Project: Convolutional Neural Networks for time-dependent fluid flow

Project Description

Computational fluid dynamics (CFD) simulations are a numerical tool to model and analyze the behavior of fluid flow. However, accurate simulations are generally very costly because they require high grid resolutions. Therefore, there is high demand for reduced order models which reduce the computational costs of computing fluid flow fields. The project builds upon [1], where convolutional neural network models are used for steady computational fluid dynamics simulations. The goal of this project is to move from predicting stationary Navier-Stokes flow (1) to predicting the flow field of time dependent Navier-Stokes equations (2) . Besides this, geometry and material parameters will be varied. If time allows an extra step towards 3D geometries could be made.

$$\begin{aligned} -\nu\Delta u + (u\cdot\nabla)u + \nabla\frac{p}{\rho_0} &= 0 \; in \; \Omega \\ \nabla\cdot u &= 0 \; in \; \Omega \end{aligned}$$

Eq(1) Stationary Navier-Stokes equations

$$\begin{aligned} \frac{\partial u}{\partial \mathrm{t}} - \nu\Delta u + (u\cdot\nabla)u + \nabla\frac{p(t)}{\rho_0} &= 0 \; in \; \Omega \\ \nabla\cdot u &= 0 \; in \; \Omega \end{aligned}$$

Eq(2) Time-dependent Navier-Stokes equations

with certain boundary conditions on $\partial\Omega$, kinematic viscosity $\nu > 0$, velocity u, density ρ_0 and pressure p on a computational domain Ω.

This project aims to support the high level goal of making blood flow predictions based on MRI images.

Tasks

- Install and familiarize with the software packages:
  - The open-source CFD software OpenFOAM.
  - The Python machine learning libraries TensorFlow 2.0 and Keras.
- Implement and train a CNN for a simple data set
- Set up a software pipeline based on OpenFOAM to automatically generate (stationary) fluid flow data depending on certain parameters, such as viscosity, inflow velocity, etc. and varying geometries
- Based on the previous tasks, train a CNN for predicting stationary flow fields based on certain input parameters.
- Based on the previous tasks, use a CNN for time-stepping in reduced order (for example blood flow during one heartbeat)
- Optimization of the model and comparison against the reference data

Contact

Alexander Heinlein, a.heinlein@tudelft.nl.

References

[1] M. Eichinger, A. Heinlein, and A. Klawonn. Surrogate convolutional neural network models for steady computational fluid dynamics simulations. Technical report. Submitted December 2020. Preprint https://kups.ub.uni-koeln.de/29760/.