Technische Universität Braunschweig
Institute of Applied Mechanics
Pockelsstraße 3
38106 Braunschweig

Prof. Dr.-Ing. Ralf Jänicke

www.tu-braunschweig.de/iam

Contact
Dr. Fredrik Ekre
f.ekre@tu-braunschweig.de

Algebraic multigrid preconditioner for distributed linear systems

Numerical methods for solving partial differential equations (PDE), such as e.g. the finite element method (FEM), usually results in a linear system of equations, $Ax = b$, where A is a matrix, and where b and x are vectors representing the right hand side and the solution, respectively. There are (hardware) limitations for how large such systems can be solved on a single computer, so for large problems it is necessary to split up the problem to multiple connected computers in a *computer cluster*. However, this is not trivial and, for example, the procedure for solving the linear system becomes more difficult when A is distributed over different machines.

There are two categories of solvers for the linear system, *direct* and *iterative*, where the most appropriate usually depends on the PDE. Here we will focus on iterative solvers, such as e.g. *conjugate gradients*. In order to reduce the number of iterations it is common to apply a *preconditioner* on the system. One such method is *algebraic multigrid* (AMG) which recursively identifies coarser (i.e. smaller) matrices A_1, A_2, ..., that still contain the most "important" information of the original matrix. This process is also more difficult in a distributed setting, and in particular it is not trivial how to deal with matrix entries that are shared between two or more computers. The AMG method for distributed systems is discussed in Yang [1].

The goal of this project is to implement AMG preconditioning for distributed linear systems in the Julia programming language[a]. The implementation will be based on existing software[b] that facilitates distributed matrices and vectors.

References

[1] Yang, U. M. *Parallel Algebraic Multigrid Methods - High Performance Preconditioners.* (2004).

[a] https://julialang.org/
[b] https://github.com/fverdugo/PartitionedArrays.jl