ENGINEERING MATHEMATICS AND COMPUTING LAB (EMCL) RESEARCH GROUP

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Master Thesis:

Geometric Multigrid Method on Heterogeneous Hardware

Target Subject:

- Applied Computer Science
- Mathematics
- Scientific Computing

Prerequisites:

- Experience in C++ programming
- Experience in parallel programming, e.g. MPI, OpenMP, Cuda (recommended)
- Basic knowledge in linear algebra and numerics (e.g. lecture “Introduction to numerics”),
- Basic knowledge in numerical methods for partial differential equations (recommended, e.g. lecture “Finite Elements”)

Contact:

- Philipp Gerstner, philipp.gerstner@uni-heidelberg.de
- Aksel Alpay, aksel.alpay@uni-heidelberg.de
- Vincent Heuveline, vincent.heuveline@uni-heidelberg.de

Nowadays, the geometric multigrid method is considered among the most efficient ways to solve the linear systems of equations that arise when discretizing partial differential equations (PDE). In this method, the discretized PDE is solved on a hierarchy of grids and involves four main tasks: matrix-vector multiplication, interpolation of discrete solutions between different grids, low-accuracy iterative linear solvers (smoothers) and high-accuracy (direct) linear solvers. Each of these tasks can be executed in parallel on multiple computing cores, however, to a varying extend.

On the other hand, modern high-performance computing systems provide the users a variety of different hardware architectures to run their applications on. The most prominent examples of such architectures are central processing units (CPU) and graphics processing units (GPU). The SYCL standard was established to address these different units by means of a single C++ interface. Using SYCL, it is possible to run individual computing tasks on that piece of hardware, that is best suited.

The goal of this master thesis is to implement a geometric multigrid method by using SYCL and to perform numerical experiments to determine relationships of the form “algorithmic sub-task → architecture” that minimize overall computation time. Moreover, different types of smoothers should be investigated w.r.t. computational efficiency.

The implementation should be embedded into the open-source finite element package HiFlow³, which allows to the test the multigrid implementation for various types of physical problems, e.g. diffusion processes, elasto-mechanics, or fluid dynamics.