

Efficient Poisson solver for multi-GPU **WaterLily.jl** simulations

M.Sc. graduation project. Collaboration between EEMCS & ME

Graphical processing units (GPUs) have enabled massive scaling of computational fluid dynamics simulations due to their enormous computational power and relatively low cost. Modern programming languages, such as Julia, fully leverage this new era of computing and enable users to write kernel-specialized code at a high level of abstraction.



Figure 1: Unsteady flow around a leading-edge inflated kite. Simulation run on a NVIDIA A6000 GPU using 320M grid cells.

These opportunities have triggered the recent development of specialized computational fluid dynamics (CFD) software. **WaterLily.jl**, a Julia-based open-source CFD solver developed at TU Delft, can run using many GPUs in parallel and perform scale-resolving simulations; see Figure 1.

While recent developments in WaterLily have enabled such distributed-memory parallelism, the solver's efficiency is currently limited by the scalability of the multigrid solver used to solve the pressure–Poisson equation at each time step. Multigrid methods exhibit optimal computational complexity for Poisson-type problems, but most of their algorithmic components were developed in an era of computing based on central processing units (CPUs) and therefore may not perform as efficiently on GPUs. Since many physical simulations ultimately require Poisson-like solves, the development of efficient multi-GPU multigrid methods is crucial for improving performance across a wide range of scientific and engineering applications.

Task

In this project, you will investigate an efficient distributed-memory parallel multigrid Poisson solver for CFD simulations in WaterLily.jl, targeting multi-GPU systems. This includes not only implementation aspects, but also the development and adaptation of multigrid algorithms to better exploit GPU architectures. The new linear iterative solver will be compared with reference solvers, such as the currently used multigrid solver, in terms of efficiency and parallel scalability.

What's in for you

- Work on an exciting project involving numerical linear algebra and high-performance computing, impacting the solution of problems in computational science and engineering.
- Contribute to the development of the popular open-source CFD solver WaterLily.jl
- Learn or improve your skills in Julia through a general scientific computing problem.

Contact

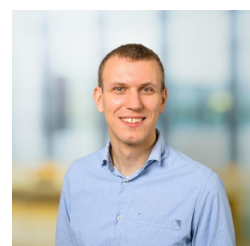
For more information, get in touch with us!



Bernat Font
b.font@tudelft.nl



Marin Lauber
m.lauber@tudelft.nl



Alexander Heinlein
a.heinlein@tudelft.nl