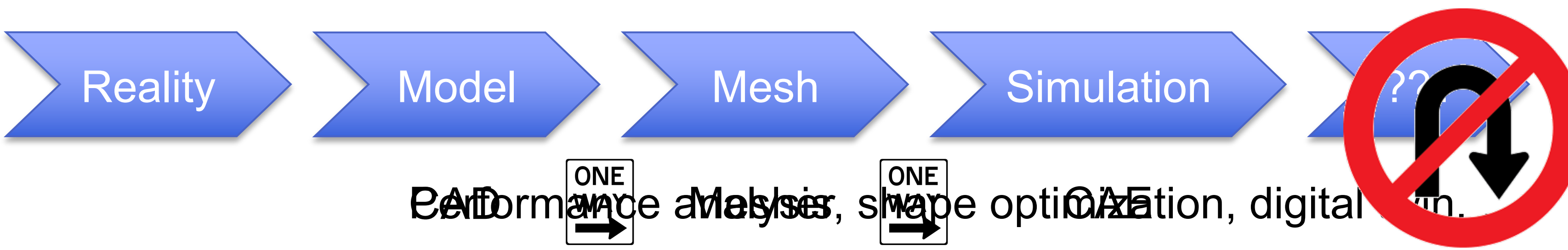
MOTOR
Multi-ObjecTive design Optimization of fluid eneRgy machines



Traditional workflow







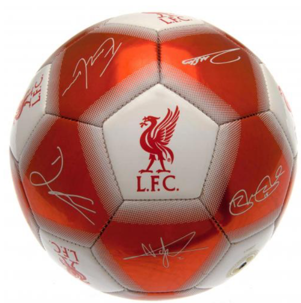
CAD

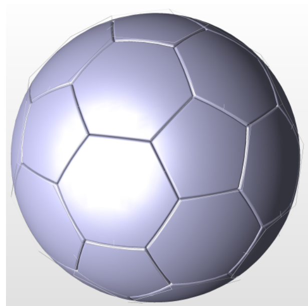
Mesher

CAE

Performance analysis, shape optimization, digital twin.

Paradigm shift: Design-Through-Analysis

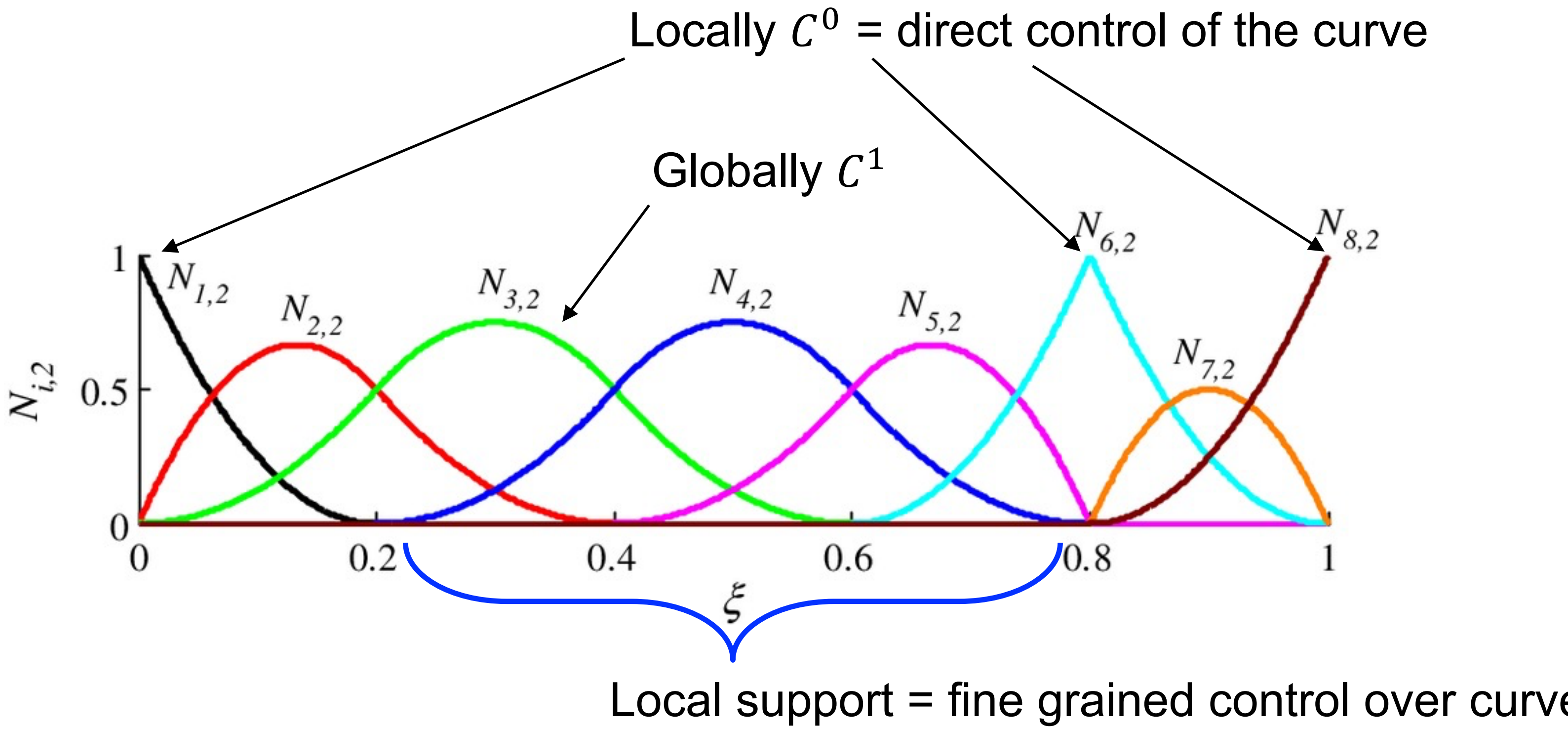




Reality

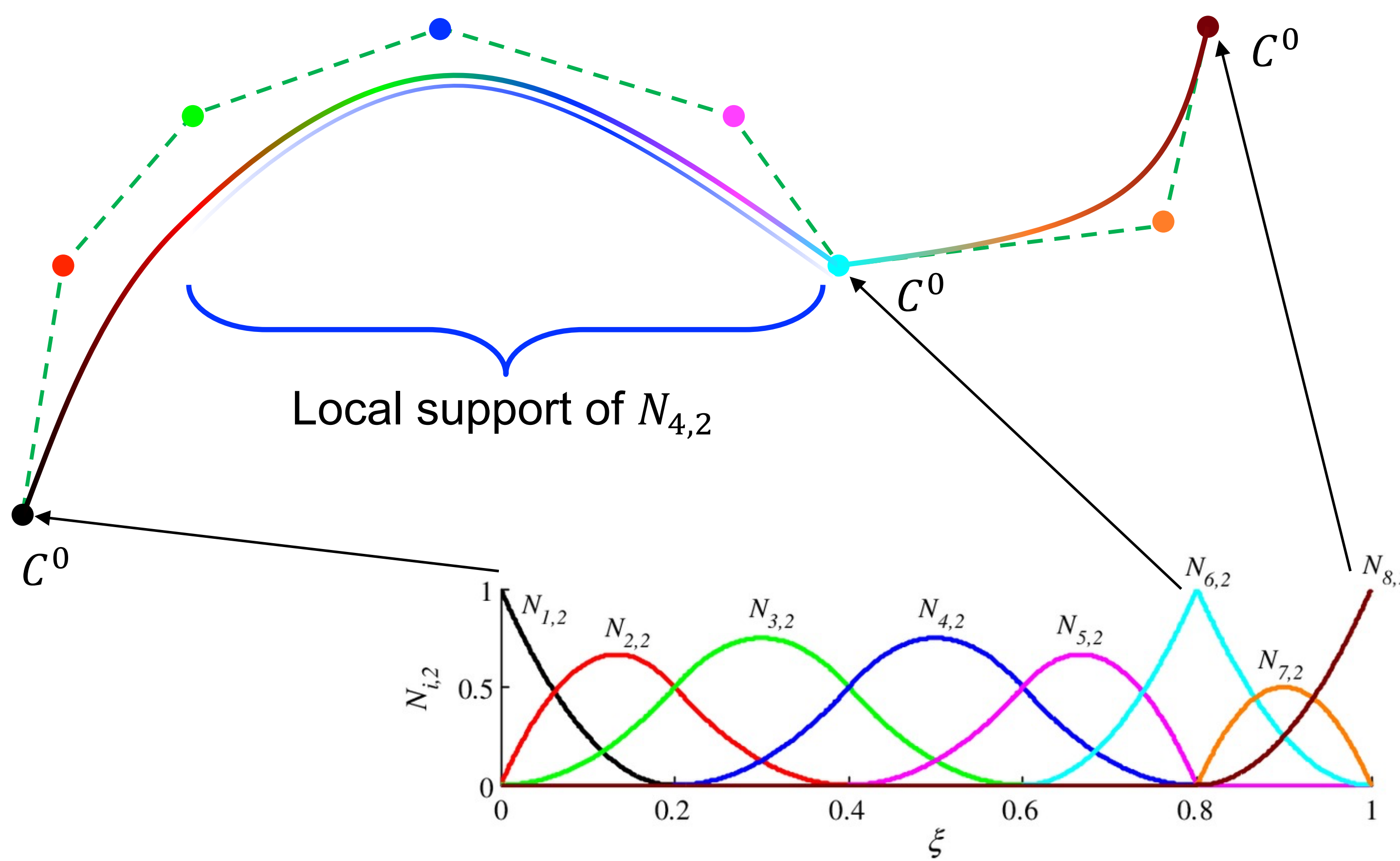
CAD-integrated CAE pipeline

B-spline basis functions (quadratic, $p = 2$)

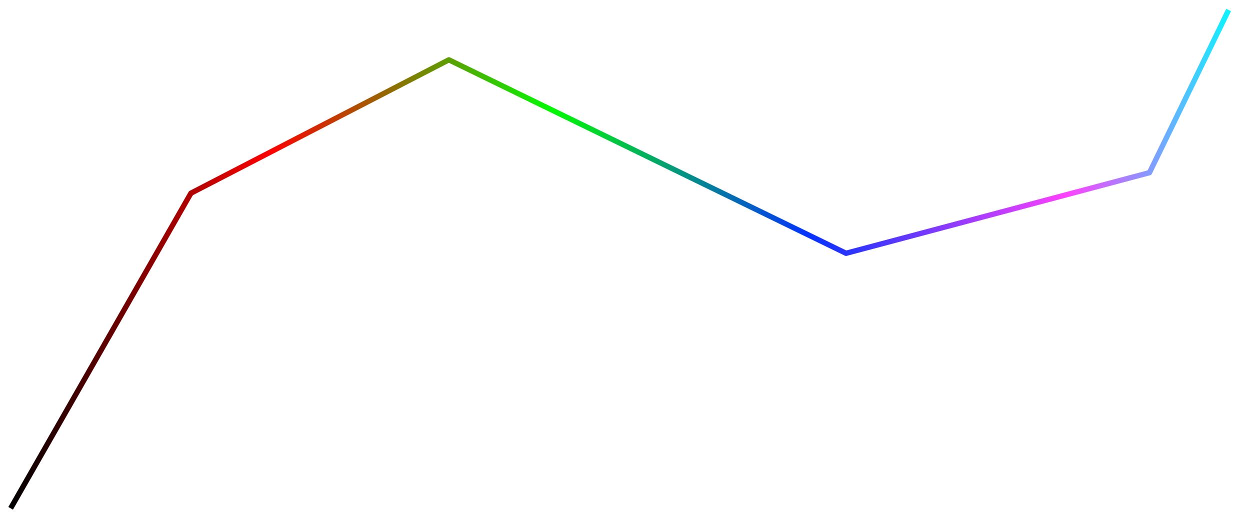


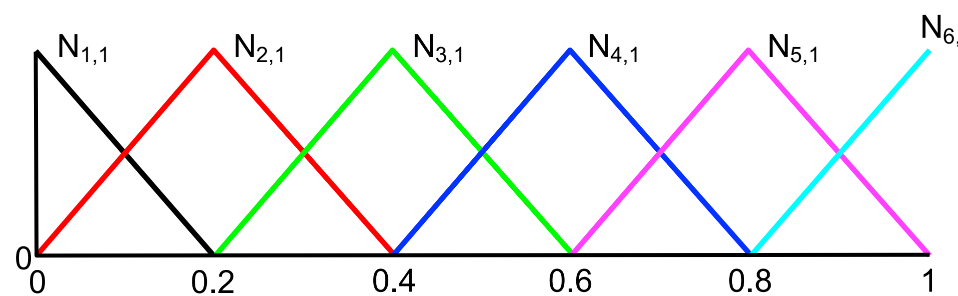
De Boor: Subroutine packages for calculating with B-Splines, TecRep LA-4728-MS, 1971

B-spline curves

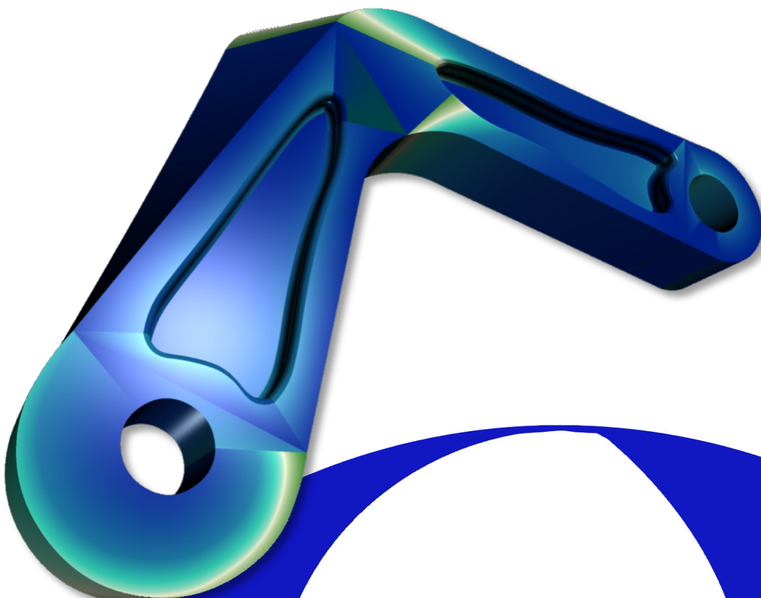


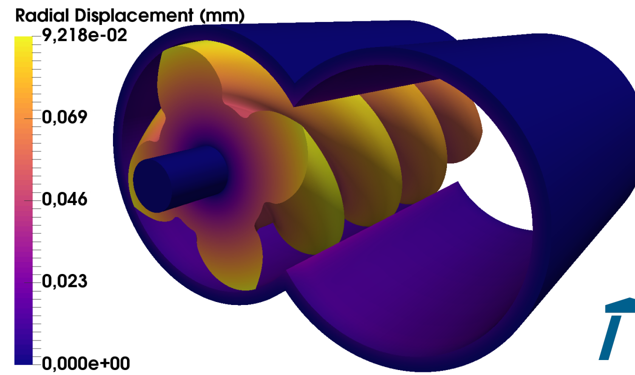
Polygons are just linear B-spline curves

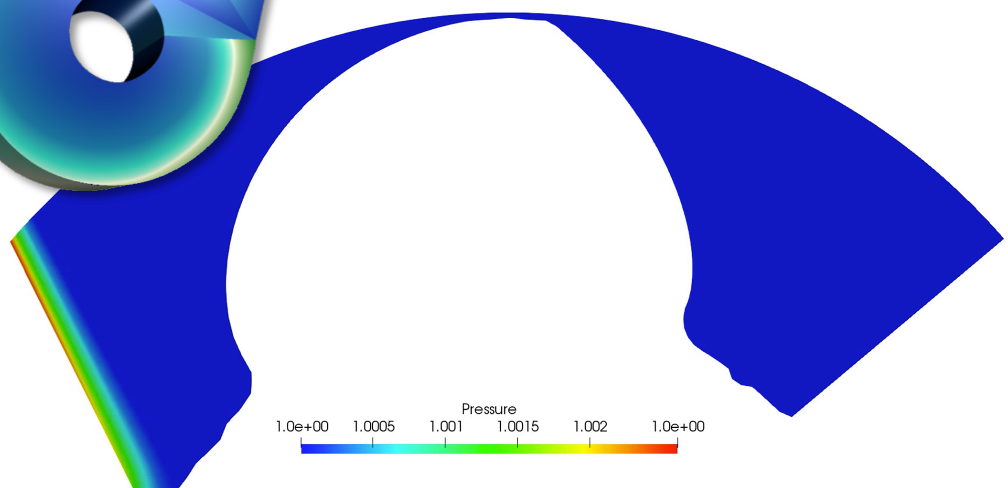


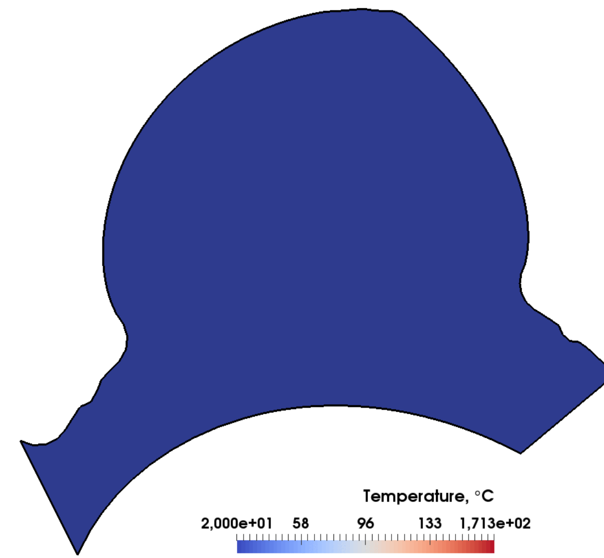


Open-source software









G+SMO

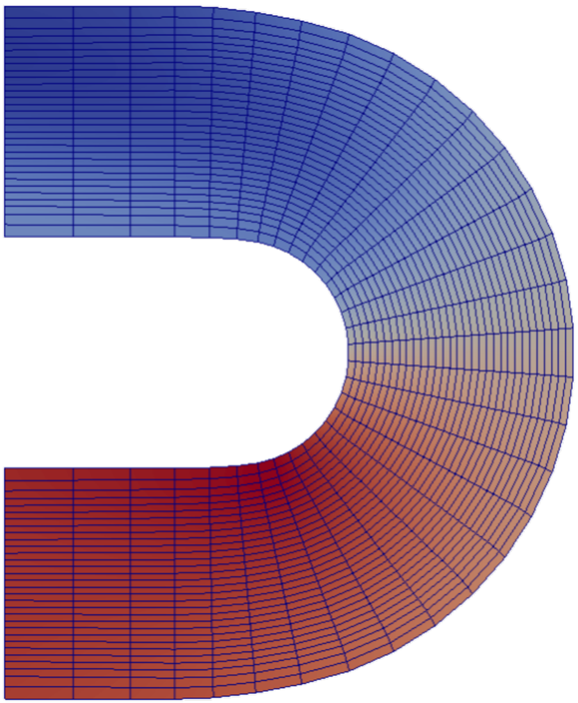
Inria

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

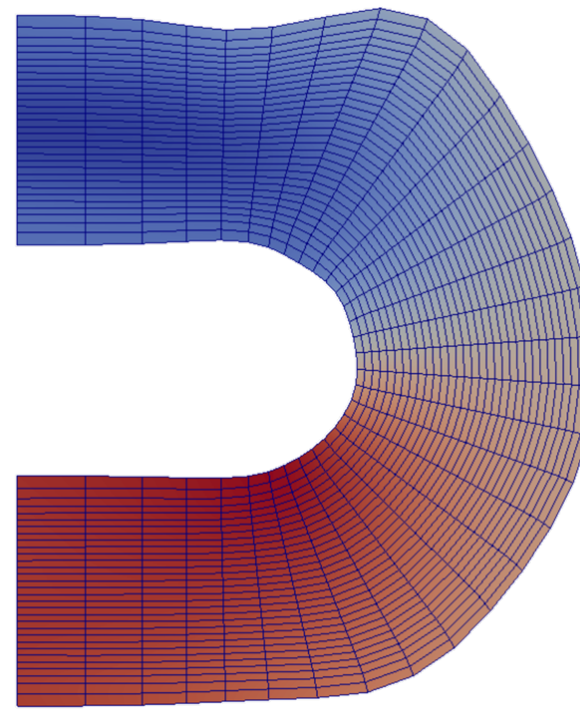
TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Adjoint-based optimization

Lodz University of Technology

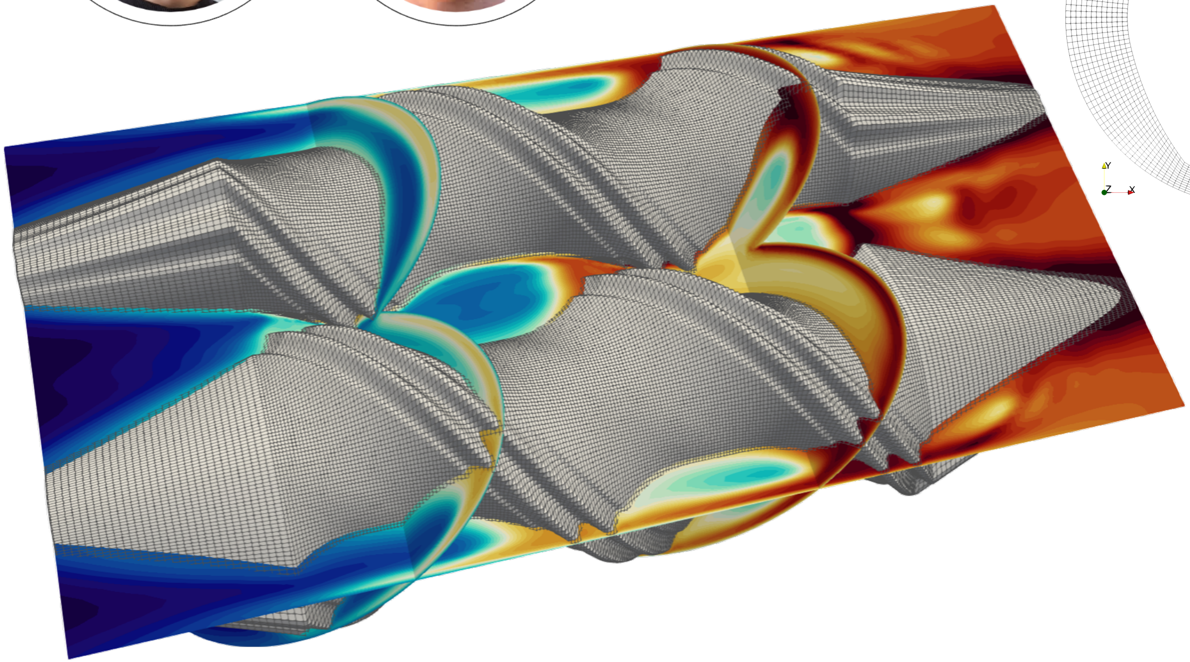


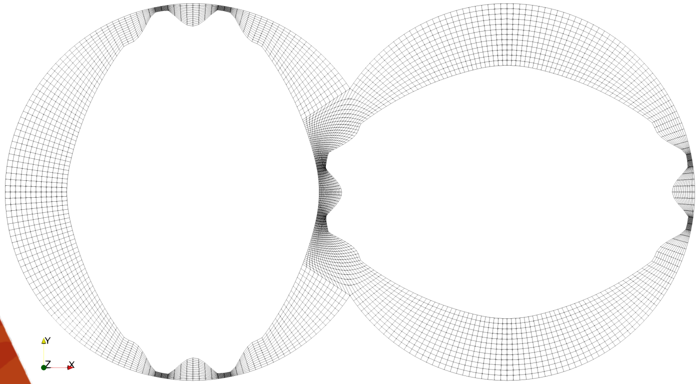
$$“\frac{\partial CFD}{\partial Design}”$$

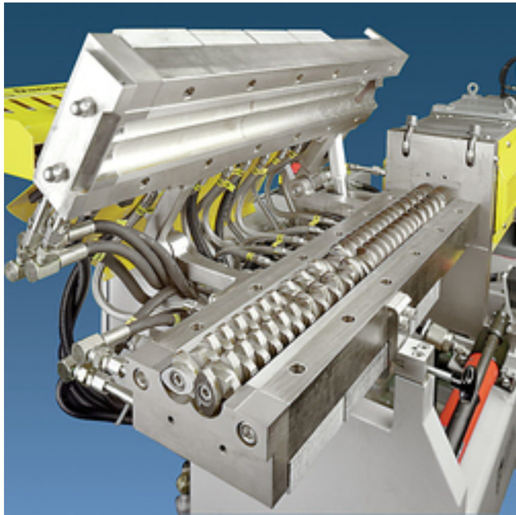


Twin-screw extruders

CATS Chair for Computational Analysis of Technical Systems | RWTH AACHEN UNIVERSITY

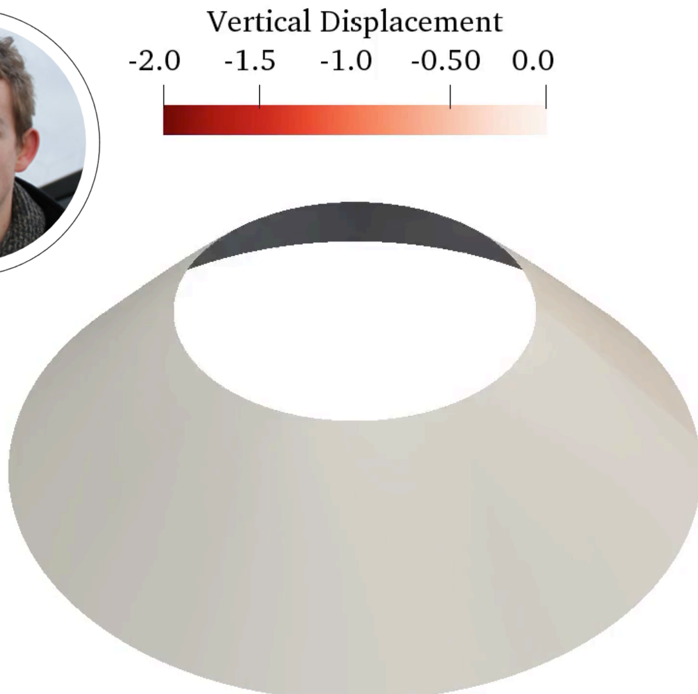


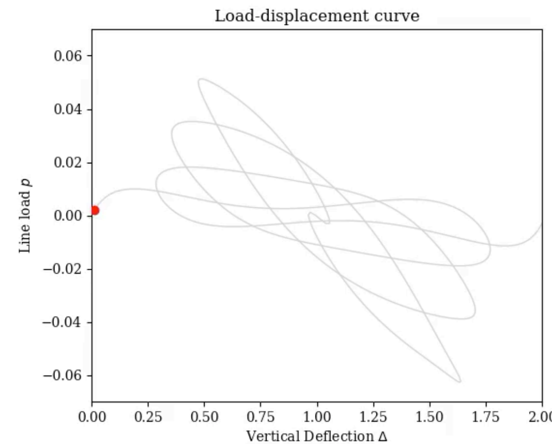


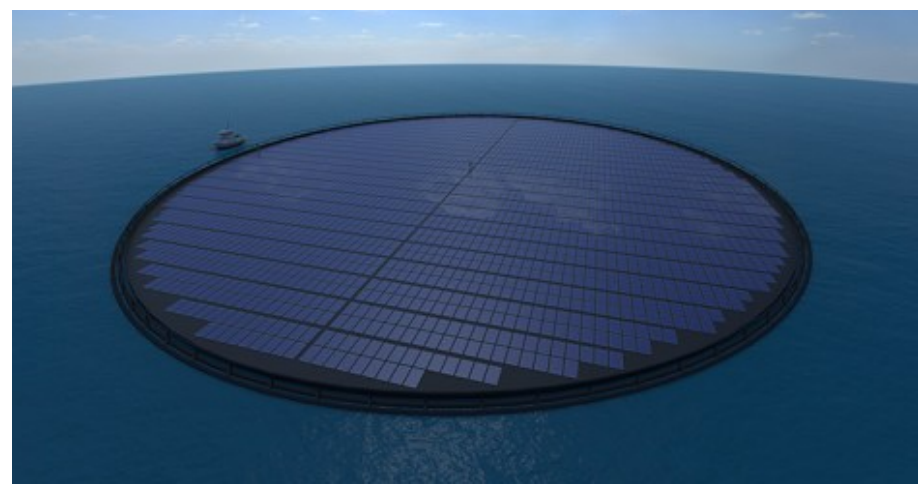


Flexible floating structures

TU Delft 3mE
Ships and Offshore Structures

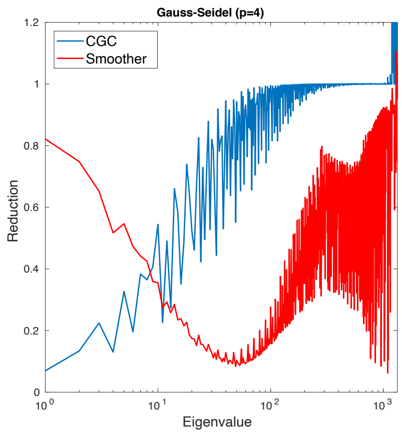


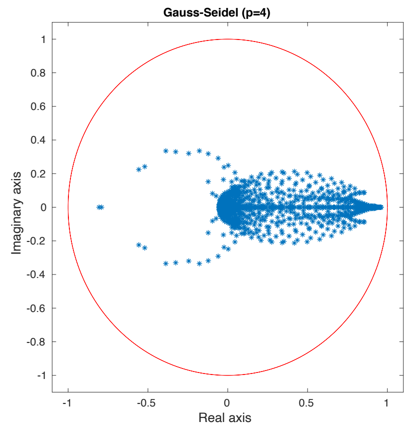


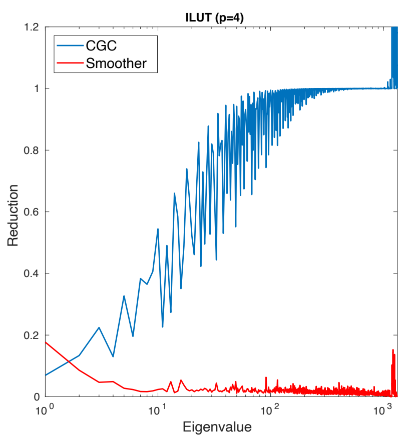


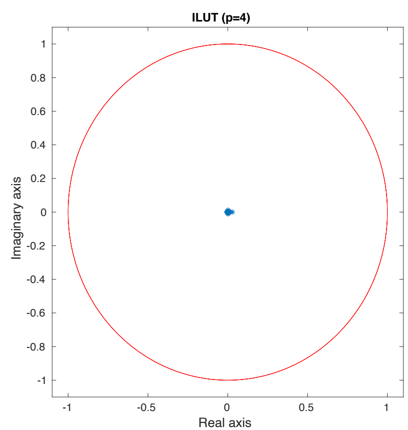
TU Delft Inria

Efficient multigrid solvers (not A\b)









	$p=2$		$p=3$		$p=4$		$p=5$	
	ILUT	GS	ILUT	GS	ILUT	GS	ILUT	GS
$h=2^{-6}$	2	13	2	18	2	41	2	78
$h=2^{-7}$	2	12	2	20	2	41	2	92
$h=2^{-8}$	3	13	2	19	2	43	2	95
$h=2^{-9}$	3	13	2	21	2	41	2	95

(a) Poissons's equation on quarter annulus

$h=2^{-6}$	2	7	2	13	2	29	2	65
$h=2^{-7}$	2	8	2	13	2	29	2	70
$h=2^{-8}$	2	7	2	12	2	29	2	64
$h=2^{-9}$	2	7	2	14	2	28	2	72

(b) CDR-equation on unit square

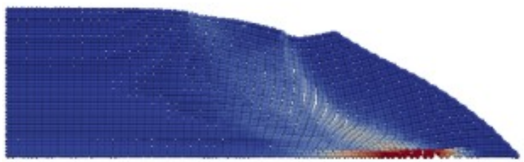
$h=2^{-6}$	3	10	2	16	2	26	2	52
$h=2^{-7}$	3	10	2	17	2	32	2	57
$h=2^{-8}$	3	10	2	17	2	33	2	66
$h=2^{-9}$	4	11	2	18	2	36	2	64

(c) Poissons's equation on L-shaped domain

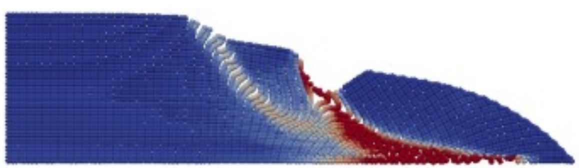
$h=2^{-2}$	2	14	2	30	2	94	3	276
$h=2^{-3}$	2	16	2	40	2	105	2	229
$h=2^{-4}$	2	19	2	44	2	119	3	285
$h=2^{-5}$	2	19	2	49	3	136	3	310

(d) Poissons's equation on the unit cube

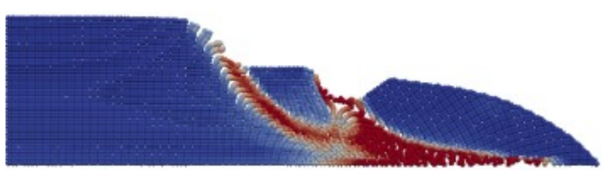
Material Point Method



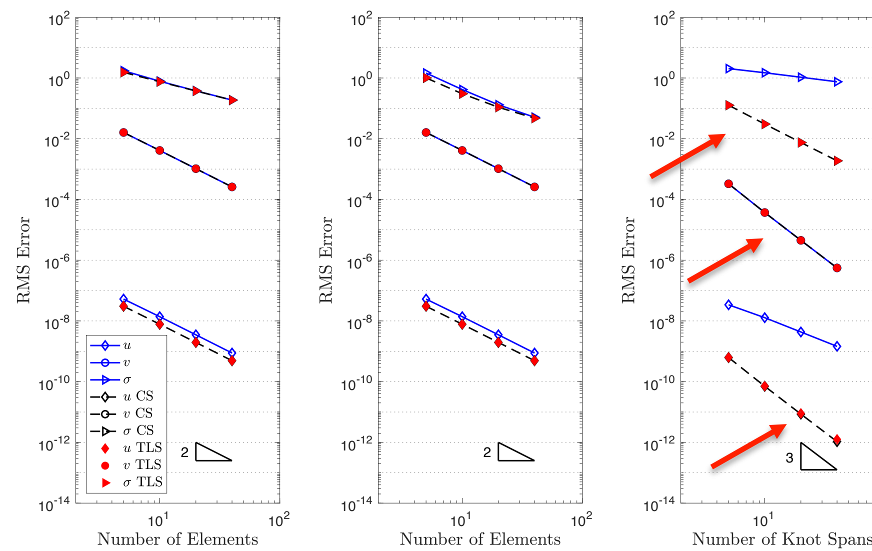
(a) formation of the first slip plane at $t = 6.0$ s



(b) formation of the second slip plane at $t = 23.25$ s



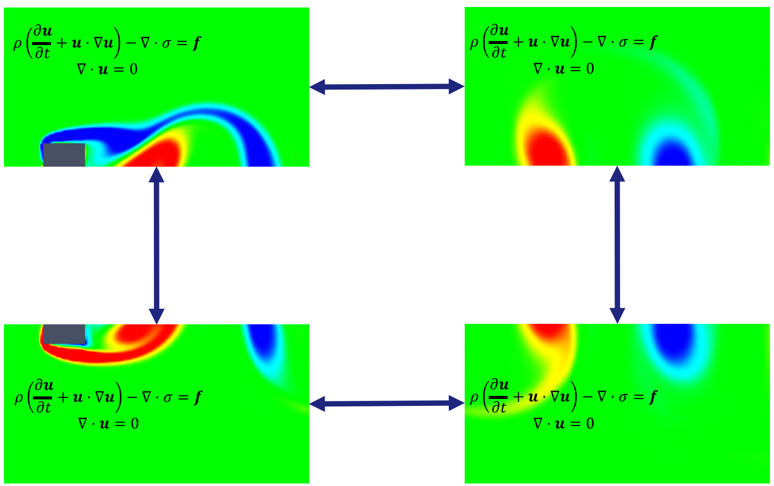
(c) final quasi-static state at $t = 38.25$ s



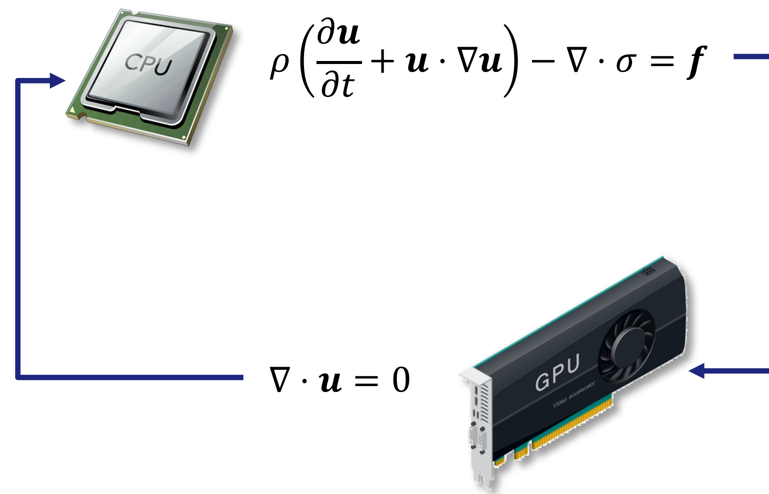
Faster convergence is better!

Scientific computing

- Divide-and-conquer

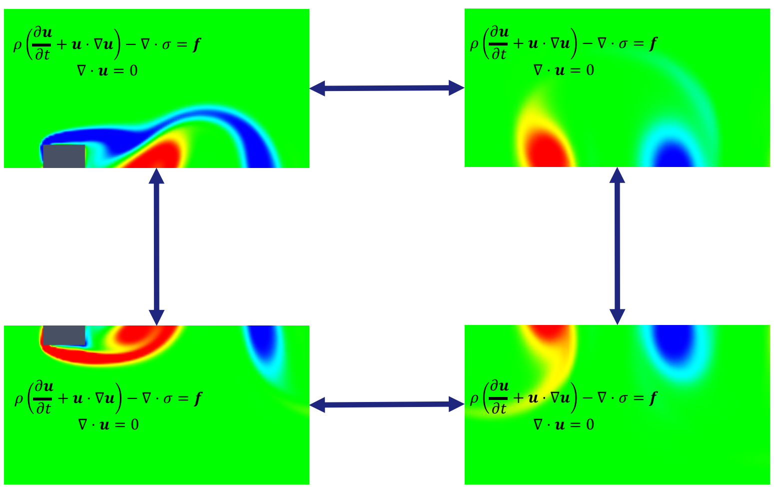


- Offloading

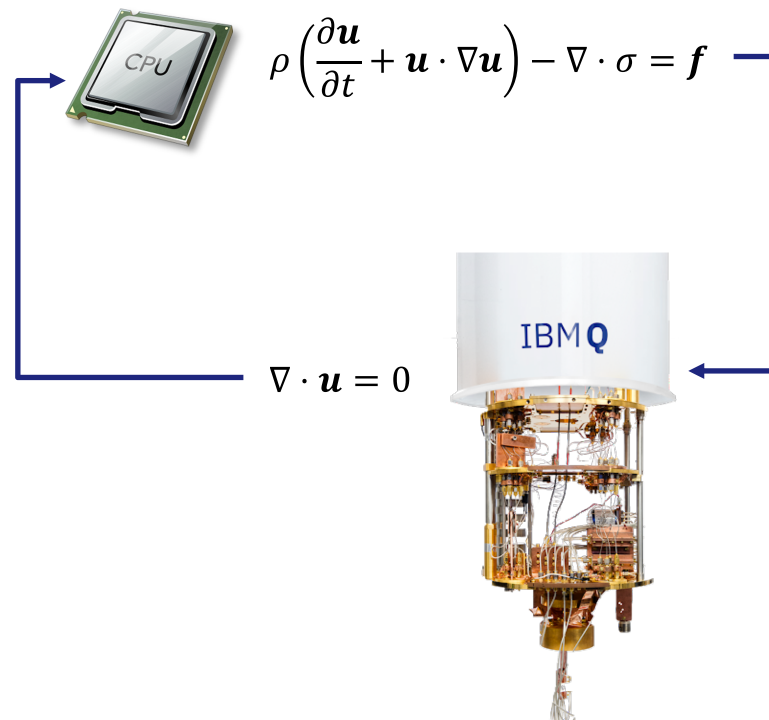


Scientific computing in the quanum era

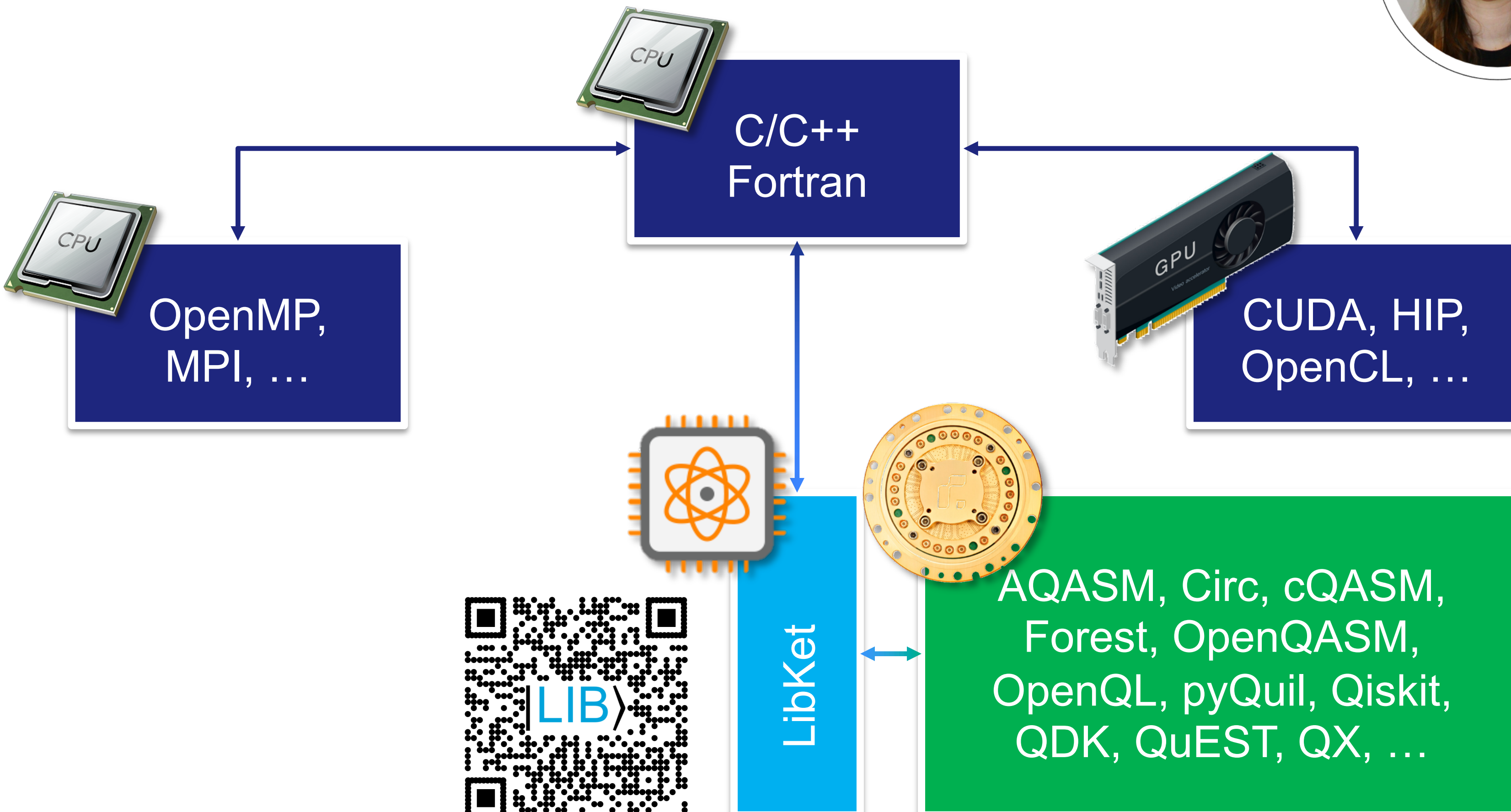
- Divide-and-conquer



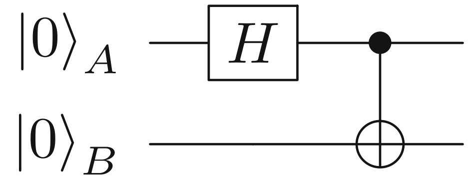
- Offloading



LibKet: a cross-platform programming framework

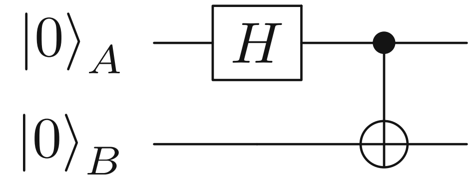


Application: First bell state

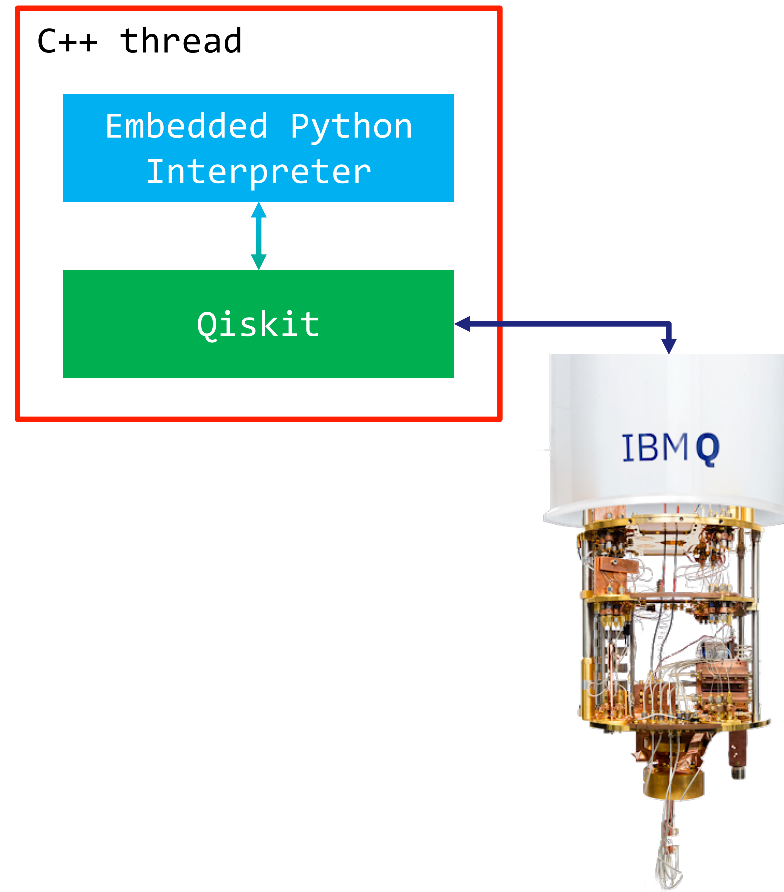



```
#include <LibKet.hpp>

// Create quantum expression
auto expr = cnot(h(sel<1>()),
                 sel<3>(init()));

// Select quantum device
QDevice<ibmq_london, 5> device;

// Populate quantum kernel
device(expr);

// Execute quantum job
auto job = device.execute_async(…, [stream]);

// Wait for job and retrieve result
auto result = job->get();
```

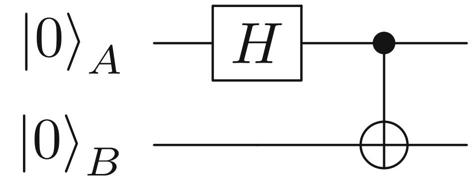
Application: First bell state



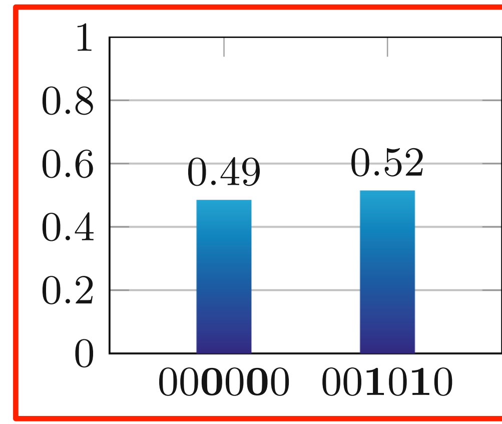

```
#include <LibKet.hpp>

// Create quantum expression
auto expr = cnot(h(sel<1>()),
                 sel<3>(init()));

// Select quantum device
QDevice<ibmq_london, 5> device;

// Populate quantum kernel
device(expr);

// Execute quantum job
auto job = device.execute_async(…, [stream]);

// Wait for job and retrieve result
auto result = job->get();
```



Application: First bell state

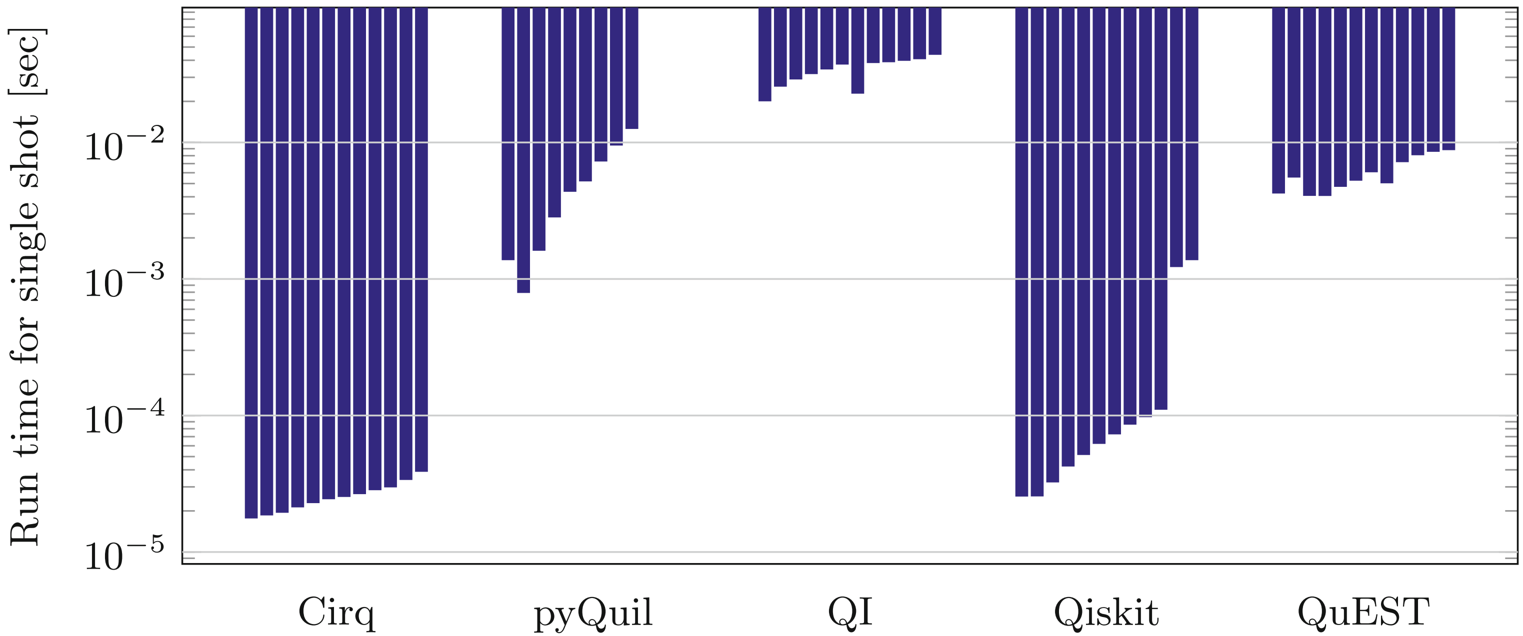



```
#include <LibKet.hpp>

// Create quantum expression
auto expr = cnot(h(sel<1>()),
                 sel<3>(init()));

// Select quantum device
QDevice<ibmq_london, 5> device;

// Populate quantum kernel
device(expr);

// Execute quantum job
auto job = device.execute_async(…, [stream]);

// Wait for job and retrieve result
auto result = job->get();
```



Towards a “QTOP-500”: n-qubit QFT benchmark

- Execute n-qubit QFT for `n=1..12` on different quantum simulators



Ongoing projects

- Quantum Linear Solvers
  [S. Sigurdson]
- Quantum Linear Solvers with application to CFD / CSM
  [E. Cappanera & G. Balducci]
- Quantum-accelerated optimization for graph problems [J. Bus]
- Quantum Machine Learning
  [C. Swart & M. v Loenen]

Ethics Inf Technol (2017) 19:253–269
DOI 10.1007/s10676-017-9438-0
CrossMark
ORIGINAL PAPER

On the impact of quantum computing technology on future developments in high-performance scientific computing

Matthias Möller[1] · Cornelis Vuik[1]

Microprocessors and Microsystems 66 (2019) 67–71
Contents lists available at ScienceDirect
Microprocessors and Microsystems
journal homepage: www.elsevier.com/locate/micpro
ELSEVIER

A conceptual framework for quantum accelerated automated design optimization

Matthias Möller[a,*], Cornelis Vuik[a]
Delft University of Technology, Delft Institute of Applied Mathematics (DIAM), Van Mourik Broekmanweg 6, XE Delft 2628, The Netherlands

Author Proof

|Lib⟩: A Cross-Platform Programming Framework for Quantum-Accelerated Scientific Computing

Matthias Möller(✉) and Merel Schalkers

Department of Applied Mathematics (DIAM) and 4TU.CEE, Centre for Engineering Education, Delft University of Technology, Delft, The Netherlands
m.moller@tudelft.nl, merelschalkers@gmail.com
http://ta.twi.tudelft.nl/nw/users/matthias/

Scientific machine learning

General-purpose ML

- GP-ANN learns to mimic behavior by massive, time-consuming training
- no guarantee of physically correct results (e.g., negative densities)

Scientific ML

- integrate prior knowledge (e.g. physic laws) into network architecture

Ongoing projects

- Matrix decompositions via SML [S. Veldkamp]
- Physics-informed neural networks for ideal MHD equations [J. Bouma]
- SML for solving PDE problems [F. v Ruiten & S. Ul Haq]
- Optimization in optics using SML [J. Imhoff]

Summary

- **Research at NAG**
  - Numerical Methods
  - Fast Solvers
  - Scientific Computing
  - Applications

- If you are interested in collaboration please contact me/us →

SCAN ME

- Thank you!