Parallel Algebraic Multigrid Solvers for PDEs from Biomedical Applications

Gundolf Haase

Institute for Mathematics and Scientific Computing, University of Graz, Austria

Abstract

Cardiovascular simulations require the solution of coupled PDE/ODE equations with several internal couplings of the PDEs depending on the underlying model and the available compute capabilities.

The MPI/OpenMP and GPU (CUDA) parallelization of the ODE solver and the elliptic/parabolic potential problem solver has been successfully performed in the past. We use a cg-iteration with an algebraic multigrid preconditioner (AMG) for the elliptic problem and its general parallelization will be presented in the talk.

This parallelization concept has been revisited with respect to load balancing of the subdomain interfaces of the decomposed domain and resulted in a much better strong parallel efficiency, especially on clusters of GPUs.

The matrices for the potential problems remain unchanged during the whole calculation, i.e., the matrices are computed and assembled on the CPU and transferred only once to the GPU. The same holds for the AMG setup.

In order to provide a GPU solver for elasticity we extended the AMG for coupled problems. Here, several versions have been investigated and the AMG with coupled degrees of freedom in each node together with a graph coarsening proved the best robustness and the best timings. The relation between costs for setup and solver changes completely in case of non-linear elasticity.

The original CPU code spent 50% of its time in matrix calculation and accumulation, i.e., transferring only the solver onto the GPU would result in a speedup less than 2 compared to one CPU core. Therefore, we had a closer look at the matrix calculation and assembling. Currently, the calculation of the local stiffness matrices is accelerated by 1000 on a GTX 680.

The assumption that the matrix graph won't change during the non-linear calculation supports the GPU acceleration of this step (and also the CPU acceleration).

This assembling of the local contributions into the global stiffness matrix on the GPU is subject to ongoing work. Additionally, the deformed geometry requires a mesh smoothing which will be provided on basis of radial basis functions (RBF).

A similar assumption will be used for the AMG setup on the CPU/GPU resulting in several setup entry points with very different computational costs and data transfers between CPU and GPU. Besides a setup phase, the full non-linear iteration algorithm will run completely on the GPU.

Taking also into account the dramatically reduced data transfer between host and device we expect an acceleration of the non-linear iteration by a factor of 30 with respect to one CPU core.