TU Delft

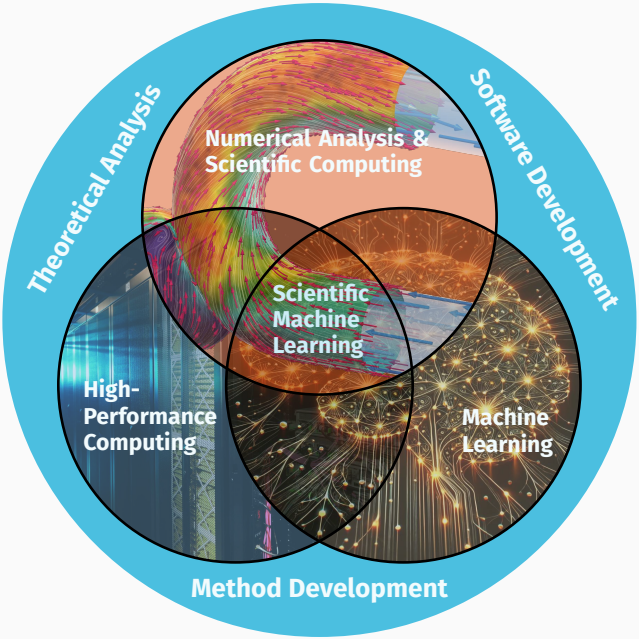
Some Algorithmic Advances in Cardiovascular Simulations

Alexander Heinlein[1]

Interdisciplinary Aneurysm Science Workshop (IASW2025), July 29, 2025

[1]Delft University of Technology

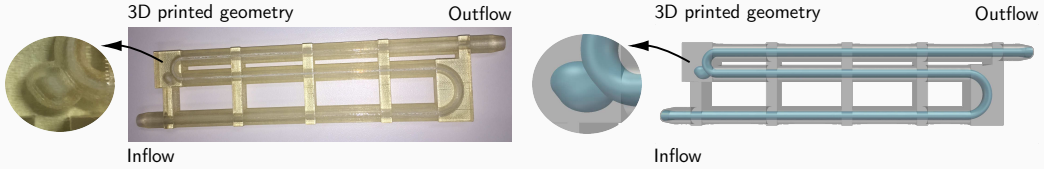
SCaLA – Scalable Scientific Computing and Learning Algorithms



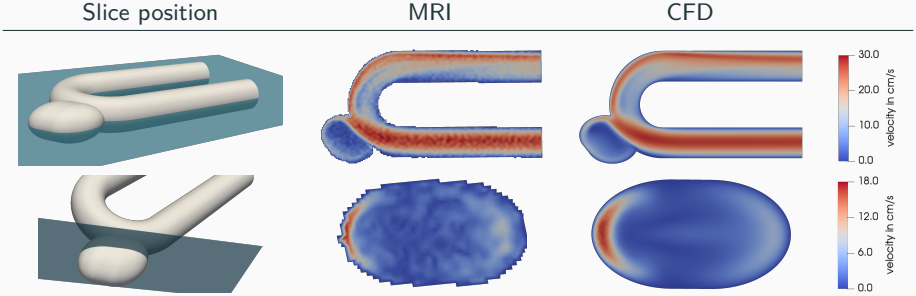
Surrogate models for aneurysm simulations

Computational Fluid Dynamics (CFD) Simulations are Time Consuming

In **Giese, Heinlein, Klawonn, Knepper, Sonnabend (2019)**, a benchmark for **comparing MRI measurements and CFD simulations** of hemodynamics in **intracranial aneurysms** was proposed.

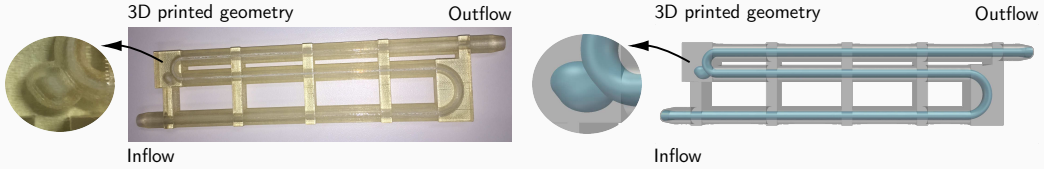


To obtain accurate simulation results, a simulation with $\approx 10\,\mathrm{m}$ d.o.f.s has been carried out. On $O(100)$ MPI ranks, the computation of a steady state took $O(1)\,\mathrm{h}$ on CHEOPS supercomputer at UoC.

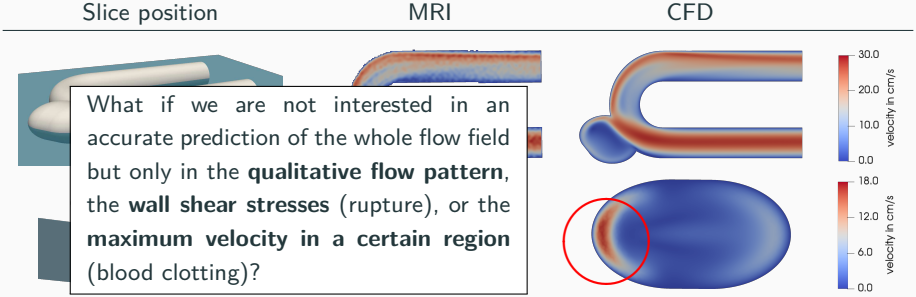


Computational Fluid Dynamics (CFD) Simulations are Time Consuming

In **Giese, Heinlein, Klawonn, Knepper, Sonnabend (2019)**, a benchmark for **comparing MRI measurements and CFD simulations** of hemodynamics in **intracranial aneurysms** was proposed.



To obtain accurate simulation results, a simulation with $\approx 10\,\mathrm{m}$ d.o.f.s has been carried out. On $O(100)$ MPI ranks, the computation of a steady state took $O(1)\,\mathrm{h}$ on CHEOPS supercomputer at UoC.

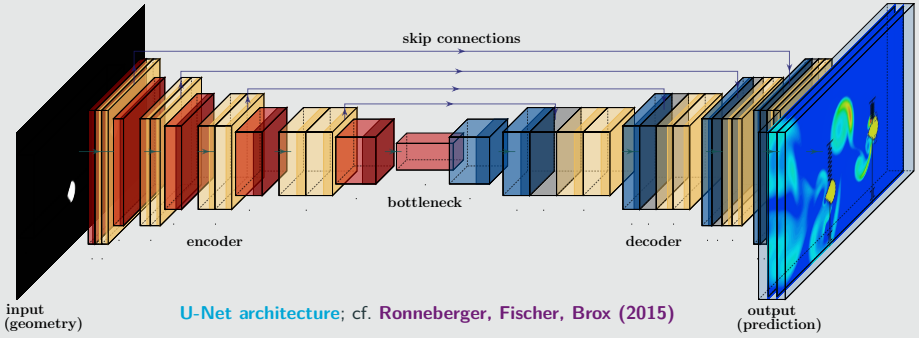


What if we are not interested in an accurate prediction of the whole flow field but only in the **qualitative flow pattern**, the **wall shear stresses** (rupture), or the **maximum velocity in a certain region** (blood clotting)?

Convolutional Neural Network-Based Surrogate Model

CNN-based approach

We employ a **convolutional neural network (CNN)** (**LeCun (1998)**) to predict the stationary flow field, given an **image of the geometry as input**.



U-Net architecture; cf. **Ronneberger, Fischer, Brox (2015)**

Related works (non-exhaustive)

- **Guo, Li, Iorio (2016)**
- **Niekamp, Niemann, Schröder (2022)**
- **Stender, Ohlsen, Geisler, Chabchoub, Hoffmann, Schlaefer (2022)**

Operator learning (non-exhaustive)

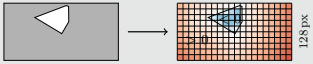
- **FNOs**: **Li et al. (2021)**
- **PCA-Net**: **Bhattacharya et al. (2021)**
- **Random features**: **Nelsen and Stuart (2021)**
- **CNOs**: **Raonić et al. (2023)**

Comparison OpenFOAM® Versus CNN (Relative Error 2 %)

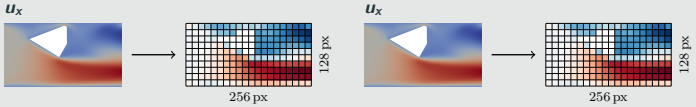
We automatically generate geometries and compute the corresponding flow fields using OpenFOAM®.

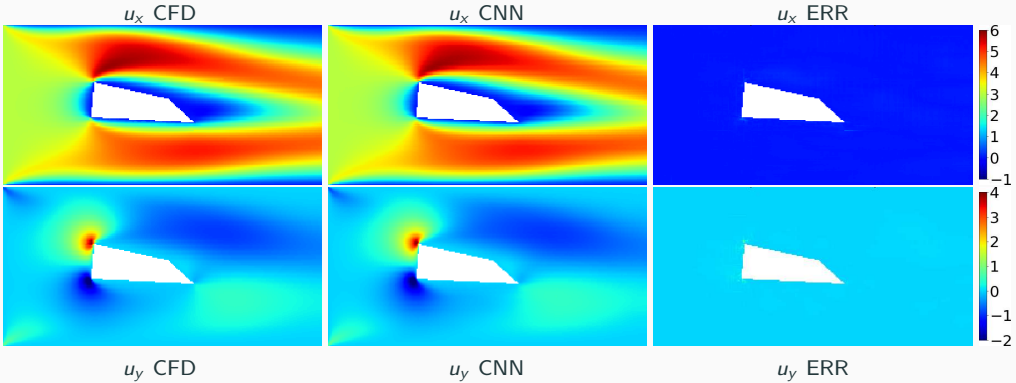
Input data

SDF (Signed Distance Function)



Output data





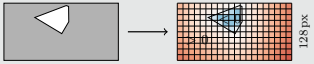
Cf. **Eichinger, Heinlein, Klawonn (2021, 2022)**.

Comparison OpenFOAM® Versus CNN (Relative Error 14 %)

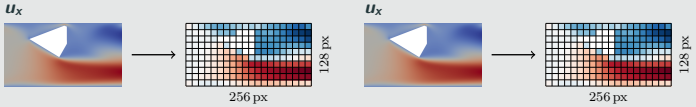
We automatically generate geometries and compute the corresponding flow fields using OpenFOAM®.

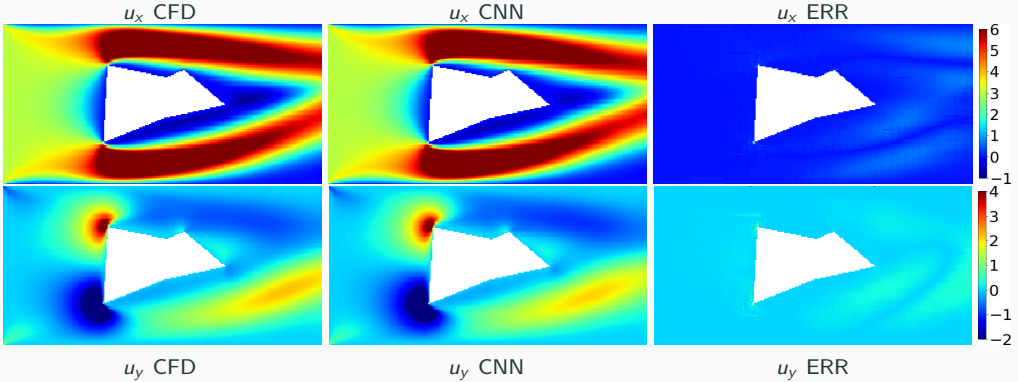
Input data

SDF (Signed Distance Function)



Output data





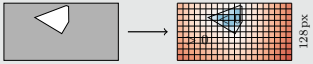
Cf. **Eichinger, Heinlein, Klawonn (2021, 2022)**.

Comparison OpenFOAM® Versus CNN (Relative Error 31 %)

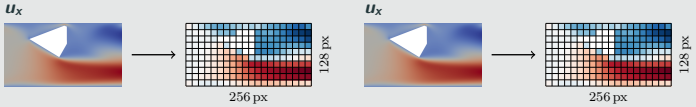
We automatically generate geometries and compute the corresponding flow fields using OpenFOAM®.

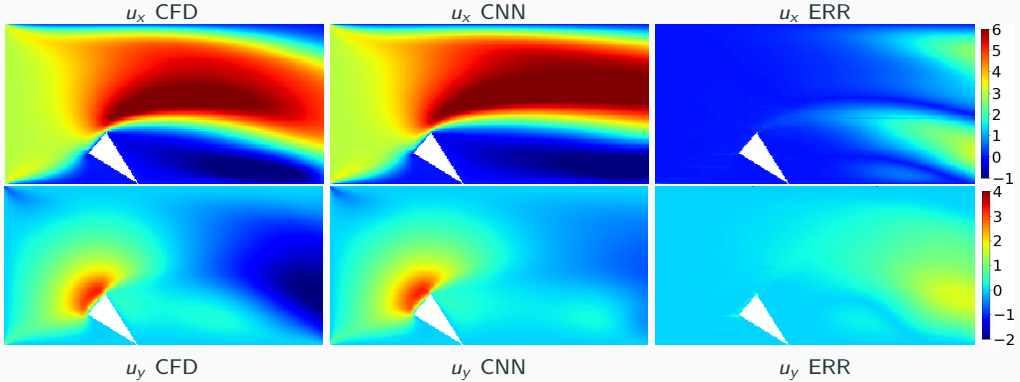
Input data

SDF (Signed Distance Function)



Output data





Cf. **Eichinger, Heinlein, Klawonn (2021, 2022)**.

Computing Times

Data generation:

	avg. runtime per case (serial)
create STL	0.15 s
`snappyHexMesh`	37 s
`simpleFoam`	13 s
total time	$\approx$ 50 s

Training:

	U-Net	
# decoders	1	2
# parameters	$\approx$ 34 m	$\approx$ 53.5 m
time/epoch	**195 s**	**270 s**

Comparison CFD Vs NN:

	OpenFOAM®	**U-Net**	
	CPU	CPU	GPU
avg. time	**50 s**	**0.092 s**	**0.0054 s**

$\Rightarrow$ Flow predictions using neural networks may be less accurate and the **training phase expensive**, but the **flow prediction is** $\approx 5 \cdot 10^2 - 10^4$ **times faster**.

Unsupervised Learning Approach – PDE Loss Using Finite Differences

Physics-informed loss function

We train the CNN by incorporating the **squared PDE residuals** into the **loss function**:

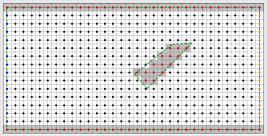
$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \|\mathcal{R}(u_{\text{CNN}}, p_{\text{CNN}})\|^2$$

Here, N_{PDE} is the number of training configs.

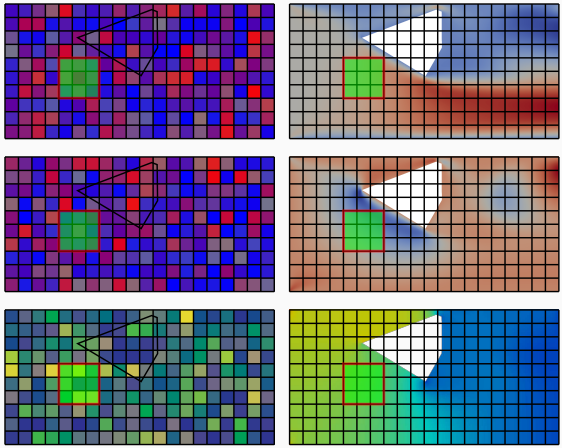
Cf. **Raissi et al. (2019), Dissanayake and Phan-Thien (1994), Lagaris et al. (1998)**.

We discretize the differential operators using **finite differences on the output pixel image**.

Boundary conditions



We explcitly enforce boundary conditions on the output image → **hard constraints**



$\|\mathcal{R}(u_{\text{CNN}}, p_{\text{CNN}})\|^2 >> 0$ $\|\mathcal{R}(u_{\text{CNN}}, p_{\text{CNN}})\|^2 \approx 0$

Here, we consider ther **Navier–Stokes equations**:

$$\mathcal{R}(u_{\text{CNN}}, p_{\text{CNN}}) = \begin{bmatrix} -\nu \Delta \vec{u} + (u \cdot \nabla)\, \vec{u} + \nabla p \\ \nabla \cdot u \end{bmatrix}$$

Cf. **Grimm, Heinlein, Klawonn (2025)**.

Results on $\approx$ 5 000 Geometries – Data-Based Versus Physics-Informed

	training data	error	$\frac{\|u_{NN}-u\|_2}{\|u\|_2}$	$\frac{\|p_{NN}-p\|_2}{\|p\|_2}$	mean residual momentum	mean residual mass	# epochs trained
data-based	10%	train.	2.07%	10.98%	$1.1 \cdot 10^{-1}$	$1.4 \cdot 10^{0}$	500
		val.	4.48 %	15.20 %	$1.6 \cdot 10^{-1}$	$1.7 \cdot 10^{0}$	
	25%	train.	1.93%	8.45%	$9.1 \cdot 10^{-2}$	$1.2 \cdot 10^{0}$	500
		val.	3.49 %	10.70 %	$1.2 \cdot 10^{-1}$	$1.4 \cdot 10^{0}$	
	50%	train.	1.48%	8.75%	$9.0 \cdot 10^{-2}$	$1.1 \cdot 10^{0}$	500
		val.	2.70 %	10.09 %	$1.1 \cdot 10^{-1}$	$1.2 \cdot 10^{0}$	
	75%	train.	1.43%	7.30%	$1.0 \cdot 10^{-1}$	$1.5 \cdot 10^{0}$	500
		val.	**2.52 %**	**8.67 %**	$1.2 \cdot 10^{-1}$	$1.5 \cdot 10^{0}$	
physics-informed	10%	train.	5.35%	12.95%	$3.5 \cdot 10^{-2}$	$7.8 \cdot 10^{-2}$	**5 000**
		val.	6.72%	15.39%	$6.7 \cdot 10^{-2}$	$2.0 \cdot 10^{-1}$	
	25%	train.	5.03%	12.26%	$3.2 \cdot 10^{-2}$	$7.3 \cdot 10^{-2}$	**5 000**
		val.	5.78 %	13.38 %	$5.3 \cdot 10^{-2}$	$1.4 \cdot 10^{-1}$	
	50%	train.	5.81%	12.92%	$3.9 \cdot 10^{-2}$	$9.3 \cdot 10^{-2}$	**5 000**
		val.	5.84 %	12.73 %	$4.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	
	75%	train.	5.03%	11.63%	$3.2 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$	**5 000**
		val.	**5.18 %**	**11.60 %**	$4.2 \cdot 10^{-2}$	$1.1 \cdot 10^{-1}$	

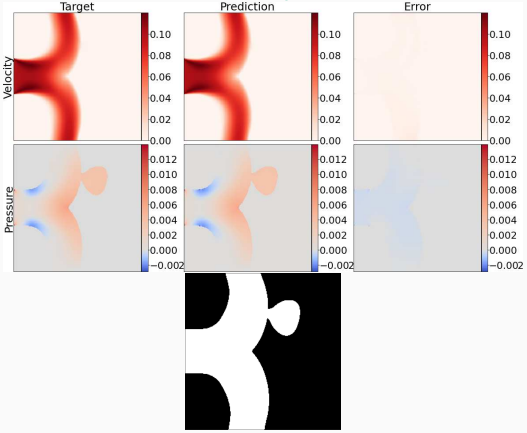
$\rightarrow$ The results for the **physics-informed approach** are **comparable to the data-based approach**; the **errors are slightly higher**. However, no **reference data at all is needed for the training**.

Aneurysm Geometries

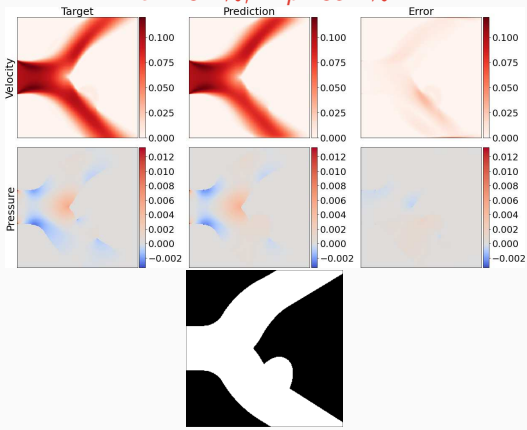
Training: 500 geometries **Validation:** ≈ 1 200 geometries

Relative $L2$**-error** on the **validation data set** in u: **4.9** %, in p: **9.5** %.

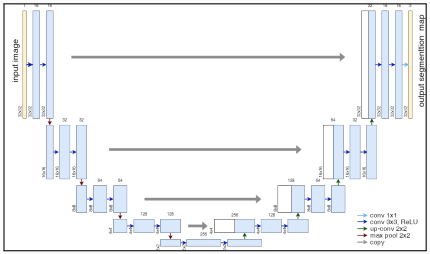
u: 3.3%, p: 7.4%



u: 13.2%, p: 35.1%



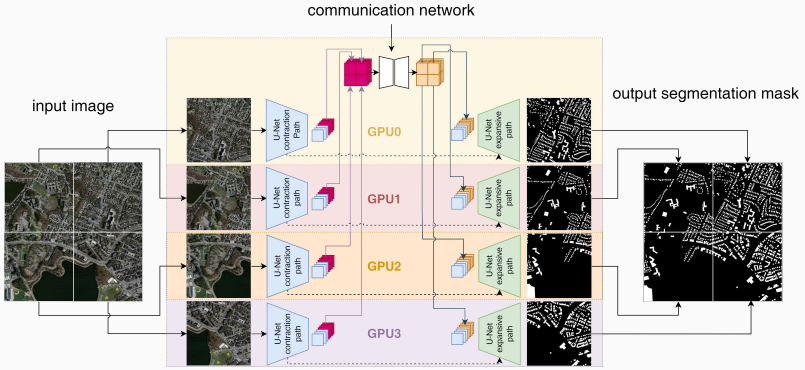
Domain Decomposition-Based U-Net Architecture



name	mem. feature maps		mem. weights	
	# of values	MB	# of values	MB
input block	268 M	**1 024.0**	38 848	**0.148**
encoder blocks	314 M	**1 320**	18 M	**72**
decoder blocks	754 M	**3880**	12 M	**47**
output block	3.1 M	**12.0**	195	**0.001**

Most memory in the **U-Net** is used by **feature maps**, **not weights**
→ **Decompose feature maps** to **distribute memory consumption**.

Cf. **Verburg, Heinlein, Cyr (2025).**



Scalable Solvers for Blood Flow Simulations

FROSch (Fast and Robust Overlapping Schwarz) Framework in Trilinos

Software

- Object-oriented C++ domain decomposition solver framework with MPI-based distributed memory parallelization
- Part of Trilinos with the parallel linear algebra based on Tpetra
- Node-level parallelization and performance portability on CPU and GPU architectures through Kokkos and KokkosKernels
- Accessible through unified Trilinos solver interface Stratimikos

Methodology

- **Parallel scalable multi-level Schwarz domain decomposition preconditioners**
- **Algebraic construction** based on the parallel distributed system matrix
- **Extension-based coarse spaces**

Team (active)

- Filipe Cumaru (TU Delft)
- Alexander Heinlein (TU Delft)
- Kyrill Ho (UCologne)
- Sebastian Kinnewig (LUH)
- Axel Klawonn (UCologne)
- Jascha Knepper (UCologne)
- Stephan Köhler (TUBAF)
- Friederike Röver (TUBAF)
- Siva Rajamanickam (SNL)
- Oliver Rheinbach (TUBAF)
- Lea Saßmannshausen (UCologne)
- Ichitaro Yamazaki (SNL)

Monolithic (R)GDSW Preconditioners for CFD Simulations

Consider the discrete saddle point problem

$$\mathcal{A}x = \begin{bmatrix} K & B^\top \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} = b.$$

Monolithic GDSW preconditioner

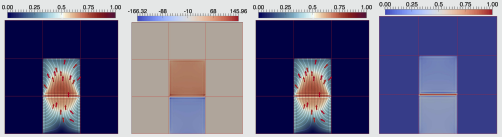
We construct a **monolithic GDSW preconditioner**

$$M^{-1}_{\text{GDSW}} = \phi \mathcal{A}_0^{-1} \phi^\top + \sum_{i=1}^{N} \mathcal{R}_i^\top \overline{\mathcal{P}}_i \mathcal{A}_i^{-1} \mathcal{R}_i,$$

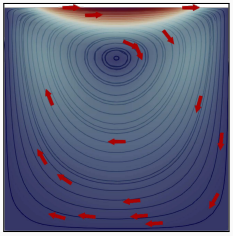
with block matrices $\mathcal{A}_0 = \phi^\top \mathcal{A} \phi$, $\mathcal{A}_i = \mathcal{R}_i \mathcal{A} \mathcal{R}_i^\top$, local pressure projections $\overline{\mathcal{P}}_i$, and

$$\mathcal{R}_i = \begin{bmatrix} \mathcal{R}_{u,i} & 0 \\ 0 & \mathcal{R}_{p,i} \end{bmatrix} \quad \text{and} \quad \phi = \begin{bmatrix} \Phi_{u,u_0} & \Phi_{u,p_0} \\ \Phi_{p,u_0} & \Phi_{p,p_0} \end{bmatrix}.$$

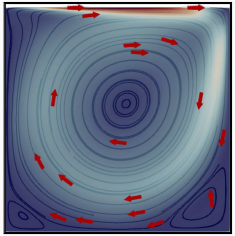
Using $\mathcal{A}$ to compute extensions: $\phi_I = -\mathcal{A}_{II}^{-1} \mathcal{A}_{I\Gamma} \phi_\Gamma$; cf. **Heinlein, Hochmuth, Klawonn (2019, 2020)**.



Φ_{u,u_0} Φ_{p,u_0} Φ_{u,p_0} Φ_{p,p_0}



Stokes flow



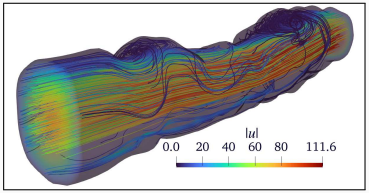
Navier–Stokes flow

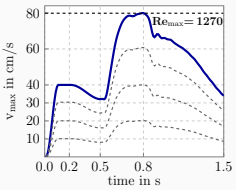
Related work:

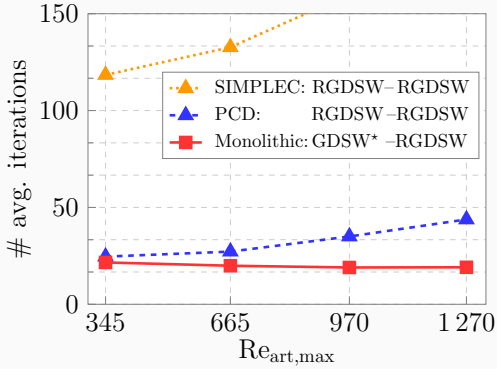
- Original work on monolithic Schwarz preconditioners: **Klawonn and Pavarino (1998, 2000)**
- Other publications on monolithic Schwarz preconditioners: e.g., **Hwang and Cai (2006)**, **Barker and Cai (2010)**, **Wu and Cai (2014)**, and the presentation **Dohrmann (2010)** at the *Workshop on Adaptive Finite Elements and Domain Decomposition Methods* in Milan.

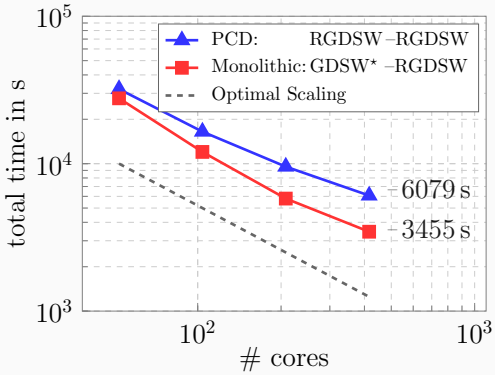
Results for Blood Flow Simulations

- **3D unsteady flow simulation** within the **geometry of a realistic artery** (from **Balzani et al. (2012)**) and kinematic viscosity $\nu = 0.03\,\mathrm{cm}^2/\mathrm{s}$
- **Parabolic inflow profile** at inlet
- **Time discretization**: BDF-2; **space discretization**: P2-P1 elements









Cf. **Heinlein, Klawonn, Knepper, Saßmannshausen (arXiv 2025)**

Investigating the impact of arterial wall models

Parallel Fluid-Structure Interaction Simulation Framework

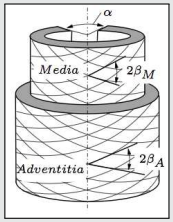
Arterial wall models

We consider **anisotropic hyperelastic material models** of the form

$$\Psi_* = \underbrace{\Psi^{\text{iso}} + \Psi^{\text{pen}}}_{\text{isotropic (neo-Hookean type)}} + \underbrace{\sum_{a=1}^{2} \Psi_a^{\text{ti}}}_{\text{transversely isotropic}}$$

to account for the fibers in the arterial wall and to obtain **realistic material behavior and stresses**.

We use the material models Ψ_A, Ψ_B, Ψ_C, Ψ_D, Ψ_E, and the material parameters (**fitted to experiments**) from **Brands, Klawonn, Rheinbach, Schröder (2008)**.



Composition of the arterial wall

Monolithic solver

- We solve the **fully coupled FSI problem** using an **inexact Newton method**.
- **Monolithic block preconditioner** with a **parallel domain decomposition preconditioner** for the blocks, e.g., `FROSch`.

Software framework

`Trilinos` (C++)
`LifeV` (C++)
`FEAP` (Fortran)

} **Lightweight coupling library**; see **Heinlein (2016)**.

Comparing Different Material Models

We compare the material models considered in **Brands, Klawonn, Rheinbach, Schröder (2008)**.

- Model ψ_A (**Balzani, Neff, Schröder, Holzapfel (2006)**):

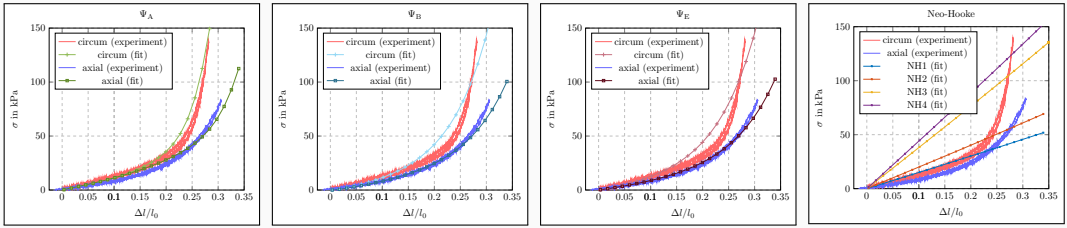
$$\psi_A = c_1\left(\frac{I_1}{I_3^{1/3}} - 3\right) + \varepsilon_1\left(I_3^{\varepsilon_2} + \frac{1}{I_3^{\varepsilon_2}} - 2\right) + \sum_{a=1}^{2} \alpha_1 \left\langle I_1 J_4^{(a)} - J_5^{(a)} - 2 \right\rangle^{\alpha_2},$$

- Model ψ_B (**Holzapfel, Gasser, Ogden (2000)**):

$$\psi_B = c_1\left(\frac{I_1}{I_3^{1/3}} - 3\right) + \varepsilon_1\left(I_3^{\varepsilon_2} + \frac{1}{I_3^{\varepsilon_2}} - 2\right)^{\alpha_5} + \sum_{a=1}^{2} \frac{k_1}{2k_2}\left\{\exp\left(k_2\left\langle \frac{J_4^{(a)}}{I_3^{1/3}} - 1\right\rangle^2\right) - 1\right\}$$

- Model ψ_E (anisotropic parts: **Holzapfel, Gasser, Ogden (2004)**):

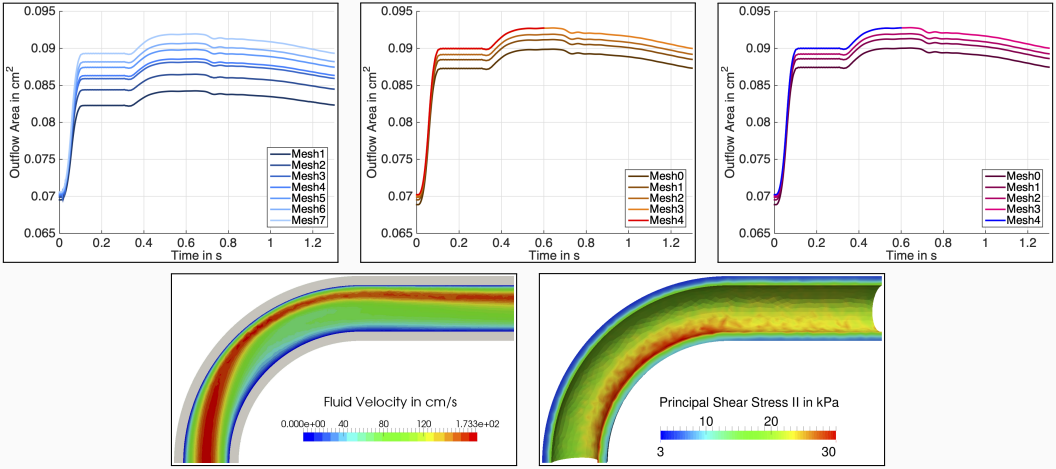
$$\psi_E = c_1(I_1 - \ln(I_3)) + \varepsilon_1\left(I_3^{\varepsilon_2} + \frac{1}{I_3^{\varepsilon_2}} - 2\right) + \sum_{a=1}^{2} \frac{k_1}{2k_2}\left\{\exp\left(k_2\left\langle J_4^{(a)} - 1\right\rangle^2\right) - 1\right\}$$



Cauchy stress σ [kPa] vs. strain λ (comp) compared to the experimental data (exp).

Results Ramp & Heart Beat – Anisotropic Hyperelastic Material Model (Ψ_A)

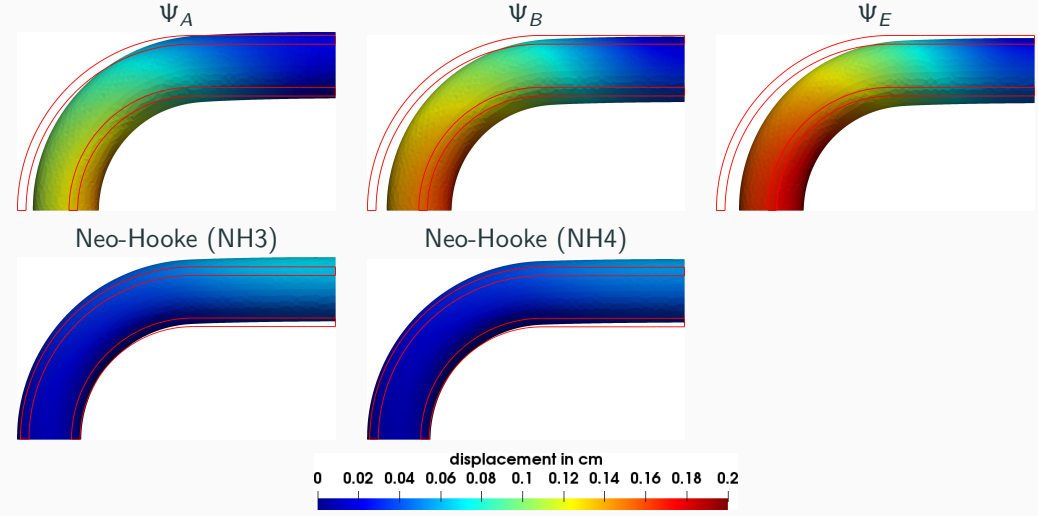
Balzani, Deparis, Fausten, Forti, Heinlein, Klawonn, Quarteroni, Rheinbach, Schröder (2016)



Top: Outflow cross sectional lumen area for P1 (left), P2 (middle), and $\bar{F} = P2 - P0 - P0$ (right) elements. $\Rightarrow$ **Mesh convergence for P2 and $\bar{F}$ elements**

Bottom: Fluid velocities and wall stresses (left), principal shear stresses in the wall (right).

Peak Flow (0.536 s) – Structural Displacement

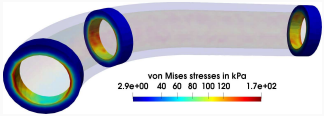


$\Rightarrow$ The **qualitative deformation behavior** of the neo-Hookean material model **differs significantly** from the other material models.

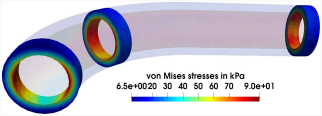
Cf. **Balzani, Heinlein, Klawonn, Rheinbach, Schröder (2023)**

Peak Flow (0.536 s) – von Mises Stresses

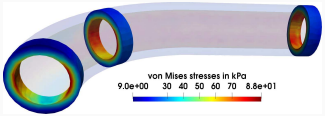
Ψ_A



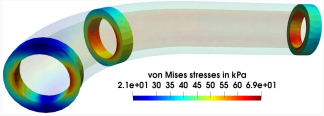
Ψ_B



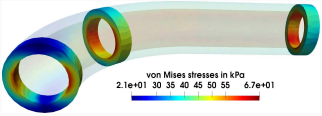
Ψ_E



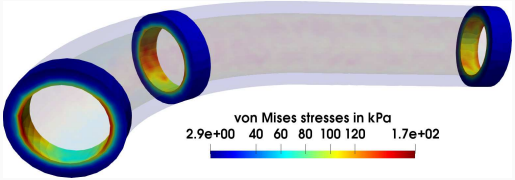
Neo-Hooke (NH3)



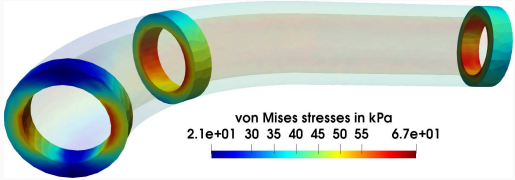
Neo-Hooke (NH4)



Ψ_A



Neo-Hooke (NH4)



$\Rightarrow$ Also the **qualitative stress distribution** (von Mises stresses) **differs significantly** for Neo-Hooke.

Cf. **Balzani, Heinlein, Klawonn, Rheinbach, Schröder (2023)**

Summary

Algorithmic Advances in Cardiovascular Simulations

- CNNs enable **fast surrogate models for CFD simulations**, combining data and physics for **significant prediction speedups**
- **High memory consumption** can be mitigated by employing a **domain decomposition-based U-Net architecture**
- **Monolithic Schwarz preconditioners** yield **scalable solvers** for **large numbers of processors** and **high Reynolds numbers**
- **Fiber directions** and **exponential stiffening** in the arterial wall model **strongly affect simulation outcomes**

Acknowledgements

- **Financial support**: DFG (KL2094/3-1, RH122/4-1), DFG SPP 2311 project number 465228106, Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE)
- **Computing resources**: CHEOPS (UoC), DelftBlue (TU Delft), Fritz (FAU), magnitUDE (UDE), OCuLUS (Uni Paderborn)

Thank you for your attention!