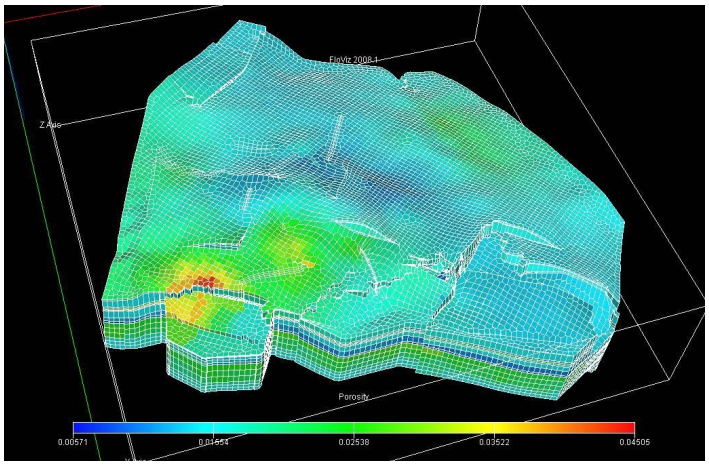

In reservoir simulation we predict the recovery of gas and oil by solving for the fluid flow in porous media. Our reservoir simulator is designed from the ground up to run in parallel to speed up the computations and to reduce the overall simulation time. In general we distinguish between two properties when assessing the performance of an algorithm executed in parallel: weak and strong scalability. Weak scalability is the "speed up" which results from increasing the number of processors and keeping the same problem size per processor. In the ideal case we want to solve twice the problem size in the same time given twice the number of processors. Strong scalability is defined as the speed up which results from increasing the number of processors and freezing the original problem size. In the ideal case we solve the original problem in half the time given twice the number of processors.

In reality these ideal scenarios are hard to achieve because of latency in communication, i.e. there exists no communication between processors with infinite speed. Moreover, we pay a penalty when the ratio between communication and the actual amount of work per processor becomes unfavorable. This means that a processor is without work while waiting for data from neighboring processors.

The time-dependent, non-linear equations which underlie reservoir simulation are discretized by the Finite Volume method and solved by the Newton-Raphson method and implicit time integration. In every step of the Newton-Raphson method we evaluate the derivatives of the non-linear reservoir simulation equations and solve the corresponding linear system. Typically for a simulation of cells we solve for unknowns, with unknowns per cell. These are pressure, water (and gas) saturations and molefractions.

The matrices in the resulting linear systems are non-symmetric. We solve these systems with FGMRES preconditioned by the constrained pressure residual (CPR). To tackle the two different numerical properties of the discrete operator, elliptic and hyperbolic, we have two stages in CPR: 1st stage solve the pressure equations, 2nd stage solve the complete system. We solve the 1st stage by AMG and the 2nd stage by block-ILU(k), where k is 0 or 1.

The AMG solver is the bottleneck of this algorithm. It takes considerable amount of time, mostly in the setup. AMG is the state-of-the-art and (near-) optimal for solving elliptic equations. Moreover, AMG is well known for its excellent weak scalability property, it scales well for increasingly bigger problem sizes. However, the strong scalability of AMG is poor. The poor strong scalability of AMG is a critical business issue as clients expect to run cases (considerably) faster when running cases of fixed size on a cluster.

The aim of this workshop is to explore if domain decomposition methods can improve the strong scalability of the pressure solver.