RESULTS OF THE DLR PROJECT GROUP “MACHINE LEARNING AND QUANTUM COMPUTING – DIGITALIZATION OF AIRCRAFT DEVELOPMENT 2.0”

Deutscher Luft- und Raumfahrtkongress 2024

30. September – 02. Oktober 2024 – Hamburg

Stefan Langer

Institute of Aerodynamics and Flow Technology

Braunschweig

Department $C^2A^2S^2E$

DLR

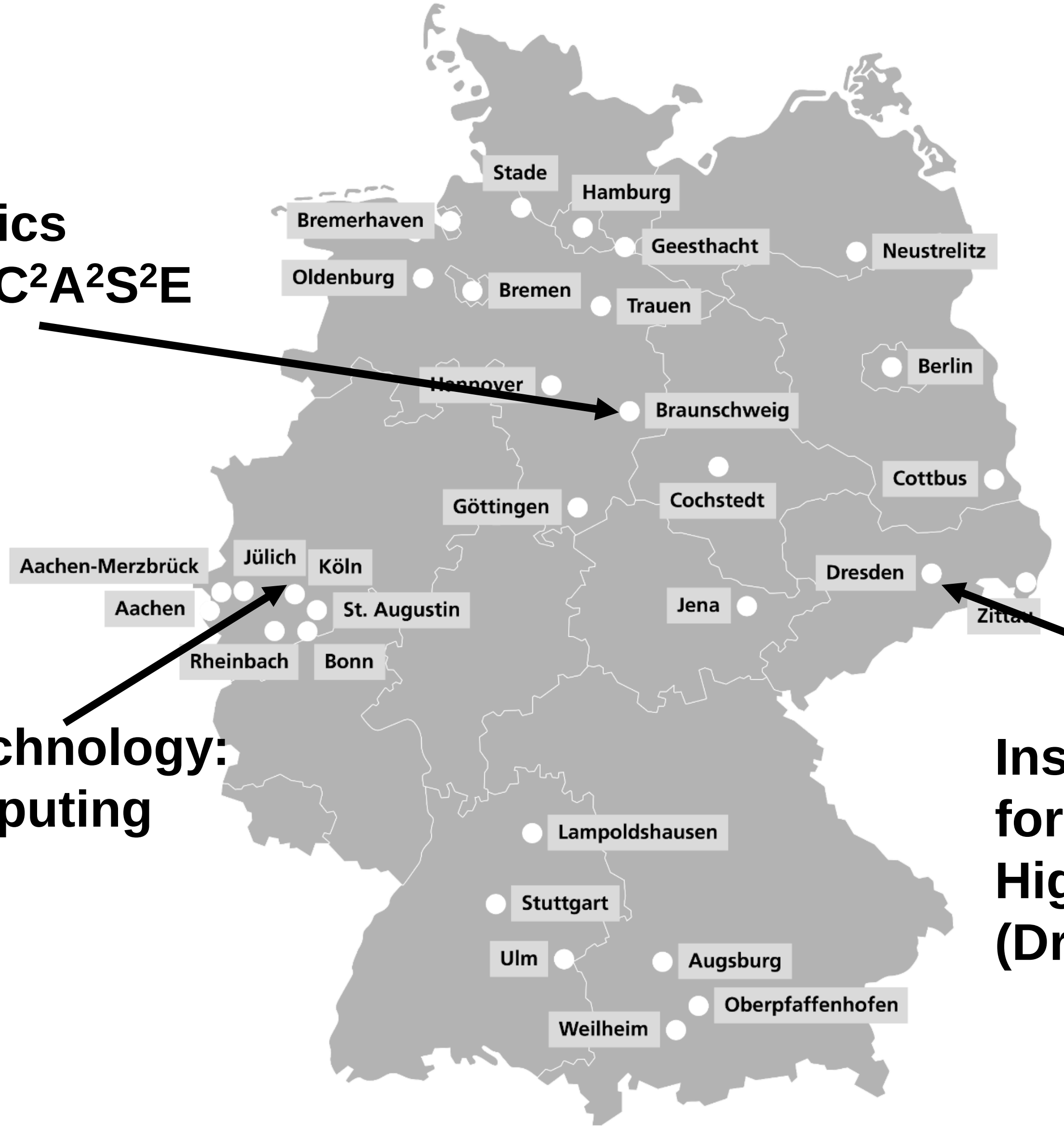
Outline

- Overview of the project
- Presentation of selected topics and results
- Summary and outlook

Participating DLR Institutes and Departments

Institute of Aerodynamics and Flow Technology: $C^2A^2S^2E$ (Braunschweig)

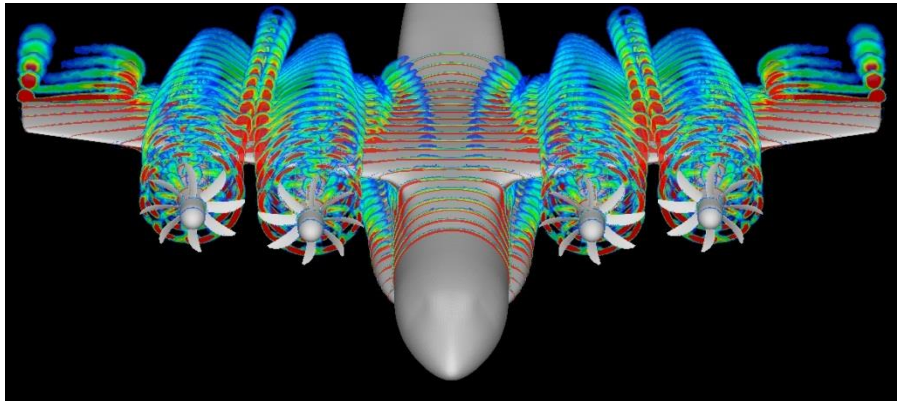
Institute of Software Technology: High-Performance Computing (Köln)

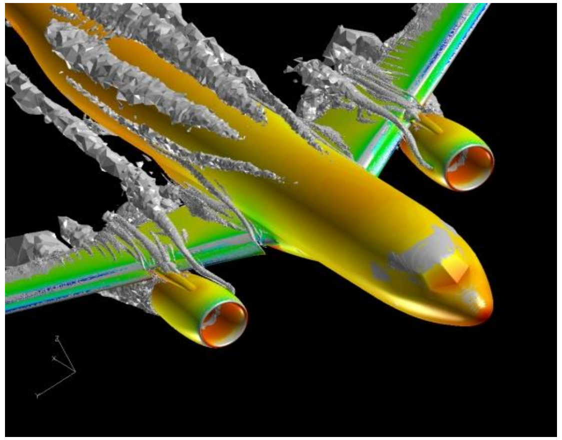


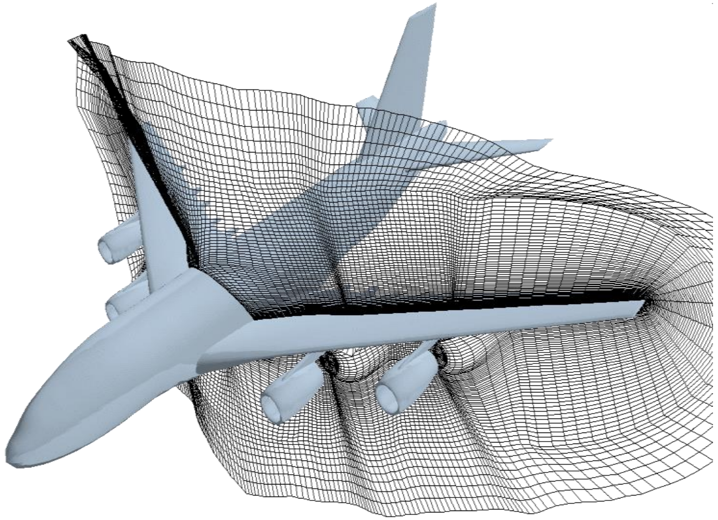
Duration of the project:
01.04.2021 – 31.03.2024
(Final meeting 06.03.2024)

Institute of Software Methods for Product Virtualization: High Performance Computing (Dresden)

Computational Fluid Dynamics







- Computational Fluid Dynamics (CFD) is an important, established tool in the aircraft industry

In order to significantly increase further capabilities and prediction accuracy, more and more complex models and larger computational meshes are required.

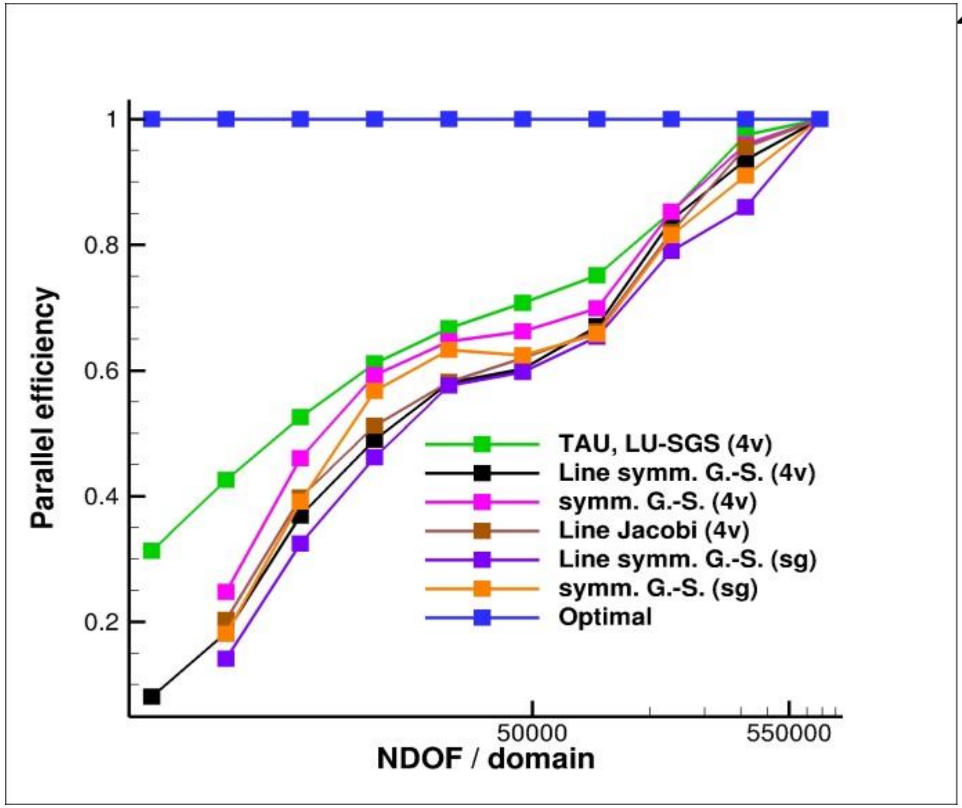
In particular scale resolving (e.g. large eddy simulation, direct numerical simulation) and unsteady computations are unfeasible for industrial test cases on classical computer hardware, even under very optimistic assumptions.

How much acceleration is still possible with classic methods?

Increasing complexity of the algorithms

- Multigrid (linear and nonlinear)
- Multistage implicit Runge-Kutta schemes
- Combination of linear solvers:
  - GmRes
  - Jacobi and Gauss-Seidel Iterations
  - LU-Solvers

- Truncation criteria for all the components
- ….



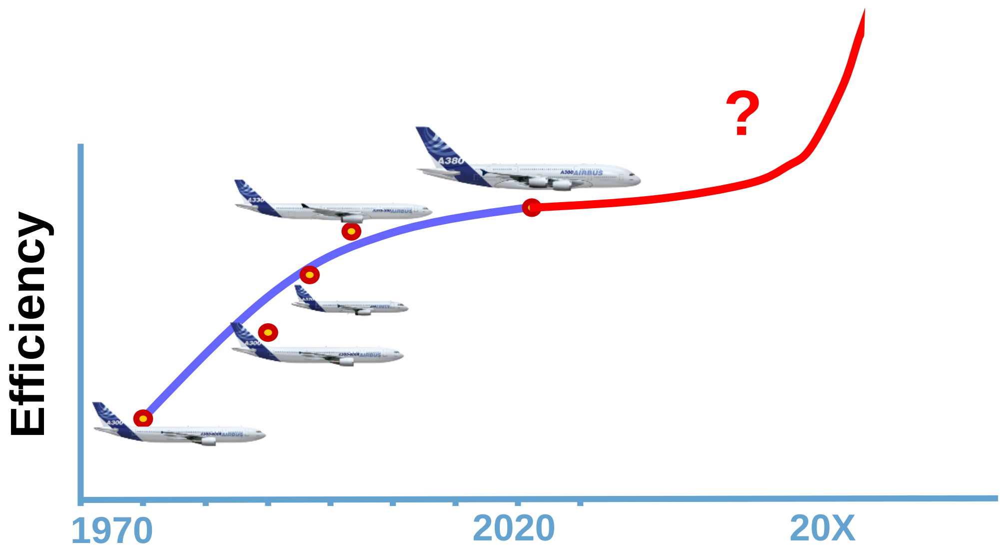
The developed algorithms show a severe loss of parallel efficiency

→It is becoming more and more difficult to achieve a significant acceleration with the classic approaches

Motivation

Industry requirements:

- Real-time simulation of full aircraft
- Optimization and certification before first flight
- Scale resolving simulations



Significantly more efficient aircraft are required to achieve climate goals

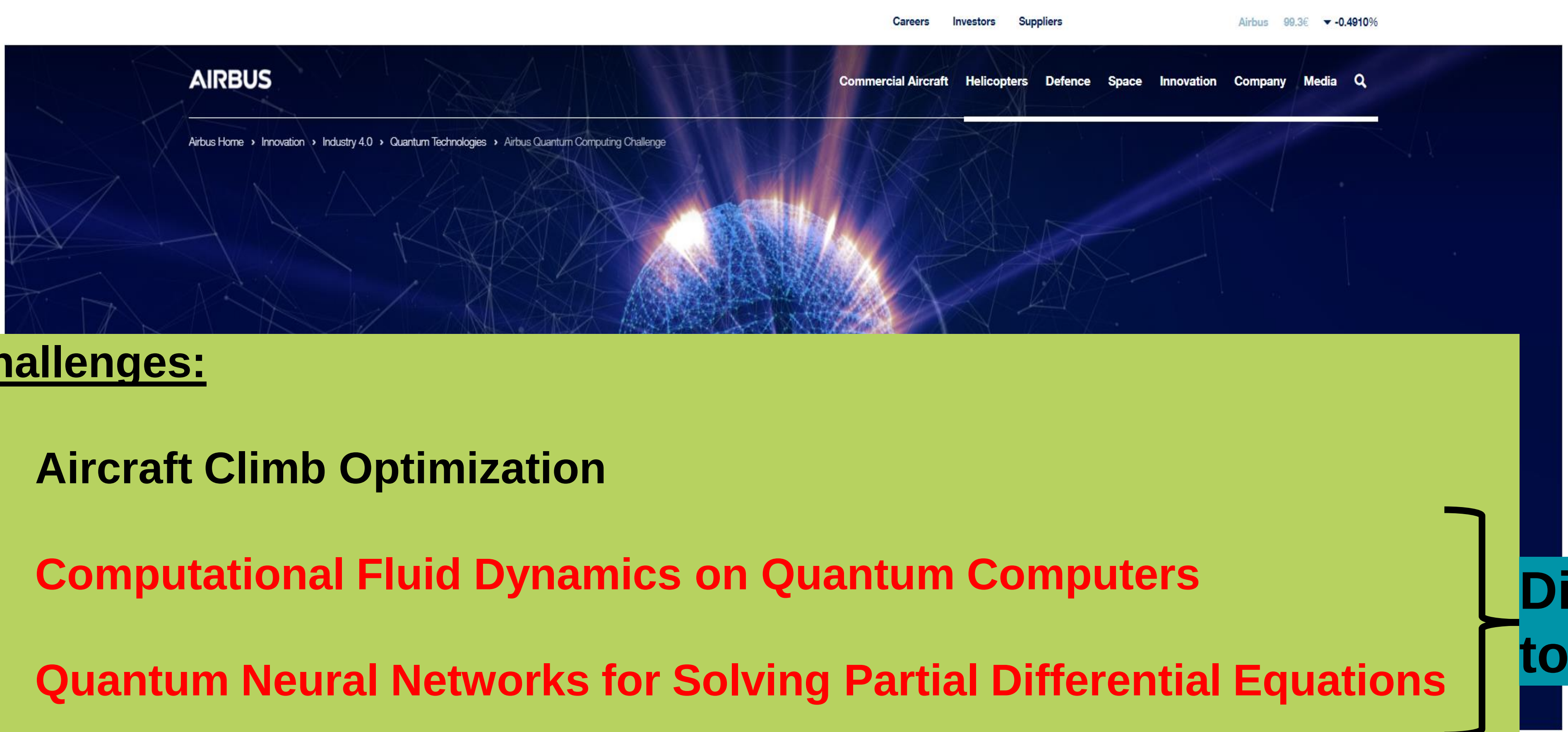
Can quantum computers be used to design future aircraft?
Is quantum computing a game changer for the aerospace industry?

Careers Investors Suppliers Airbus 99.3€ -0.4910%

Outcomes of the challenge | About the competition | Challenges | Podcasts

Toulouse, 10 December 2020

The Airbus Quantum Computing challenge has ended and we are proud to announce the 2020 winner : the team Machine Learning Reply.



<u>Challenges:</u>

1) **Aircraft Climb Optimization**
2) **Computational Fluid Dynamics on Quantum Computers**
3) **Quantum Neural Networks for Solving Partial Differential Equations**
4) **Wingbox Design Optimisation**
5) **Aircraft Loading Optimisation**

Direct reference to the project (items 2 and 3)

→ A new Quantum Computing challenge is raised in 2024 by BMW and Airbus

Goals and Perspective

Dealing with scientific and technical issues in order to **determine the potential of quantum computing and machine learning in aviation**

Possibility to permanently establish participating institutions as a **partner, supporter and pioneer** of the German and European aviation industry for questions regarding the use of quantum computing in the context of digitalization

Establish close relationships between aviation industry and DLR in questions regarding machine learning and quantum computing

Short overview of some of the topics and a selection of results

Topics

- Correcting the discretization error of coarse grid CFD simulations with Machine Learning

- Physics-Informed neural networks for solving compressible flow equations

- Quantum machine learning for partial differential equations

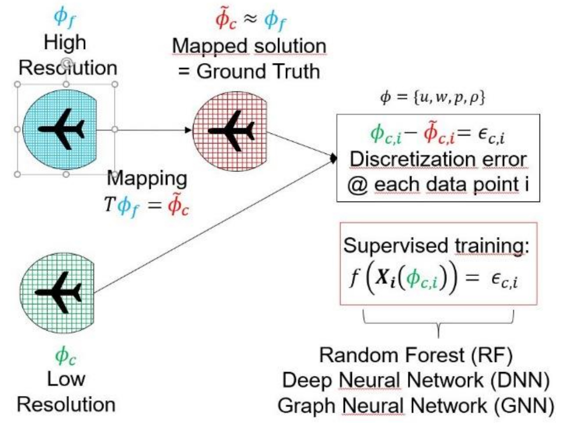
CORRECTING THE DISCRETIZATION ERROR OF COARSE GRID CFD SIMULATIONS WITH MACHINE LEARNING

Stefan Langer, Institute of Aerodynamics and Flow Technology, 01.10.2024

Motivation

- Take **fast but comparatively inaccurate** CFD simulations, such as:
  - Coarse grid finite-volume method (FVM) solution
  - Low-order discontinuous Galerkin (DG) solution

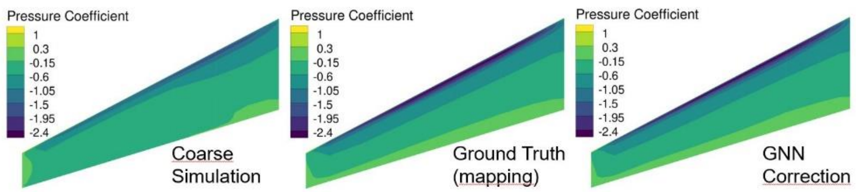
→ And then, try to improve accuracy with data-driven methods

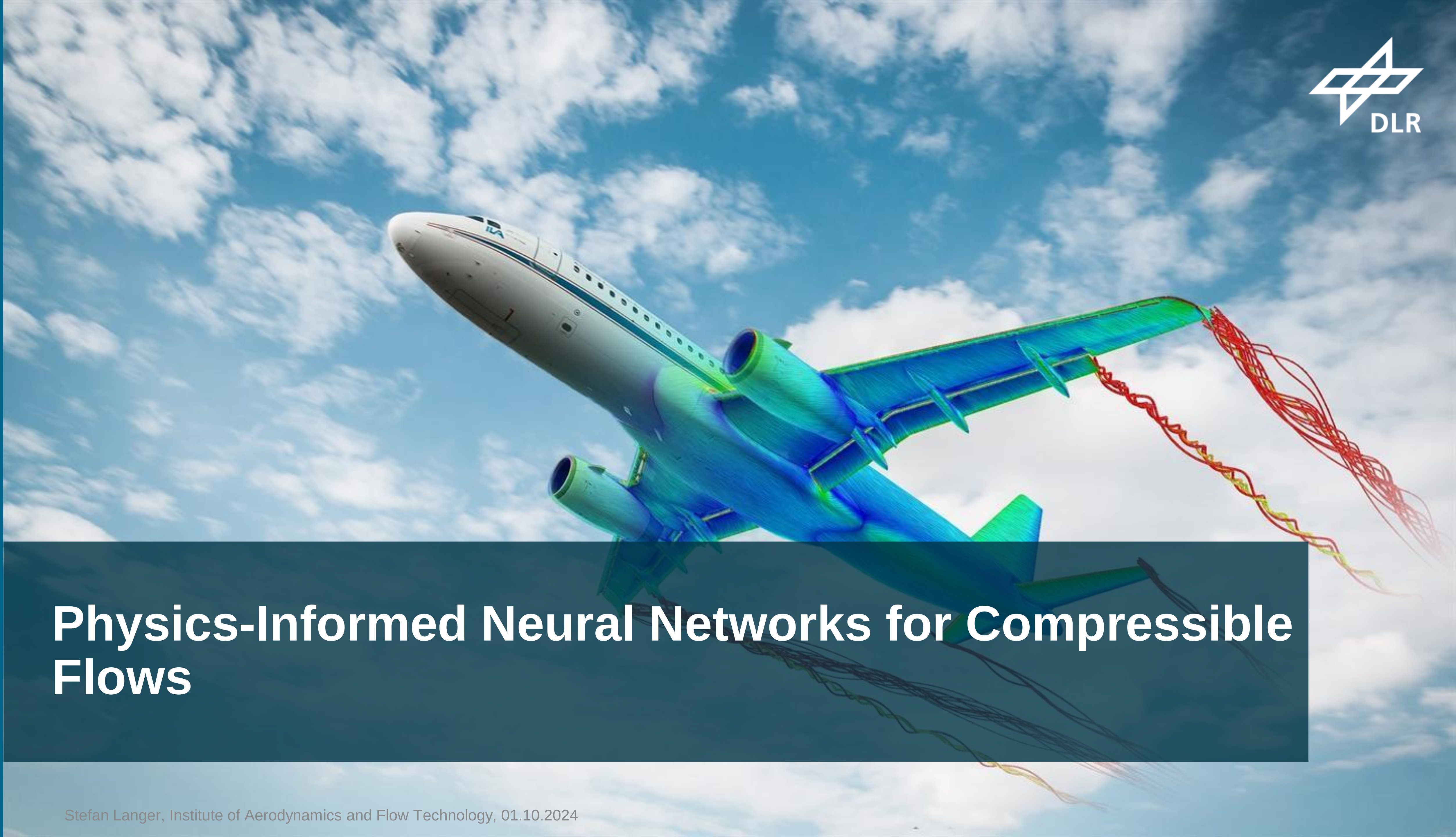


Investigated topics

1. Post-Processing Correction of Coarse Grid **FVM** Solutions
2. Post-Processing Correction of Lower-Order **DG** Solutions
3. Correcting **Unsteady** DG Simulations
4. **External Source Term** Correction and **Reinforcement Learning**

→ Talk of **Anna Kiener**





Physics-Informed Neural Networks for Compressible Flows

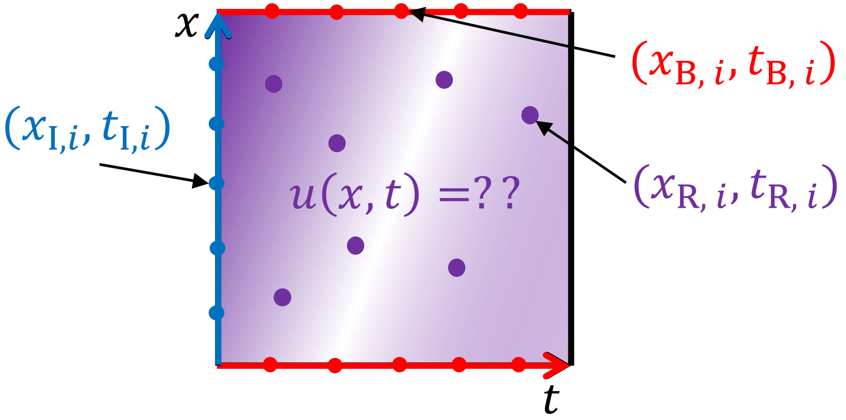
Solving Differential Equations with Physics-Informed Neural Networks

boundary value problem

$\mathcal{D}(u(x,t),x,t) = 0,$ in $\Omega \times [0,T)$

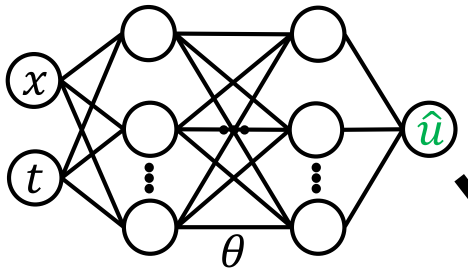
$\mathcal{B}(u(x,t),x,t) = 0,$ on $\partial\Omega \times [0,T)$

$\mathcal{I}(u(x,0),x) = 0,$ in Ω



neural network as global ansatz function for solution

$u(x,t) \approx \hat{u}_\theta(x,t)$



calculate partial derivatives with automatic differentiation

$\partial_x \hat{u}(x_{\mathrm{R},\,i}, t_{\mathrm{R},\,i}),$
$\partial_t \hat{u}(x_{\mathrm{R},\,i}, t_{\mathrm{R},\,i}),$
$\partial_x \hat{u}(x_{\mathrm{B},\,i}, t_{\mathrm{B},\,i}),$
$\ldots$

training

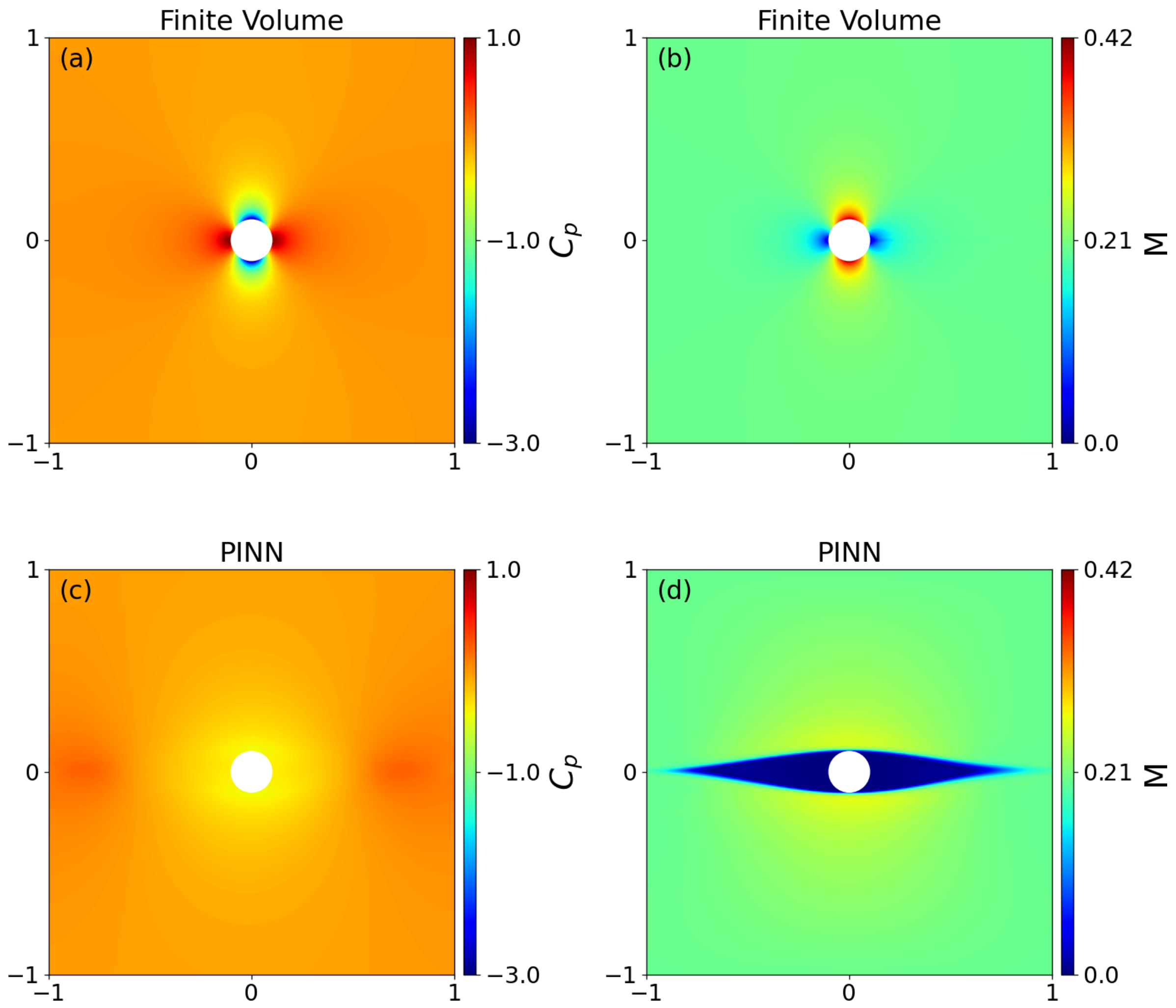
find optimal parameters for network
≙
find solution

$\arg_\theta(\min(\mathcal{L}(\hat{u}_\theta)))$

$$\mathcal{L} = \iint\limits_{\Omega\times(0,T)} \mathcal{D}(\hat{u})^2 \mathrm{d}x\,\mathrm{d}t + \iint\limits_{\partial\Omega\times(0,T)} \mathcal{B}(\hat{u})^2 \mathrm{d}s\,\mathrm{d}t + \int\limits_{\Omega} \mathcal{I}(\hat{u})^2 \mathrm{d}x$$

$$\approx \sum_{i=0}^{n} \mathcal{D}\big(\hat{u}(x_{\mathrm{R},\,i}, t_{\mathrm{R},\,i})\big)^2 + \sum_{i=0}^{n} \mathcal{B}\big(\hat{u}(x_{\mathrm{B},\,i}, t_{\mathrm{B},\,i})\big)^2 + \sum_{i=0}^{n} \mathcal{I}\big(\hat{u}(x_{\mathrm{I},i}, t_{\mathrm{I},i})\big)^2$$

Unphysical Results with Naive PINN Implementation



(a)-(b) reference finite volume solution
(c)-(d) naive PINN results

Naive/direct PINN implementation yields unphysical results!

Stefan Langer, Institute of Aerodynamics and Flow Technology, 01.10.2024

How to avoid unphysical results?

- From the theory of finite volume methods we know that **artificial dissipation** is necessary to stabilize compressible flow simulations
- For PINNs we can add a **dissipative/viscous regularization term** to the PDE
- Similarly to a simplistic scalar valued artificial viscosity in finite volume methods the dissipation is locally scaled based on the spectral radius of the Euler equations

$$\eta = \nu \cdot (a + |\boldsymbol{q}|)$$

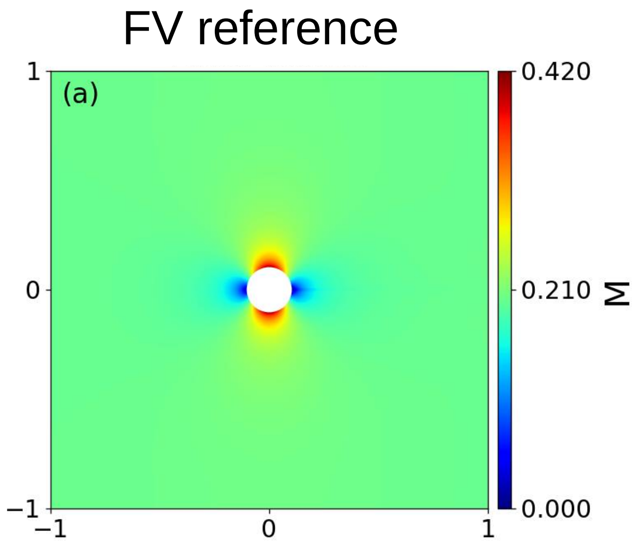
- First, we determine initial ν empirically and reduce during training

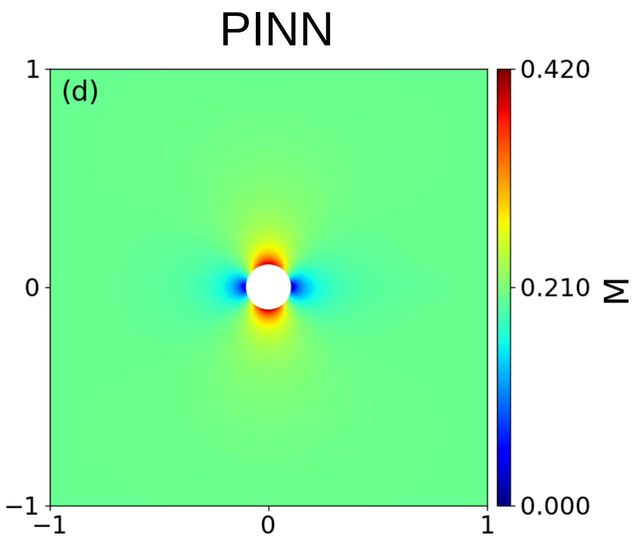
2D Compressible Euler Equations with artificial dissipation:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = \eta \nabla^2 \mathbf{W}$$

$$\boldsymbol{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \boldsymbol{F}_x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho E u \end{pmatrix}, \mathbf{F}_y = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho E v \end{pmatrix},$$

Initial Results with Global Viscosity





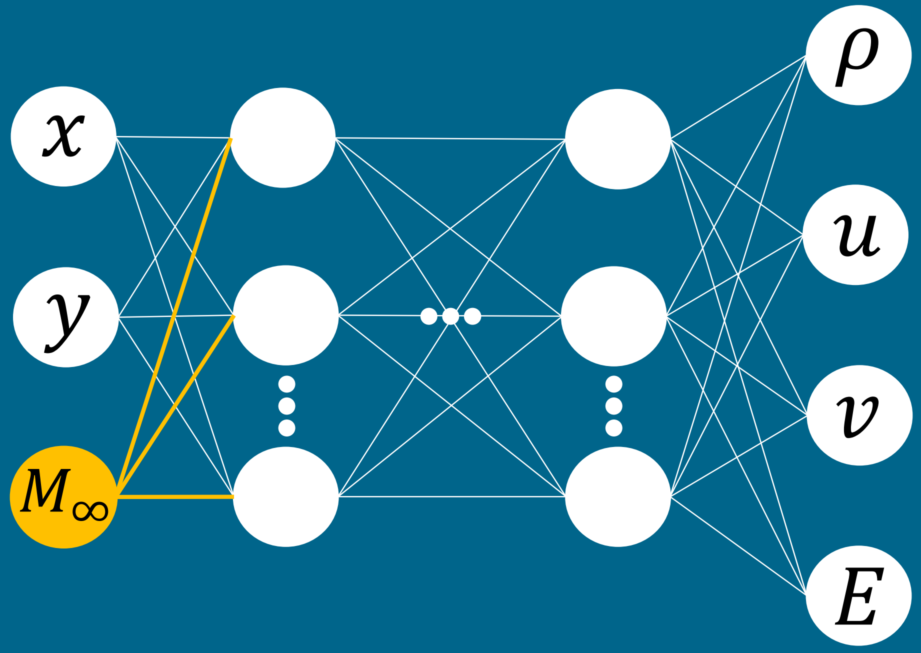
Unphysical results can be avoided!

- Accuracy is still relatively low (e.g. slight asymmetries in solution were observed)
- Training time is of the order of multiple hours
- Once trained, evaluation is possible in < 1s

→ What are possible advantages compared to classical solvers besides QC ?

Extension with Parametric Boundary Conditions

- For parametric problems the parameter can be added to the input space of the NN
- A single PINN is trained in a continuous parameter space



→ Talk of Simon Wassing

Solving Transport Equations on Quantum Computers

Solving transport equations on quantum computers

Goal: Compare performance of **Physics-Informed Neural Networks (PINN)** and its quantum computing analogon, the **Physics-Informed Quantum Circuit (PIQC).**

Challenge: Very **limited availability and access** to quantum hardware.
→ Computations are performed on a **simulator**
→ Simplistic 1D problems

Outline:

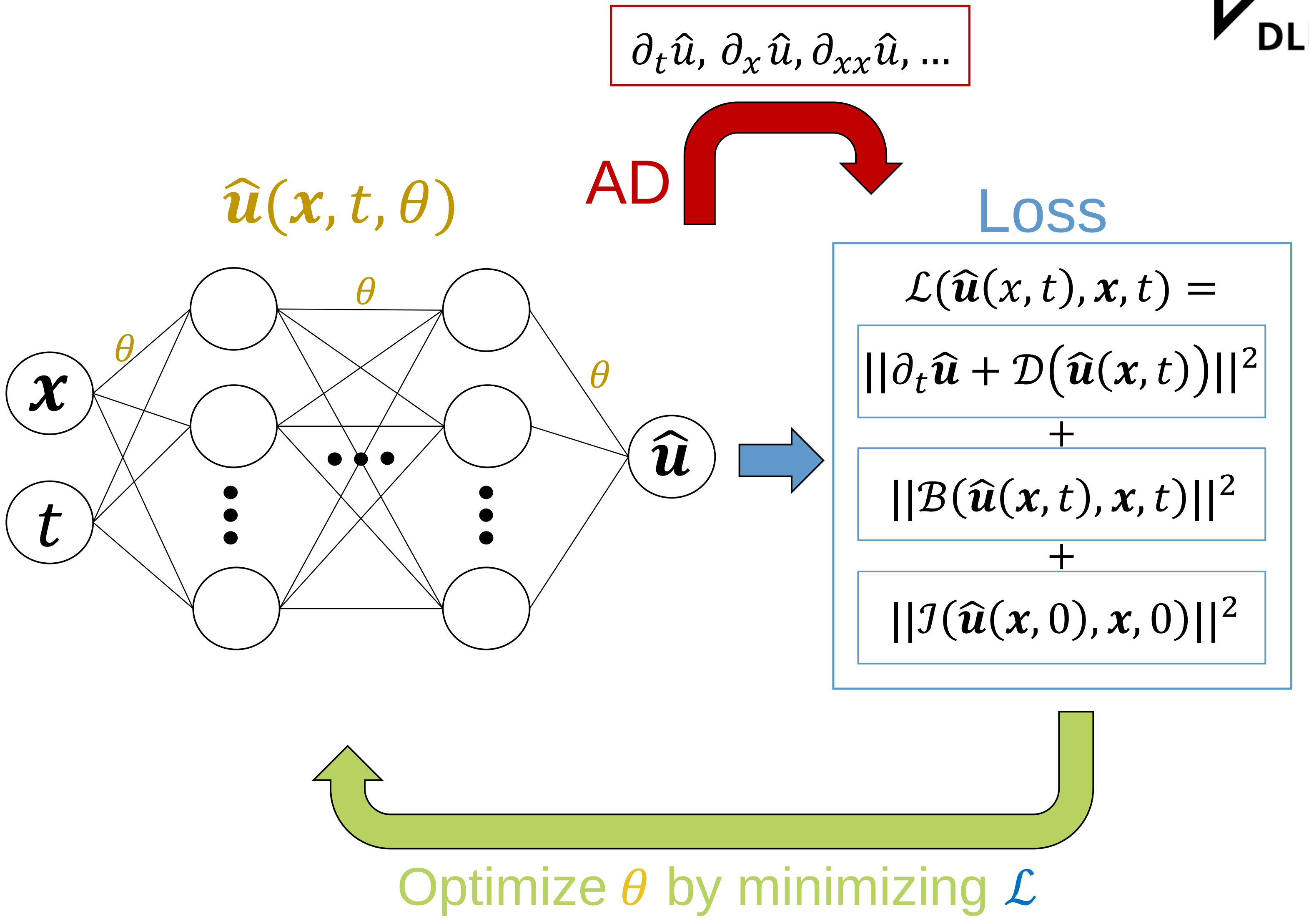
- PINNs and PIQCs
- Experimental Setup
- Results

Physics-Informed Neural Network (PINN)

Boundary Value Problem:

$$\partial_t u + \mathcal{D}(u) = 0, \quad \text{in } \Omega \times [0, T)$$
$$\mathcal{B}(u(x,t), x, t) = 0, \quad \text{in } \partial\Omega \times [0, T)$$
$$\mathcal{J}(u(x,t), x, t) = 0, \quad \text{in } \Omega$$

- $\hat{u}(x,t,\theta)$ approximates solution $u(x,t)$
- Loss terms measure the agreement with PDE, initial cond. $\mathcal{J}$ and boundary cond. $\mathcal{B}$
- Automatic Differentiation (AD) is used to calculate derivatives in loss
- Network parameters θ are optimized based on the calculated loss



Physics-Informed Quantum Circuit (PIQC)

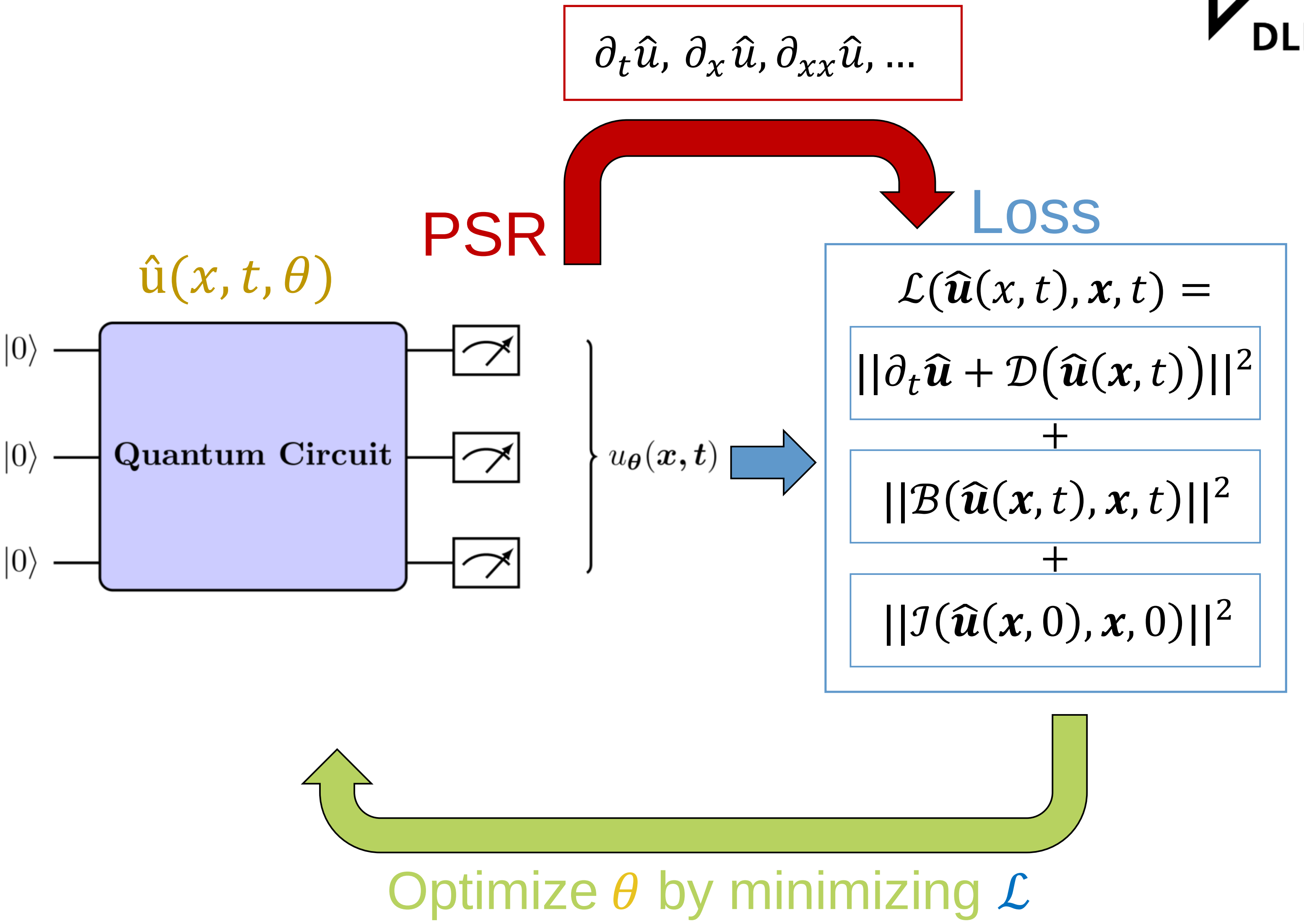
Boundary Value Problem:

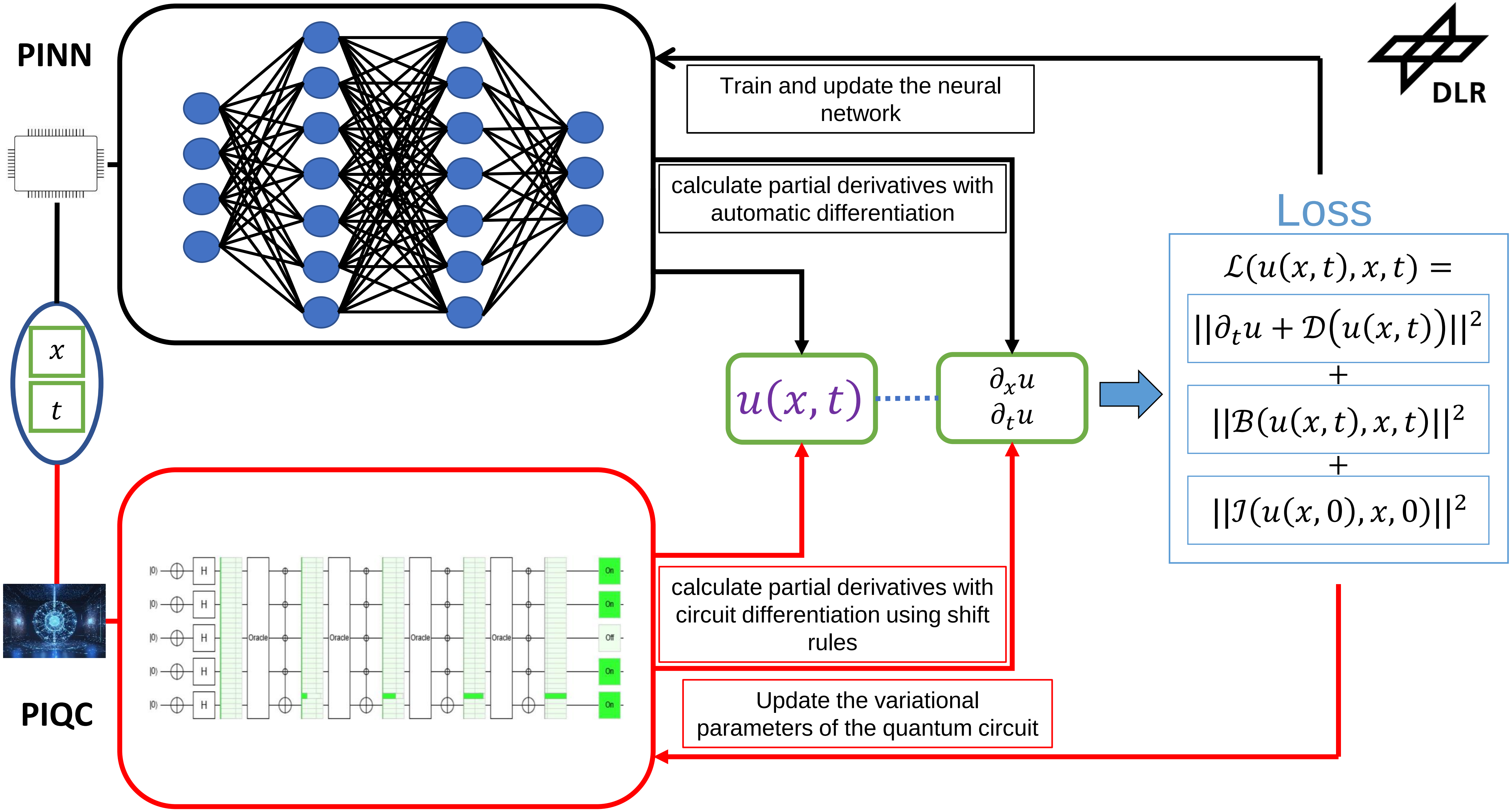
$\partial_t u + \mathcal{D}(u) = 0, \;\text{ in } \Omega \times [0,T)$

$\mathcal{B}(u(x,t),x,t) = 0, \;\text{ in } \partial\Omega \times [0,T)$

$\mathcal{I}(u(x,t),x,t) = 0, \;\text{ in } \Omega$

- $\hat{u}(x,t,\theta)$ approximates solution $u(x,t)$
- Loss terms measure the agreement with PDE, initial cond. $\mathcal{I}$ and boundary cond. $\mathcal{B}$
- Parameter Shift Rule (PSR) is used to calculate derivatives in loss
- Network parameters θ are optimized based on the calculated loss



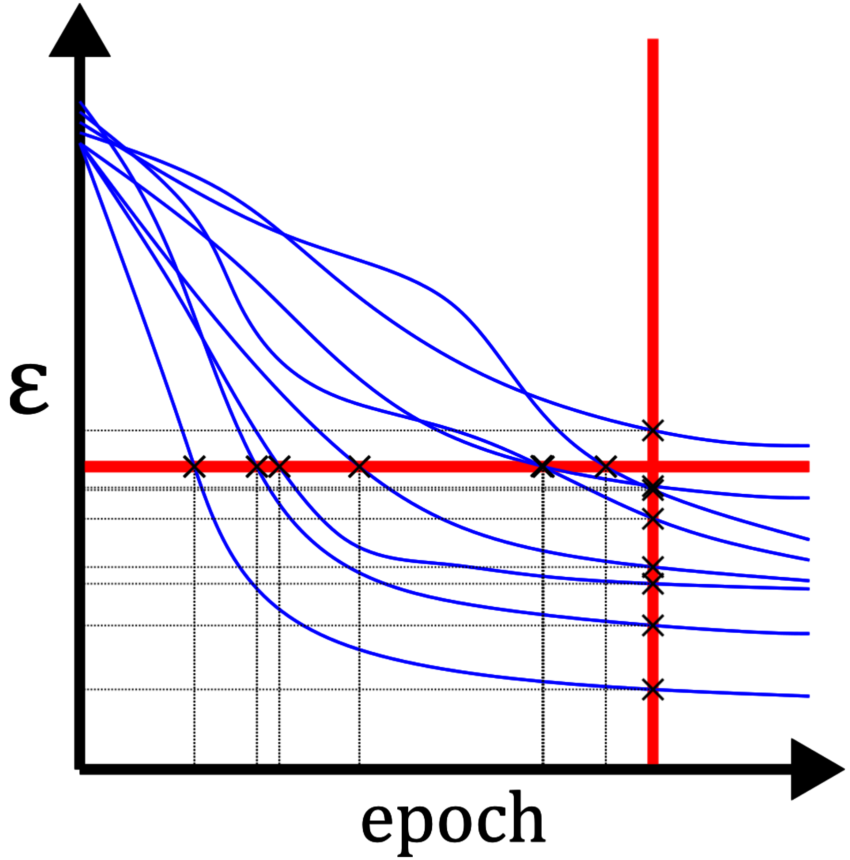

PINN
Train and update the neural network
calculate partial derivatives with automatic differentiation
Loss
$\mathcal{L}(u(x,t),x,t) =$
$||\partial_t u + \mathcal{D}(u(x,t))||^2$
$+$
$||\mathcal{B}(u(x,t),x,t)||^2$
$+$
$||\mathcal{I}(u(x,0),x,0)||^2$
x
t
$u(x,t)$
$\partial_x u$
$\partial_t u$
H
Oracle
On
Off
calculate partial derivatives with circuit differentiation using shift rules
Update the variational parameters of the quantum circuit
PIQC
DLR

How we compare Accuracy and Convergence

- Repeating runs for varying quantum circuit size and neural network shape
- Consider the **mean absolute error** ε:

$$\varepsilon = \frac{1}{N_{val}} \sum_{i=1}^{N_{val}} |\hat{u}_\theta(x_i, t_i) - u(x_i, t_i)|$$

- Compare PINNs and PIQC by measuring:
  1) No. of epochs required to reach a threshold value of ε
  2) Error ε reached after set number of epochs
- Use same optimization parameters (e.g. learning rate) for fair comparison



Considered Problems

Problem I: Ordinary Differential Equation

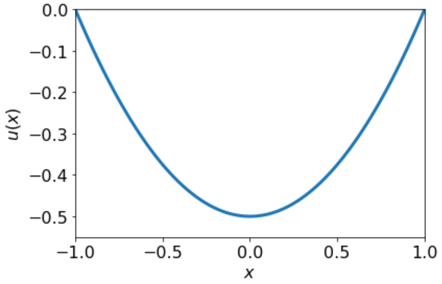
$$\frac{d^2u}{dx^2} = 1 \qquad x \in [-1, 1]$$

Boundary Conditions:

$$u(x = -1) = u(x = 1) = 0$$

Analytical Solution:

$$u(x) = \frac{1}{2}(x^2 - 1)$$



Problem II: Linear Transport of a Gauss Pulse

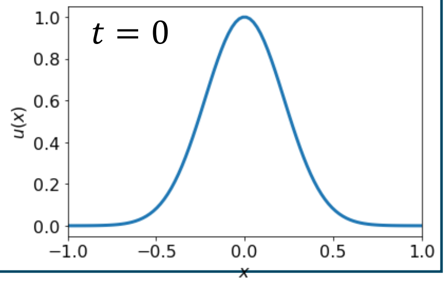
$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \qquad x \times t \in [-1, 1] \times [0, 0.5], \quad c = 0.5$$

Initial Conditions:

$$I(x) = e^{-x^2/0.1}$$

Analytical Solution:

$$u(x, t) = I(x - ct)$$



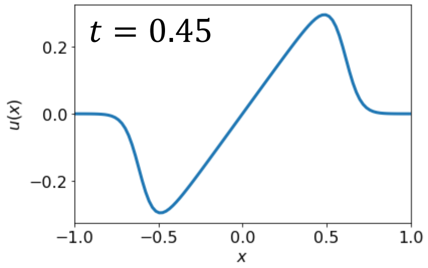
Problem III: Burgers Equation (Non-Linear)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \vartheta\frac{\partial^2 u}{\partial x^2} \qquad x \times t \in [-1, 1] \times [0, 1], \quad \vartheta = 0.01$$

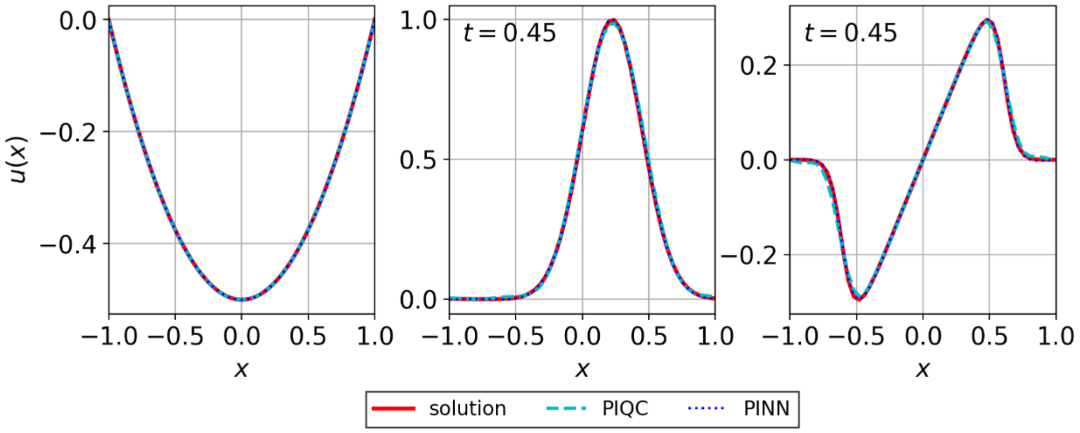
Analytical Solution:

$$u(x, t) = \frac{\frac{x}{t+1}}{1 + \sqrt{\frac{t+1}{t_0}} e^{\frac{x^2}{4\vartheta(t+1)}}}$$

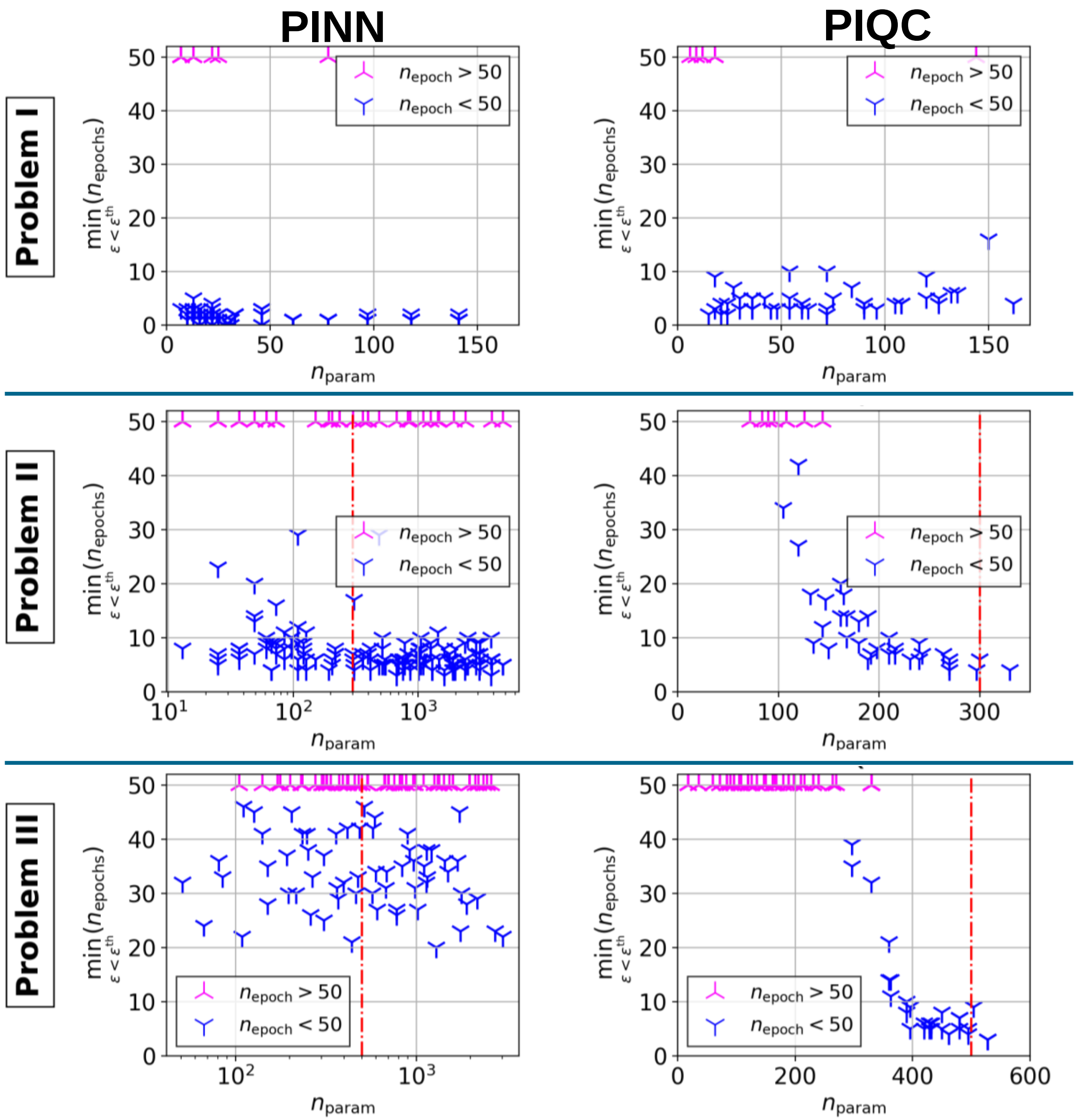
$$t_0 = e^{\frac{1}{8\vartheta}}$$



Problems I-III can be well represented by PINNs and PIQCs



Results: Number of Epochs



- Both approaches need **< 10 epochs** for most runs
- PINNs converge slightly faster

- PIQCs show **clear trend** with n_{param}
- PINN has **training inconsistencies** over whole range of n_{param}

- PIQC shows **faster convergence** as soon as $n_{\text{param}} > 350$.

Summary

- PINNs and PIQCs reach similar accuracies
- PIQCs depend more on n_{param}
- PIQCs converge more consistently for problem II & III
- PIQCs converge faster for large enough n_{param} for problem III
- The PIQC simulation is very time-consuming, allowing only for simple problems

Summary and outlook

Software and Simulation Framework, availability in SMARTy

SMARTy

The **S**urrogate **M**odeling for **AeR**o data **T**oolbox P**y**thon Package [1]

NN Module

PyTorch
Deep learning framework

PyTorch L-BFGS

Gradients & NNs/circuits

Loss function & Optimization

QC Module

PennyLane
Quantum machine learning framework

PyTorch L-BFGS

- Capabilities developed in the project are available in SMARTy and can be used
  - Physics-informed Neural Networks
  - Correction of Coarse Grid Solutions
  - Quantum Circuits
- Interactive Workshops on some topics have been provided
- Reach out to Anna Kiener, Simon Wassing or smarty@dlr.de for further information

What‘s next?

ToQuaFlics

ML4PDE — 01.04.2021 – 31.03.2024

ToQuaFlics — 01.01.2024 – 31.12.2026

XX.XX.2(0)XX

I'm sure it will be on a Monday, but please don't ask me about the year!

What‘s next?

ML4PDE → ToQuaFlics

01.04.2021 – 31.03.2024

Funded by
Vorstandsreserve VO-L

01.01.2024 – 31.12.2026

Funded by DLR QC-I

XX.XX.2(0)XX

I'm sure it will be on a Monday, but please don't ask me about the year!

→ **Many challenges, but potential is huge**

Availability of quantum computers - assumptions in BDLI Roadmap 2021

+5 years (2026):

First commercial quantum computers & hybrid quantum HPC methods in use;

middleware products for hybrid use of QC and established optimization and AI algorithms exist

+10 years (2031):

NISQ computers (Noisy Intermediate-Scale Quantum) achieve **>100 Qbit**;

low-error QC systems

+15-20 years (2036-2040):

Error-corrected, universally programmable quantum computers in commercial use

Various applications in the field of aviation

Reality check:

2021:
IBM quantum processor „Eagle“ 127-qubit
2022:
IBM "Osprey“ 433-Qubit

2023:
- IBM Condor **1.121** qubit
- IBM Heron 133-qubit tunable-coupler QPU (eliminates crosstalk errors)
- IBM Quantum System Two (modular utility-scaled quantum computer system)

IBM roadmap:

2029-2033:
fully error-corrected systems, capable of running 100 million operations in a single quantum circuit in 2029, and a billion operations by 2033 (100,000 physical qubits form 2,000 logical qubits)

Goals of the project

- Perspective how to exploit potential of quantum computers for applications in aerodynamics
- Identification which parts of a solution algorithm can gain from bringing it on a quantum computer
- Estimate how many QuBits are required to solve relevant test cases representing industrial problems
- Investigation of error propagation in Quantum algorithms to formulate requirements on accuracy for hardware

References

- Pia Siegl, Simon Wassing, Markus Mieth, Stefan Langer, Philipp Bekemeyer: **Solving transport equations on Quantum Computers - Potential and limitations of Physics-Informed Quantum Circuits, accepted** in CEAS Journal (will be published soon)
- S. Langer: **Application of the iteratively regularized Gauss-Newton method to parameter identification problems in Computational Fluid Dynamics**, Journal of Computers & Fluids, 284: 104638, 2024, https://doi.org/10.1016/j.compfluid.2024.106438
- S. Wassing, S. Langer, P. Bekemeyer: **Physics-Informed Neural Networks for Parametric Compressible Euler Equations**, Journal of Computers & Fluids, 270:106164. 2024, https://doi.org/10.1016/j.compfluid.2023.106164
- A. Kiener, S. Langer, P. Bekemeyer: **Data-Driven Correction of Coarse Grid CFD Simulations**, Journal of Computers & Fluids, 264: 105971, 2023, https://doi.org/10.1016/j.compfluid.2023.105971
- T. Bode: **The two-particle irreducible effective action, for classical stochastic processes,** Journal of Physics A: Mathematical and Theoretical, 55 (2022) 265401, https://doi.org/10.1088/1751-8121/ac73c6

Thank you for your attention!

stefan.langer@dlr.de

References

- Gaitan, Frank (2024): Circuit implementation of oracles used in a quantum algorithm for solving nonlinear partial differential equations. In: Phys. Rev. A 109 (3). DOI: 10.1103/PhysRevA.109.032604.
- Gaitan, F. Finding flows of a Navier–Stokes fluid through quantum computing. npj Quantum Inf 6, 61 (2020). https://doi.org/10.1038/s41534-020-00291-0
- Oz, Furkan; Vuppala, Rohit K. S. S.; Kara, Kursat; Gaitan, Frank (2022): Solving Burgers' equation with quantum computing. In: Quantum Inf Process 21 (1). DOI: 10.1007/s11128-021-03391-8.
- Harrow, Aram W.; Hassidim, Avinatan; Lloyd, Seth (2009): Quantum algorithm for linear systems of equations. In: Physical review letters 103 (15), S. 150502. DOI: 10.1103/PhysRevLett.103.150502.
- Kyriienko, Oleksandr; Paine, Annie E.; Elfving, Vincent E. (2021): Solving nonlinear differential equations with differentiable quantum circuits. In: Phys. Rev. A 103 (5). DOI: 10.1103/PhysRevA.103.052416.
- Sanavio, Claudio; Succi, Sauro (2024): Quantum computing for simulation of fluid dynamics. DOI: 10.5772/intechopen.1005242.
- Peddinti, Raghavendra Dheeraj; Pisoni, Stefano; Marini, Alessandro; Lott, Philippe; Argentieri, Henrique; Tiunov, Egor; Aolita, Leandro (2024): Quantum-inspired framework for computational fluid dynamics. In: Commun Phys 7 (1). DOI: 10.1038/s42005-024-01623-8.
- Bharadwaj, Sachin S.; Sreenivasan, Katepalli R. (2023): Hybrid quantum algorithms for flow problems. In: Proceedings of the National Academy of Sciences of the United States of America 120 (49), e2311014120. DOI: 10.1073/pnas.2311014120.