FEM simulations of incompressible flow using automatic differentiation in the PDE framework Peano

Tobias Neckel[1] Hans-Joachim Bungartz[2]

Modern CSE applications such as multi-physics or fluid-structure interaction simulations need more and more efficient methods. Partitioned approaches have proven to be suitable for such simulations but heavily depend on the efficiency of the individual solvers. Therefore, the C++ PDE framework *Peano* [1] has been developed. *Peano*'s flow solver supports incompressible fluid flow on regularly or adaptively refined Cartesian grids using low-order finite elements in combination with space-filling curves and a very cache-efficient stack data concept. The steady-state or time-implicit discretisation of the Navier-Stokes equations result in a non-linear system of equations $B(\mathbf{u}, p) = 0$ with velocities $\mathbf{u}$ and pressure p. In *Peano*, this nonlinear system is solved with Newton's method within the PETSc toolkit [2] by explicitly assembling the Jacobian.

We use the well-known benchmark flow around a cylinder to compare three different approaches to compute the Jacobian: A tuned finite difference approximation, the computation of the exact Jacobian via the automatic differentiation (AD) tools ADOL-C [3] and cppAD [4], and finally the analytical differentiation of contributions to the nonlinear function B. All three versions are implemented using a strictly cell-wise operator evaluation. The analytical approach results in the lowest assembly times needing 0.4–2.6% of the total non-linear iteration durations (decreasing with increasing mesh resolution). The implementations via finite differences, ADOL-C and ccpAD show an overhead of a factor of about 5.5, 2.7, and 18, respectively. While different implementations using ADOL-C have been tested, including also tapeless variants, an efficient one-touch strategy for the tape has been used to obtain these runtimes with ADOL-C, recording data only in the very first call on the first cell and evaluating this record on all cells with the corresponding data.

Thus, we removed the bottleneck for steady-state or time-implicit non-linear solutions concerning the computation of the Jacobian. All three approaches may easily be extended to other ansatz functions or other applications within our *Peano* framework for regular or adaptive Cartesian grids. The AD variants possess the advantage that no changes in the code regarding the Jacobian are necessary for such extensions. The efficient one-touch strategy for ADOL-C is usable for any other (finite element) discretisation using a direct, cell-wise assembly.

References:

[1] M. Brenk, H.-J. Bungartz, M. Mehl, I.L. Muntean, T.Neckel, and T.Weinzierl: Numerical Simulation of Particle Transport in a Drift Ratchet. SIAM Journal of Scientific Computing 6(30), 2008, pp. 2777–2798

[2] `http://www.mcs.anl.gov/petsc/petsc-as/index.html`

[1] TU München, Institut für Informatik, Boltzmannstr. 3, 85748 Garching, Germany, `neckel@in.tum.de`

[2] TU München, Institut für Informatik, Boltzmannstr. 3, 85748 Garching, Germany, `bungartz@in.tum.de`

[3] http://www.math.tu-dresden.de/~adol-c/
[4] http://www.coin-or.org/CppAD/