HYBRID METHODS FOR POISSON AND STOKES

Franziska Griese (SC-HPC)

8th WAW ML, Jena 2022

DLR

Motivation

- Technical systems are becoming more and more complicated, making modeling them more complex and expensive

→ use NN's to build surrogate models

- By data-driven approaches some natural laws are not or only poorly considered

Conservation of

Energy | Mass | Momentum

→ Hybrid methods, which combine physical knowledge and NN's

FENN (Finite Element Neural Network)

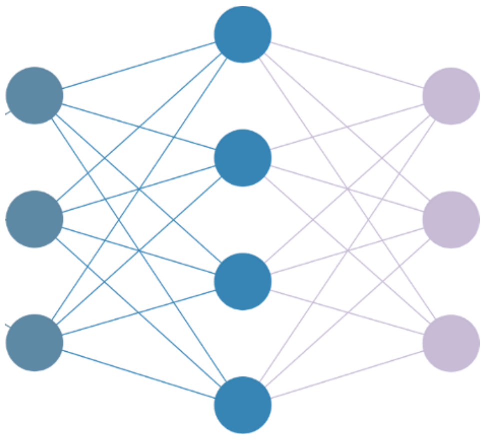
- Combines classical finite element method with a NN
- Use best of both worlds

FEM	NN
Sparsity patterns: because of locality of elements	Fast prediction after training
No multi-objective optimization → naturally including BC,...	Generalizable
Numerical theory of errors can be used	

- Physical knowledge in form of a PDE is used

FENN

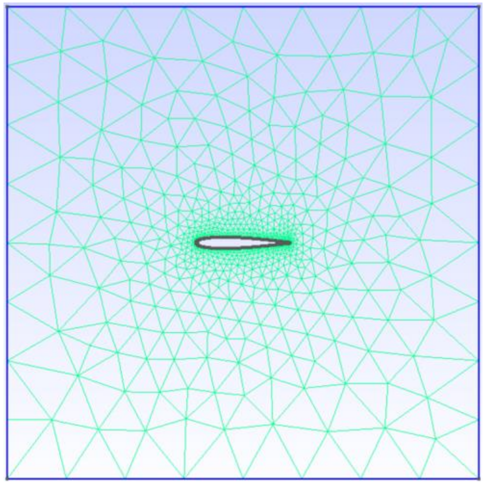
Parameter(s)
λ



Prediction for
u^h

Variable(s) x

$$Loss = (K \cdot u^h - F)^2$$

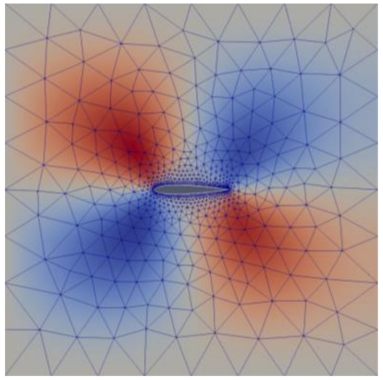


FEM: Galerkin system

$$\underbrace{\begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,n} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ k_{n,1} & k_{n,2} & \cdots & k_{n,n} \end{pmatrix}}_{K(u^h)} \underbrace{\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}}_{u^h} = \underbrace{\begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{pmatrix}}_{F}$$

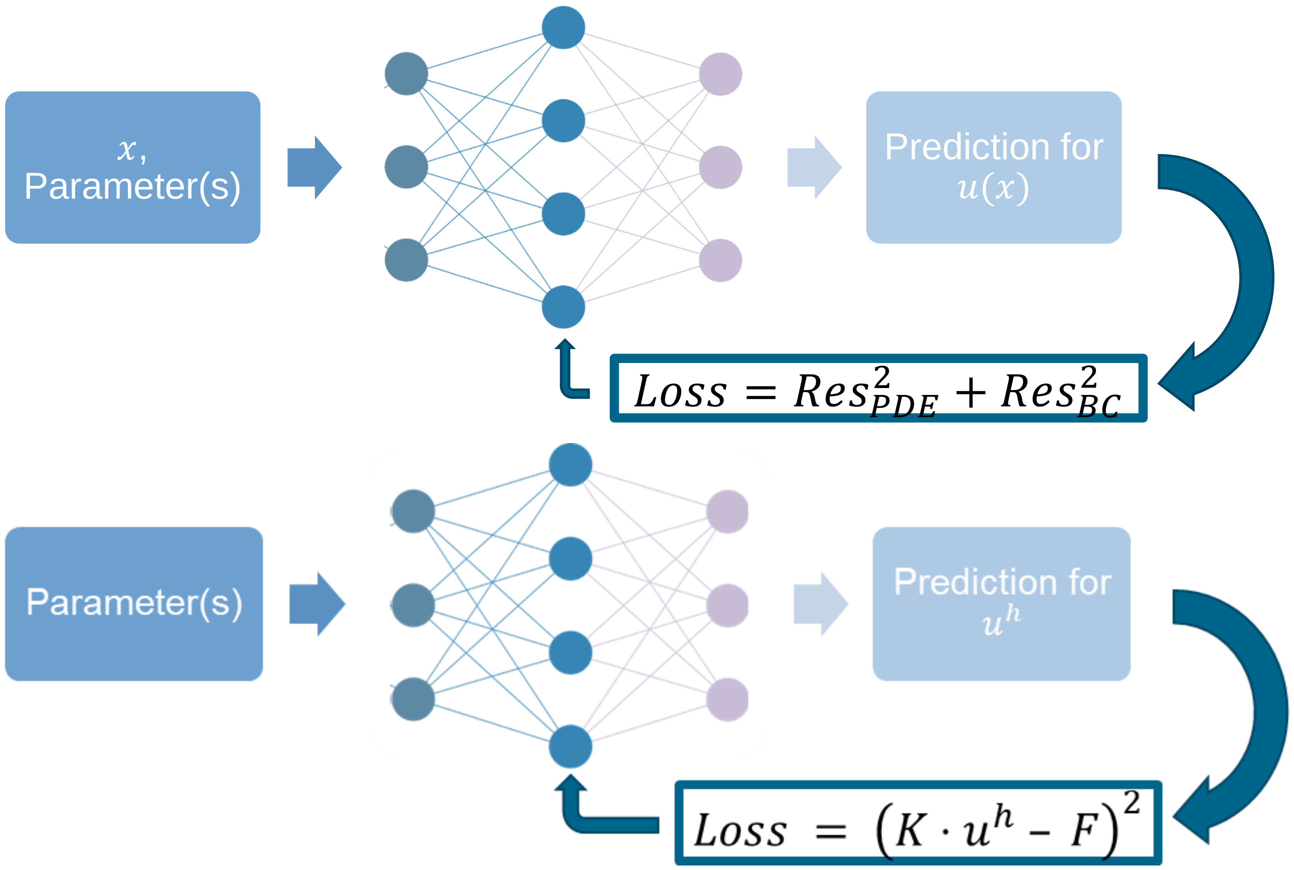
- Calculate:

$$u(x) = \sum_{j=1}^{n} u_j^h \cdot \phi_j(x)$$



Comparison to PINN's

- PINN:
- FENN:



FENN **advantages**:
- Numerical theory of errors can be used
- Sparsity patterns
- No multi-objective optimization

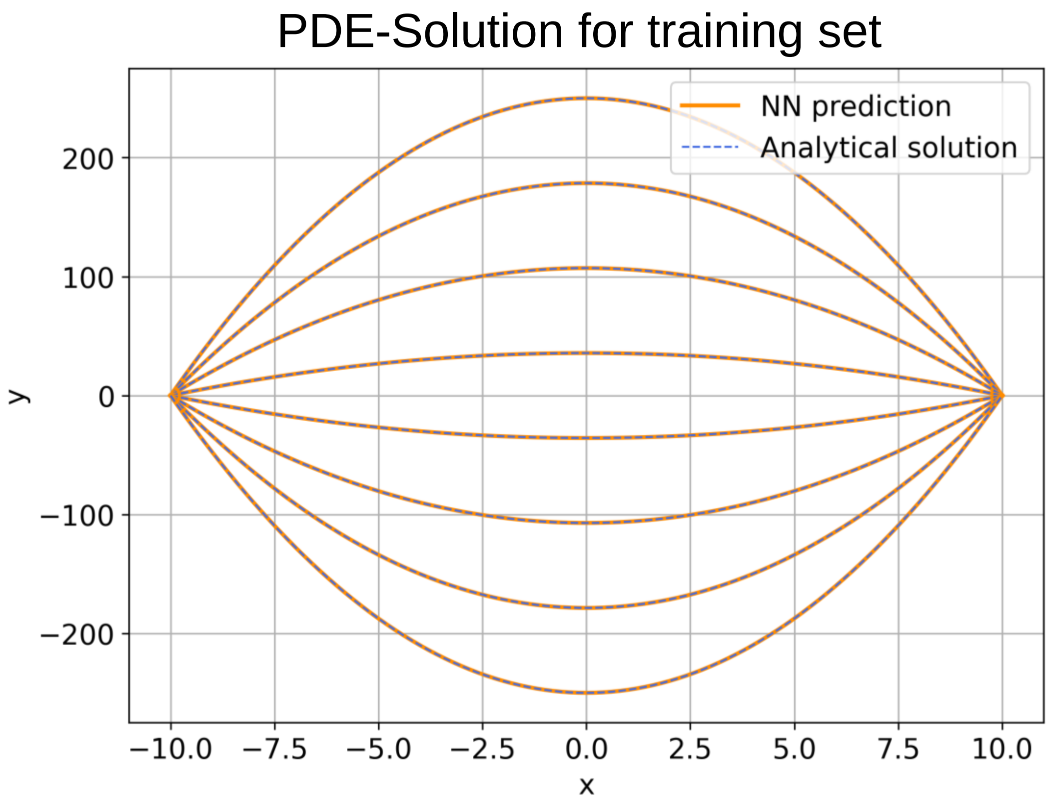
- Main difference: loss function
  - PINN: residuum of PDE using automatic differentiation
  - FENN: residuum of FEM's Galerkin system

1D Poisson Problem

- PDE:

$$\frac{\partial^2 u}{\partial x^2} = f \text{ with } f = \lambda, \lambda \in \mathbb{R}$$

- BC: $u(-10) = u(10) = 0$



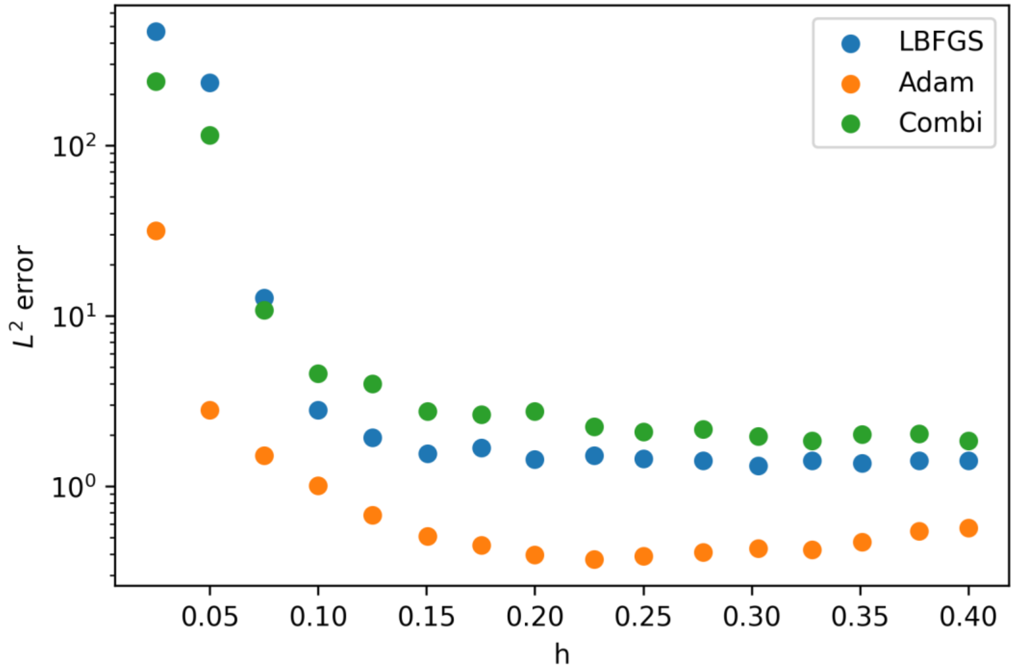
FENN - Poisson

- First issue: loss converges to local minima
  - K^2 squares condition, use loss function with residuum as minimum

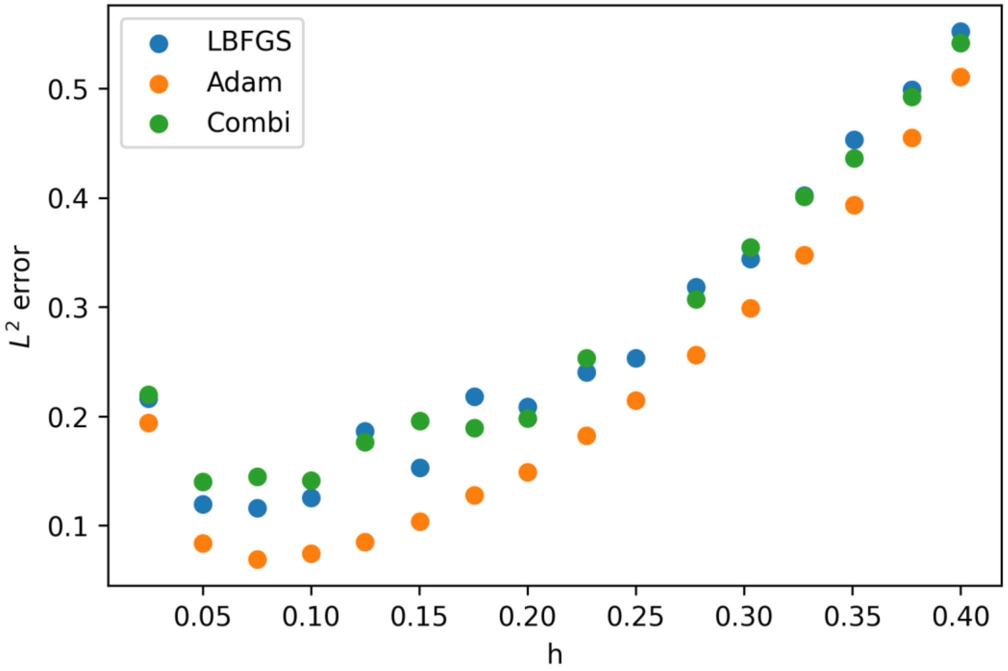
$$\left(K \cdot u^h - F\right)^2$$
$$\rightarrow {u^h}^T \cdot (0.5 \cdot K \cdot u^h - F)$$

- Second issue: bad condition of stiffness matrix K
  - Preconditioning with Cholesky decomposition $K = L \cdot L^T$

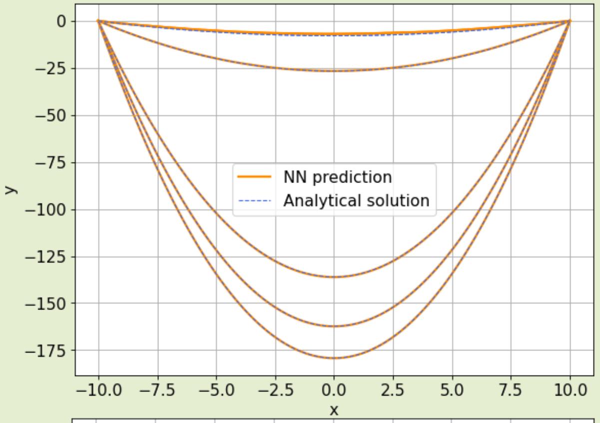
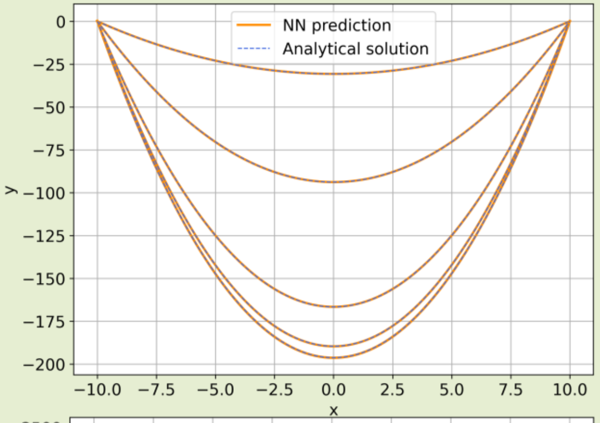
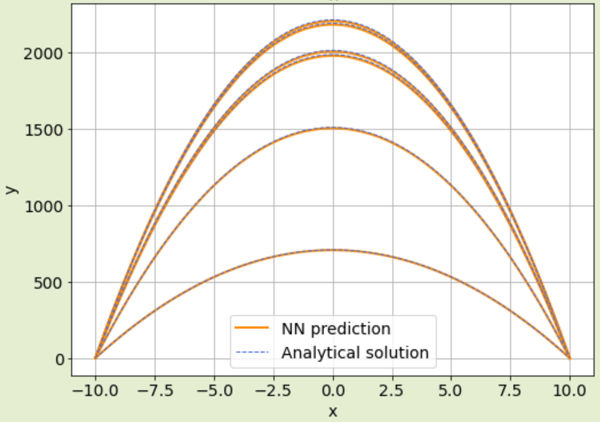
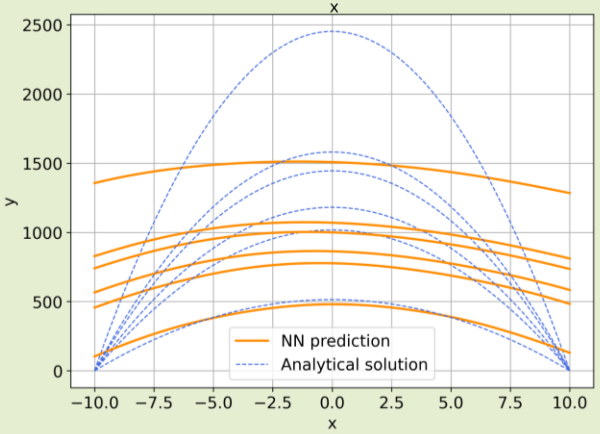
Without Preconditioning



With Preconditioning



Poisson – Comparison FENN & PINN

	FENN	PINN
Convergence	Good results for different NN-architectures and optimizers	
Training time	~ 1 sec	~ 3 min
Generalization Inside training interval		
Outside training interval		

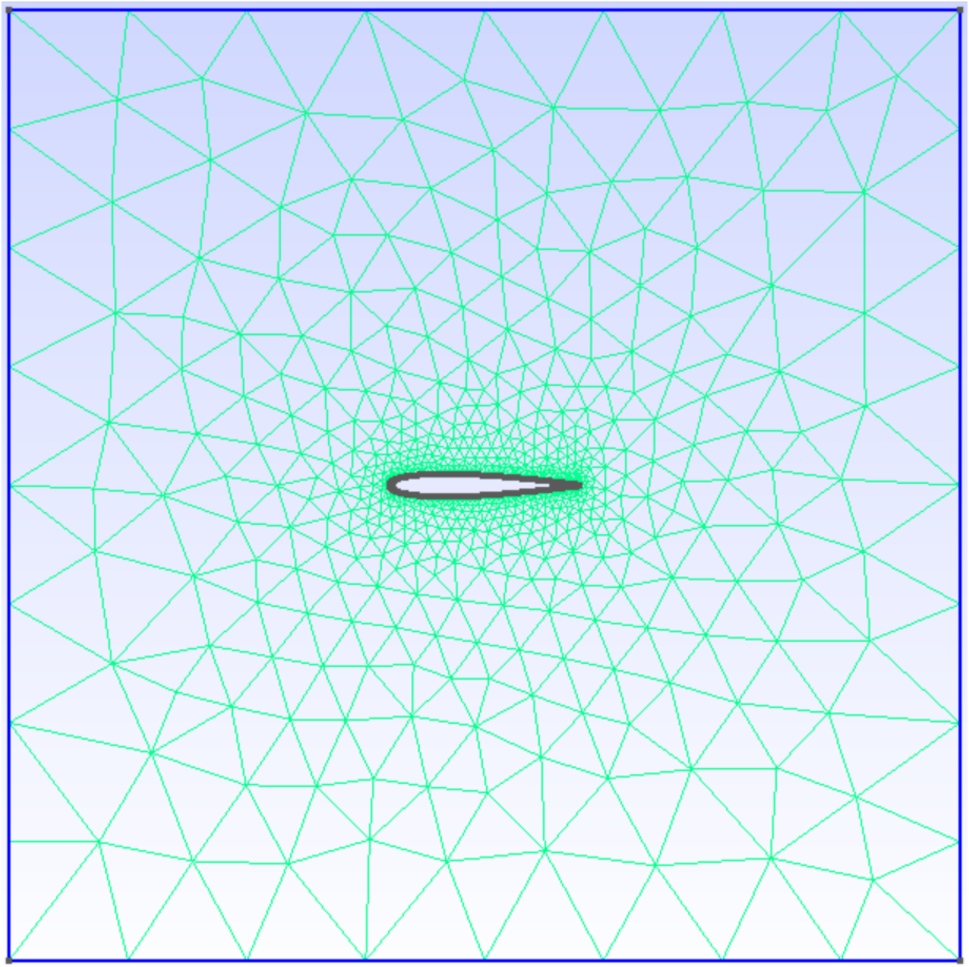
2D Stokes flow around an airfoil

- PDE:

$$\nabla \cdot u = 0,$$
$$\nabla p - \Delta u = 0,$$

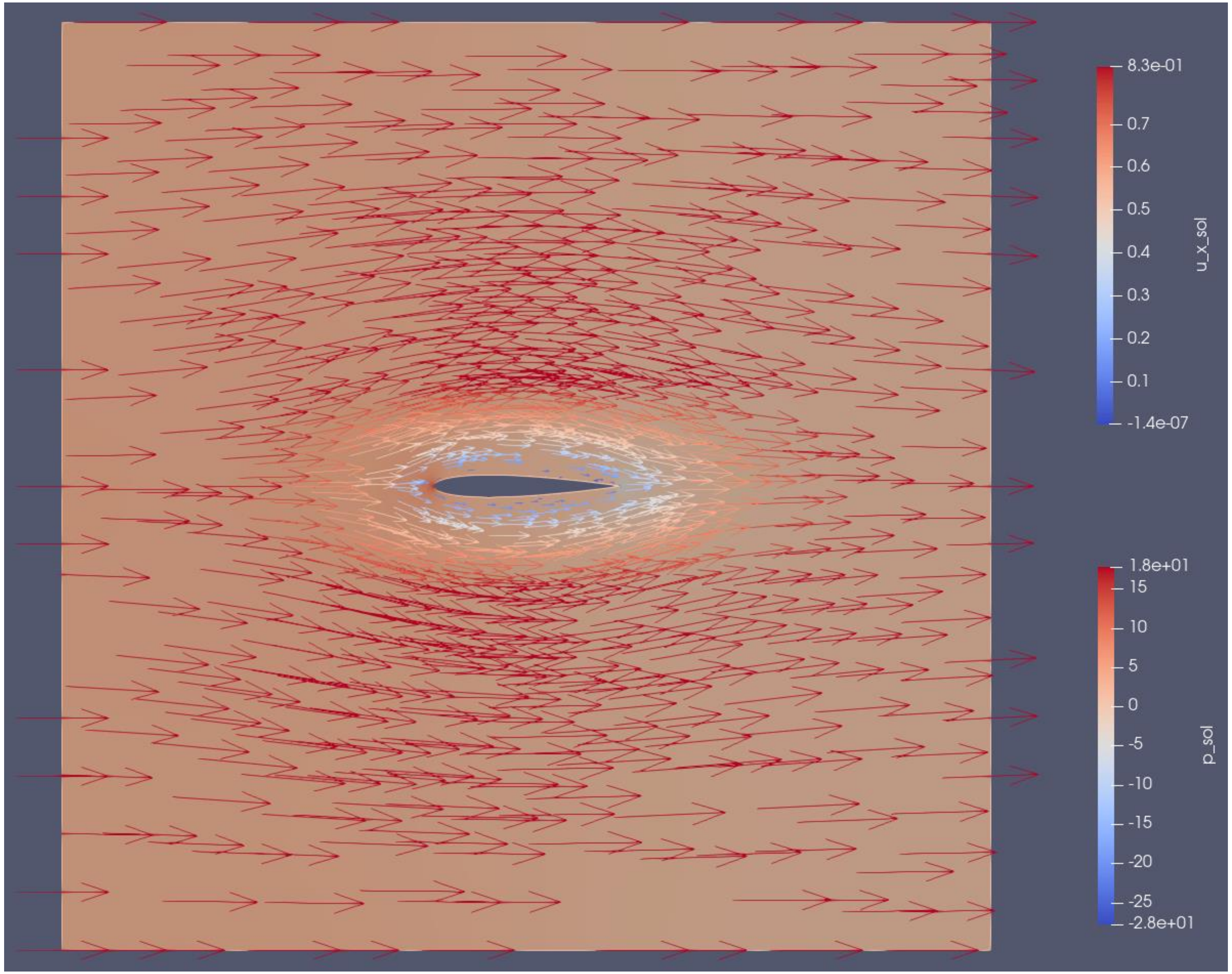
- with u the velocity and p the pressure
- BC: Dirichlet at left, top, bottom; Neumann at right; no-slip at airfoil

- Saddle point problem
- Even for FEM no easy problem



Mesh around NACA 0012 airfoil.

2D Stokes flow around an airfoil



- Constructed **fully differentiable** FEM-Solver for Stokes in PyTorch
- Used Taylor-Hood elements to construct Galerkin system

FENN - Stokes

- Ongoing process
- Indefinite stiffness matrix → both tricks used for Poisson equation don’t work
- Other preconditioners are needed
- But system differentiable for every possible parameter

Outlook

- Solve Stokes flow around airfoil with FENN and PINN
- Airfoil with parameterizable angle of attack
- Inverse problems e.g. state estimation: sensors at airfoil want to measure angle of attack
- Uncertainty Quantification

Thank you for your attention!
Questions?

Contact: Franziska.Griese@dlr.de