High Performance FEM Simulation in CFD and CSM

Prof. Dr. Stefan Turek, Dipl.-Math. Sven Buijssen, Dipl.-Math. Hilmar Wobker

1. Motivation

Processor technology is still dramatically advancing and promises further enormous improvements in *processing data* for the next decade. In contrast, much lower advances in *moving data* are expected such that the efficiency of many numerical software tools for Partial Differential Equations (PDEs) is restricted by the cost for memory access. In last year's Research Report [7] we outlined the numerical concepts pursued at our chair to overcome the pinpointed deficiencies while achieving high numerical and parallel efficiency at the same time: adaptive Finite Element Method (FEM) approaches and generalised multigrid/domain decomposition solvers of ScaRC type, realised in the FEM package FEAST.

The aim of this article is to illustrate how problems from Computational Structural Mechanics (CSM) and Computational Fluid Dynamics (CFD) can be tackled in the FEAST framework. Since this basic library only provides facilities to solve scalar problems, the question is how to treat multi-field simulations. The main focus of the article is concentrated on the design of appropriate preconditioners for the resulting saddle point problems which have a major impact on the numerical efficiency of the underlying iterative algorithms.

2. Generalised Stokes Equation

The incompressible nonstationary Navier–Stokes equations describe the behaviour of a Newtonian fluid at constant temperature with constant kinematic viscosity enclosed in a volume with Dirichlet and/or Neumann boundary conditions. Neglecting in a first step the nonlinear convection term and applying a simple time-discretisation method with timestep k leads to the generalised Stokes equation:

$$\begin{aligned} \mathbf{u} - \nu k \Delta \mathbf{u} + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \qquad (1)$$

A similar equation arises in CSM: One possibility to address the problem of *nearly incompressible* elastic material is to introduce, beside the displacements u, a second variable $p := -\lambda \nabla \cdot \mathbf{u}$, which results in a mixed formulation. When a Newmark time discretisation scheme is applied, it comes to the following generalised equation

$$\begin{aligned} \mathbf{u} - 2\mu\tilde{k}\, \nabla \cdot \varepsilon(\mathbf{u}) + \nabla p &= \mathbf{f} \\ \nabla \cdot \mathbf{u} + \frac{1}{\lambda} p &= 0 \end{aligned} \qquad (2)$$

with $\tilde{k} := \beta k^2$ and β coming from the Newmark scheme. Due to the similarity between equation (1) and (2) we will concentrate only on the Stokes equation from now on. Most of the following applies to the elasticity case, as well, while differences will be emphasised.

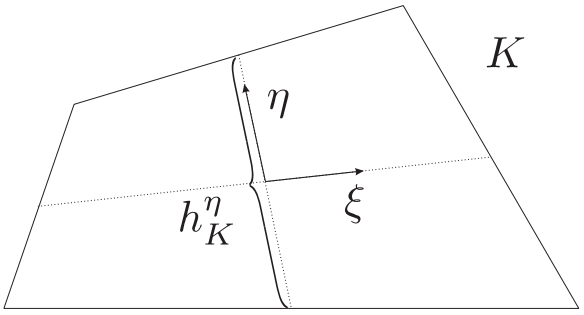

Figure 1. Local coordinate system for an element

At present, FEAST and its underlying SparseBandedBLAS library only feature discretisation by bilinear elements. Since a straight-forward discretisation with bilinear elements for both velocity and pressure (Q_1/Q_1) would violate the so-called Babuška-Brezzi condition [2] appropriate stabilisation is needed as described in [4,3,1]. In order not to lose the ability of dealing with irregular grids we extend the standard stabilisation technique by considering directional derivatives

$$c(p,\psi) = \sum_K (h_K^\xi)^2 \, (\xi\nabla p,\, \xi\nabla\psi)_K + \sum (h_K^\eta)^2 \, (\eta\nabla p,\, \eta\nabla\psi)_K$$

where h_K^ξ, h_K^η measure the extensions of each element K for the local coordinate system (ξ,η) (see fig. 1). After discretisation the problem is brought down to repeatedly solving linear systems of the following type:

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \qquad (3)$$

2.1 Solving Strategies for Saddle Point Problems

Disregarding the matrix C consisting of stabilisation terms (and the compressibility constraint in the elasticity case) equation (3) is a classic saddle point problem. For nonsingular matrices A the velocity u can be eliminated formally, yielding the scalar equation

$$(B^T A^{-1} B - C)\, p = B^T A^{-1} f - g \tag{4}$$

with the so called *pressure Schur complement* $S := B^T A^{-1} B - C$ and right hand side $b := B^T A^{-1} f - g$. To solve this equation the following *basic iteration* can be applied:

$$p_k = p_{k-1} + \tilde{S}^{-1} (b - S p_{k-1}) \tag{5}$$

In a simple version $\tilde{S}$ is a damped identity matrix, but to get reasonable convergence rates $\tilde{S}$ has to be chosen as an approximate of the Schur complement matrix S, i.e. as a *good preconditioner*. Furthermore the basic iteration (5) has to be integrated in a preconditioned Krylov-space method.

Within such a method matrix-vector-multiplications with S have to be performed. As S is only given implicitly this means three matrix-vector-multiplications and "inverting" the matrix A. The latter has to be done exactly, otherwise, instead of (4), the "wrong" system $(B^T \tilde{A}^{-1} B - C) p = B^T \tilde{A}^{-1} f - g$ would be treated. Since inverting A exactly is prohibitively expensive and (4) has to be solved repeatedly, alternatives have to be found.

The first approach is to embed the algorithm in an outer defect correction method acting on the whole system (3). The corresponding basic iteration looks like:

$$\begin{pmatrix} u^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} u^n \\ p^n \end{pmatrix} + N_S^{-1} \left[\begin{pmatrix} f \\ g \end{pmatrix} - \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} u^n \\ p^n \end{pmatrix} \right] \tag{6}$$

Thus the Schur complement method merely acts as a preconditioner (formally written as N_S^{-1}), which allows the approximate treatment of A^{-1}. The basic iteration (6) is again accelerated by using a Krylov-space method.

A second approach is to choose in the basic iteration (6) the block triangular matrix

$$N := \begin{pmatrix} A & 0 \\ B^T & -S \end{pmatrix} \tag{7}$$

as *block-preconditioner* for the whole system (3). For the preconditioned system matrix

$$K := N^{-1} \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

it can easily be shown, that the corresponding Krylov subspace $\mathrm{span}\{r, Kr, K^2 r, K^3 r, \ldots\}$ has dimension 2, i.e. the solution of the preconditioned system would require only two iterations of a Krylov-space method (see [5]). Of course, the application of N^{-1}, which involves the exact computation of A^{-1} and S^{-1}, is much too expensive, such that (7) is actually replaced by

$$\tilde{N} := \begin{pmatrix} \tilde{A} & 0 \\ B^T & -\tilde{S} \end{pmatrix}, \tag{8}$$

where $\tilde{A}$ and $\tilde{S}$ denote preconditioners for A and S, respectively. While the design of $\tilde{S}$ requires a closer look at the underlying equations, which will be done in the next section, the realisation of $\tilde{A}$ is straightforward: Considering examplarily the stationary Stokes equation in $2D$, we have

$$A = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix}$$

where L_1 and L_2 are the discretisations of scalar Laplace operators for the x- and the y-component, respectively. So, the preconditioner $\tilde{A}$ is simply realised, e.g. by independently doing one ScaRC iteration for each component. In the elasticity case, however, x- and y-direction are coupled, resulting in non-zero off-diagonal blocks in A. Consequently, we cannot simply do two independent ScaRC iterations as in Stokes case, but we have to resolve the coupling by embedding the ScaRC solves as preconditioner into another outer Krylow-space method applied to A.

Anyway, in both cases the treatment of a multi-dimensional system is brought down to the solution of scalar equations, which enables us to exploit the ScaRC solvers' strengths.

2.2 Preconditioning of the Schur Complement

In both approaches depicted above to solve the system (3) we face the problem that a preconditioner $\tilde{S}$ for the Schur complement $S = B^T A^{-1} B - C$ is needed (compare (5), (8)). Examining the generalised Stokes equation (1) we can deduce the structure of A, namely

$$A = M + \nu k L, \tag{9}$$

where M is the (lumped) mass matrix and L the Laplacian, both block-structured with zero off-diagonals. The "nature" of A clearly depends on the size of the timestep k: For very small timesteps the mass matrix dominates, while it has, in fact, no influence for very large timesteps and even vanishes for the stationary Stokes case. Our goal is to construct a preconditioner that efficiently covers the whole range of relevant timesteps.

To reach this goal we exploit the additive decomposition (9) of A by designing the preconditioning operator correspondingly. To this end we consider the distinct parts of the Schur complement S. The *reactive part*

$$B^T M_l^{-1} B$$

can be interpreted as a discretisation matrix steeming from a mixed formulation of the (continuous) Poisson problem. So, the preconditioning operator is chosen as L_p, the Laplacian matrix corresponding to the discrete

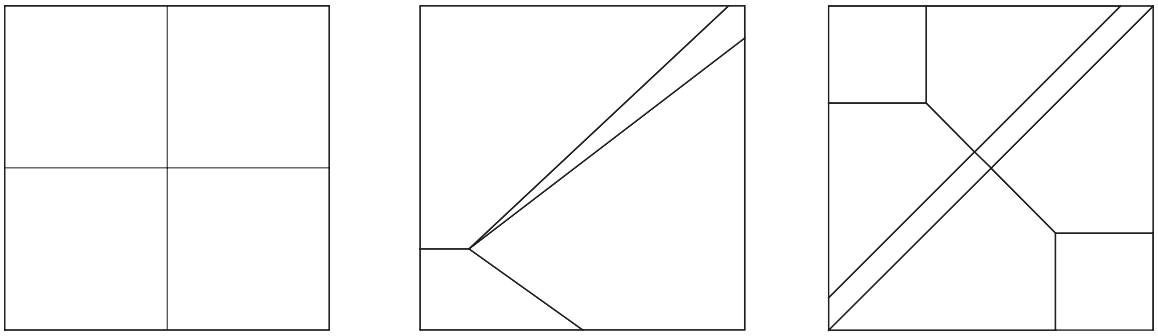

Figure 2. Prototypical coarse grids

pressure space. The continous operator associated with the *diffusive part*

$$B^T L^{-1} B$$

is spectrally equivalent to the identity [6], so $M_{l,p}$, the lumped pressure mass matrix, is an optimal preconditioner. This also holds for the elasticity case, where we have, instead of vkL, the matrix $2\mu\tilde{k}K$ with K being the discretisation of $\nabla \cdot \varepsilon(\mathrm{u})$. We now linearly combine the two parts corresponding to (9) and thus obtain the desired Schur complement preconditioner:

$$\tilde{S}^{-1} = L_p^{-1} + vkM_{l,p}^{-1} \tag{10}$$

This preconditioner seems not to cover the matrix C, which appears in the Schur complement. The entries of the stabilisation part of C are of magnitude $O(h^2)$, so it can usually be neglected. Difficulties arise, if the time step k is about the size of h^2 or smaller. Then, the influence of the stabilisation matrix – compared to that of the diffusive part – cannot be neglected anymore and C has to be implemented into the preconditioner, as well. In the elasticity case, the part of C coming from the compressibility constraint is simply a pressure mass matrix and thus can be covered by the diffusive part of the preconditioner (10).

To validate the described preconditioner we performed numerical tests on three prototypical grids of different complexity (see fig. 2). The first is an ideal orthogonal grid, the second contains a deformed near-triangle element while the last has stretched elements of aspect ratio 10, which are not parallel to the coordinate axes.

The tests cover the whole range of practically relevant time steps ($k \in [10^{-6}, 10^{6}]$). As $\tilde{S}^{-1}$ is constructed for the distinct parts of the Schur complement S its efficiency for the intermediate interval of $k \in [10^{-3}, 10]$ is of particular interest – most notably because in a fully non-stationary simulation the typical time step will be contained in this interval. Fig. 3 shows for the three different grids the number of arithmetic operations needed per degree of freedom (d.o.f.).

Apparently, the preconditioner $\tilde{S}^{-1}$, embedded in a Krylow space method, is capable to handle the

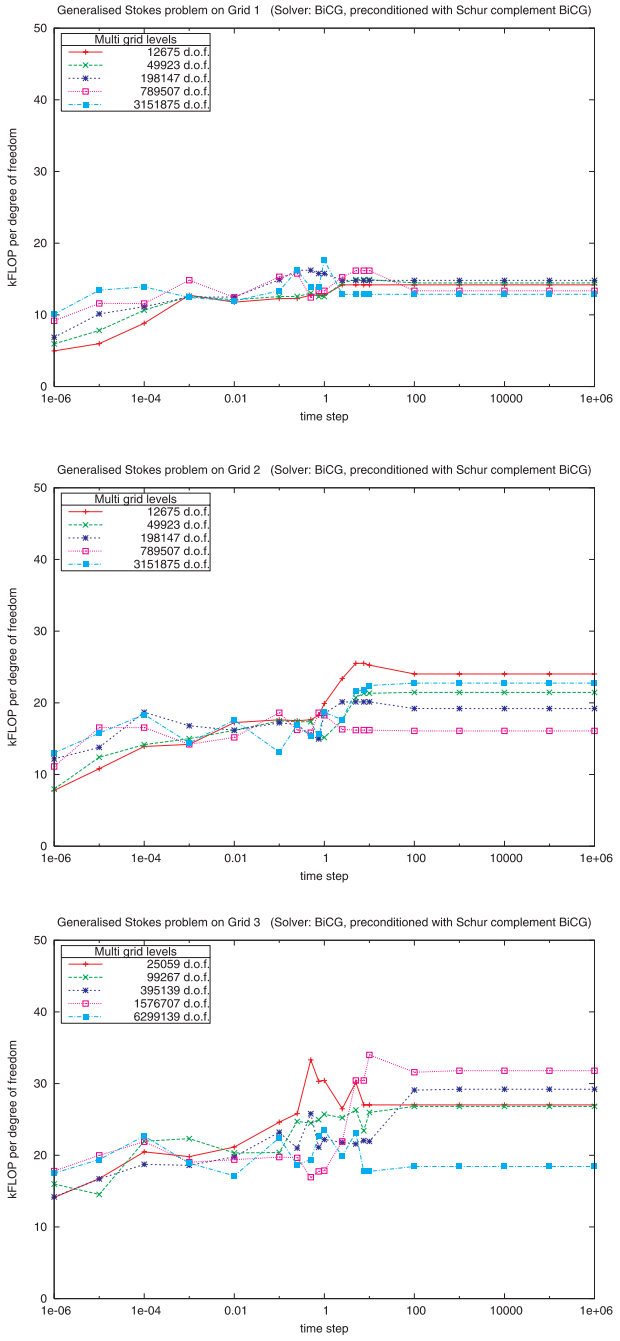

Figure 3. Operations per degree of freedom for $k \in [10^{-6}, 10^{6}]$ for the prototypical grids

d.o.f.	Grid 1	Grid 2	d.o.f.	Grid 3	d.o.f.	car-like shape
12,675	259	166	25,059	144	28,518	60
49,923	293	228	99,267	205	112,326	99
198,147	273	257	395,139	244	445,830	154
789,507	261	264	1,576,707	258	1,776,390	216
3,151,875	251	262	6,299,139	267	7,091,718	228

Table 1. MFLOP/s rates for the overall solver scheme (values time-averaged as they are almost constant for all time steps). Sequential computation on an AMD Opteron 250 with 2,4 GHz.

complete range of time steps and does not deteriorate significantly on irregular grids: the amount of arithmetic operations per d.o.f. is, in fact, only doubled. Beyond that, Table 1 shows that the solver scheme proposed succeeds in maintaining the high performance of FEAST. In contrast to the usual behaviour of FEM software the MFLOP/s rates do *not* cripple with increasing problem size, instead they rise up to and then remain at roughly 270 MFLOP/s.

We also did a more complex simulation of a car-like shape in a wind tunnel. The coarse grid consists of 36 macros including stretched and near-triangle elements (see fig. 4). Also on this irregular grid the algorithm achieves rates of more than 200 MFLOP/s (see Table 1). The resulting flow field can be seen in fig. 5.

The numerical examinations show that the described preconditioner meets the requirements we established above. Thus, we have a powerful solver mechanism for time-dependent saddle point problems at hand. It is robust with respect to grid irregularities and time step sizes and it shows high numerical efficiency while achieving high MFLOP/s rates at the same time.

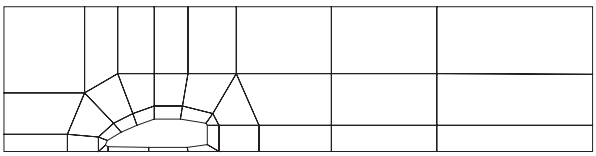

Figure 4. Car in a wind tunnel: macro grid

Figure 5. Car in a wind tunnel: simulated pressure (Stokes flow)

References

[1] R. Becker. An Adaptive Finite Element Method for the Incompressible Navier–Stokes Equations on Time-Dependent Domains. PhD thesis, Universität Heidelberg, 1995.

[2] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. RAIRO Anal. Numer., 8(R-2):129–151, 1974.

[3] T. J. R. Hughes and L. P. Franca. A new finite element formulation for computational fluid dynamics: VII. The Stokes-problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. Comp. Meth. Appl. Mech. Eng., 65:85–96, 1987.

[4] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition. A stable Petrov-Galerkin formulation of the Stokes problem accomodating equal-order interpolations. Computer Methods in Applied Mechanics and Engineering, 59(1):85–99, 1986.

[5] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. SIAM J. Sci. Comput., 21(6):1969–1972, 2000.

[6] S. Turek. Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach. Springer, 1999.

[7] S. Turek, S. H. M. Buijssen, M. Grajewski, and H. Wobker. High performance FEM simulation. Research Report, pages 52–55, Sept. 2004. NRW Graduate School of Production Engineering and Logistics.