Coupled Algebraic Multigrid Methods for the Oseen Problem

Markus Wabro *
Institute of Computational Mathematics
J.K. University Linz, Austria

February 25, 2003

Abstract

We provide a concept combining techniques known from geometric multigrid methods for saddle point problems (such as smoothing iterations of Braess- or Vanka-type) and from algebraic multigrid (AMG) methods for scalar problems (such as the construction of coarse levels) to a coupled algebraic multigrid solver. 'Coupled' here is meant in contrast to methods, where pressure and velocity equations are iteratively decoupled (pressure correction methods) and standard AMG is used for the solution of the resulting scalar problems.

To prove the efficiency of our solver experimentally, it is applied to finite element discretizations of "real life" industrial problems.

1 Introduction

An important problem in computational fluid dynamics is the numerical solution of the Oseen problem, arising from some kind of fixed point iteration for the nonlinear incompressible Navier-Stokes equations.

If this problem is discretized (in our case with finite elements), an (indefinite) saddle point system is obtained. Classical iterative methods used for the solution of this system are variants of SIMPLE schemes (c.f. Patankar [10]) or Uzawa algorithms, having in common an iterative decoupling to pressure and velocity equations, which then can be solved with methods known for the solution of positive definite systems.

As is the case for scalar equations, the efficiency of these classical methods can be outstripped by geometric multigrid (GMG) methods. First milestones in the application of GMG to saddle point systems were set by Verfürth [23] and Wittum [25]. Further important work in this direction was done by Braess and Sarazin, who use the classical Uzawa

*Supported by the Austrian Science Foundation "Fonds zur Förderung der wissenschaftlichen Forschung (FWF)"

solvers as smoothers [2] (which was generalized by Zulehner [26]). A different possibility of smoothing, introduced by Vanka [21], consists of the solution of local problems. Schöberl and Zulehner [17] provided a theoretical basis for this approach.

When dealing with real life applications, where complex three dimensional geometries are used, even the coarsest sensible discretizations lead to a need of memory and CPU time which prevents further refinement or direct solution of the problem (at least with respect to todays generation of computer hardware). Thus algebraic multigrid (AMG) methods, where the fine mesh is used as initial level and coarser levels are generated using (almost) only information from the algebraic system, were developed. These methods have since the pioneering work of Ruge and Stüben [15] and Brandt et al. [3] been applied to a wide class of linear systems arising (mostly) from scalar partial differential equations. An overview of the technique itself and its applications can be found in Stüben [19].

If we want to apply AMG methods to saddle point problems, there are generally two possibilities. The first one is to use SIMPLE or Uzawa methods as outer iteration process and to solve the inner (scalar, positive definite) systems with AMG. This approach is described e.g. by Griebel et al. [7] or Stüben [18].

The focus of this paper lies on the second possibility, on the coupled approach, described above for GMG methods. This direction is followed for example by Webster [24] or Raw [13]. We will put the emphasis on the correct 'translation' of methods known from geometric multigrid to algebraic multigrid to obtain a stable solver for "real life" three dimensional problems. This paper cannot be regarded as a rigorous theoretical basis but shows the way towards the implementation of an effective and working method.

The article is organized as follows. First we give a statement of the problem and its discretization and a general introduction to AMG methods. The main part of this work is section three, where the components of an AMG method for mixed problems are constructed and the major problems (especially the stability of the coarse level systems) are pinpointed. In section four we give a brief overview of the handling of the nonlinear Navier-Stokes problem, and in section five we present numerical results for two- and three-dimensional test cases.

2 Problem

2.1 The Oseen Problem

In a domain Ω, $\Omega \subset \mathbb{R}^2$ or $\Omega \subset \mathbb{R}^3$, we consider the system of instationary incompressible Navier-Stokes-equations

$$\delta_m \frac{\partial}{\partial t}\mathbf{u} - \nu\Delta\mathbf{u} + \delta_c(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \tag{1}$$

$$\operatorname{div}\mathbf{u} = 0, \tag{2}$$

plus boundary conditions, where $\mathbf{u}$ is the vector of velocity, p pressure, t time, $\mathbf{f}$ an external force, ν viscosity and δ_m, δ_c are auxiliary parameters usually equal 1. For $\delta_m = 0$ we get the stationary Navier-Stokes-equations, setting $\delta_c = 0$ (dropping the convective term) leads to the Stokes-equations.

If we linearize the system (for $\delta_c = 1$) by fixed point iteration we get

$$\delta_m \frac{\partial}{\partial t}\mathbf{u} - \nu\Delta\mathbf{u} + (\mathbf{w}\cdot\nabla)\mathbf{u} + \nabla p = \mathbf{f}, \tag{3}$$

$$\operatorname{div}\mathbf{u} = 0, \tag{4}$$

where $\mathbf{w}$ is the old approximation of the velocity. This system is called the Oseen problem.

2.1.1 Discretization

We discretize the problem with a stable (or stabilized) mixed finite element (FE) method and get the Galerkin FE discretization (with appropriate spaces $\mathbf{U}_h$ and Q_h, which are defined later): We search a pair $(\mathbf{u}_h, p_h) \in \mathbf{U}_h \times Q_h$ such that

$$\bar{a}(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = \langle F, \mathbf{v}_h\rangle \quad \forall \mathbf{v}_h \in \mathbf{U}_h, \tag{5}$$

$$b(\mathbf{u}_h, q_h) - c(p_h, q_h) = \langle G, q_h\rangle \quad \forall q_h \in Q_h, \tag{6}$$

where

$$\bar{a}(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) = \delta_m \frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{u}_h, \mathbf{v}_h)_0 + a_L(\mathbf{u}_h, \mathbf{v}_h) + \delta_c a_C(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) \tag{7}$$

and

$$a_L(\mathbf{u}_h, \mathbf{v}_h) = (\nabla\mathbf{u}_h, \nabla\mathbf{v}_h), \tag{8}$$

$$a_C(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) = ((\mathbf{w}_h\cdot\nabla)\mathbf{u}_h, \mathbf{v}_h), \tag{9}$$

$$b(\mathbf{u}_h, q_h) = -(\operatorname{div}\mathbf{u}_h, q_h), \tag{10}$$

Here $(.,.)_0$ is the L_2 scalar product, $\langle F, .\rangle = (\mathbf{f}, .)_0$, and $c(.,.)$ and G may be induced by some stabilizing method as below.

Possible stability problems due to the convective term can be solved e.g. by using the streamline upwinding Petrov-Galerkin (SUPG) scheme, where we replace the trilinear form $a_C(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h)$ in the equations above by $\bar{a}_C(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h)$, with

$$\bar{a}_C(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) = a_C(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) + a_S(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h), \tag{11}$$

$$a_S(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) = \beta_h(\mathbf{w}_h\nabla\mathbf{u}_h, \mathbf{w}_h\nabla\mathbf{v}_h), \tag{12}$$

where β_h is a parameter which should be of magnitude $\mathcal{O}(h)$ (for details see Pironneau [11]).

In the non-stationary case we use the method of lines combined with techniques known from the theory for ordinary-differential-equations for the treatment of the time derivative (e.g. one-step-θ, fractional-step-θ,..., cf. Rannacher [12]).

The matrix form of the system we want to solve for the coefficient vectors $\underline{\mathbf{u}}_h$ of $\mathbf{u}_h$ and $\underline{p}_h$ of p_h reads as

$$(\delta_m c_1 M + A_L + \delta_c \left(A_C(\underline{\mathbf{w}}_h) + A_S(\underline{\mathbf{w}}_h)\right))\underline{\mathbf{u}}_h + B^T \underline{p}_h = \underline{\mathbf{f}}_h, \tag{13}$$

$$B\underline{\mathbf{u}}_h - C\underline{p}_h = \underline{g}_h, \tag{14}$$

with given constant c_1 (depending on the size of the time-step and the discretization of time), convection speed $\underline{\mathbf{w}}_h$ and right hand sides $\underline{\mathbf{f}}_h$ and $\underline{g}_h$. We will often use

$$A := A(\underline{\mathbf{w}}_h) := c_1 \delta_m M + A_L + \delta_c \left(A_C(\underline{\mathbf{w}}_h) + A_S(\underline{\mathbf{w}}_h)\right).$$

In the remaining of this article we will drop the h-subscript and the underscores if it is obvious from the context.

What has not been fixed, up to now, is the concrete choice of the FE-pairing for velocity and pressure. We use two methods built upon linear elements.

- The first one is the (modified) Taylor-Hood finite element P1isoP2-P1 with linear shape-functions on a triangular/tetrahedral grid for the pressure and linear shape-functions for the velocity components on an uniformly refined mesh (where each triangle is divided into four sub-triangles, each tetrahedron into eight sub-tetrahedra), cf. Pironneau [11], Brezzi, Fortin [4].

- The other is the stabilized P1-P1 element, with linear shape-functions for both pressure and velocity. As this element is not LBB-stable the stabilizing (lower right) term
$$-c(p,q) = -\alpha_S \sum_{K\in\mathcal{C}_h} h_K^2 (\nabla p, \nabla q)_{0,K}, \tag{15}$$
and a right hand side for the continuity equation
$$\langle G, q\rangle = -\alpha_S \sum_{K\in\mathcal{C}_h} h_K^2 (\mathbf{f}, \nabla q)_{0,K}, \tag{16}$$
with some constant $\alpha_S > 0$ and $\mathcal{C}_h$ being the partitioning of $\bar{\Omega}$ into elements consisting of triangles / tetrahedra, have to be introduced (for details see Franca, Stenberg [6] and Franca, Hughes, Stenberg [5]). The value of h_K represents the size of element K.

2.2 A General Algebraic Multigrid Method

In this section we describe the construction of an AMG algorithm for a set of linear equations

$$K_1 x = b_1, \tag{17}$$

where K_1 is a regular $n_1 \times n_1$ matrix.

First we have to create a full rank prolongation matrix P_2^1, with $P_2^1 : \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}$ and $n_2 < n_1$. We want that for this purpose only information from some auxiliary matrix H_1, representing a virtual FE mesh, is used. Sensible choices for this matrix would be $H_1 = K_1$ or

$$(H_1)_{i,j} = \begin{cases} -1/\|e_{i,j}\| & \text{if } i \neq j \text{ and vertex } i \text{ and } j \\ & \text{are connected,} \\ \sum_{k\neq i} 1/\|e_{i,k}\| & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where $\|e_{i,j}\|$ is the length of the edge connecting the nodes i and j. Details can be found in Reitzinger [14].

We also need a restriction matrix $R_1^2 : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$. If not stated differently we use $R_1^2 = (P_2^1)^T$. Now we can build the Galerkin projected matrix

$$K_2 = R_1^2 K_1 P_2^1, \tag{19}$$

and the auxiliary matrix on this level

$$H_2 = R_1^2 H_1 P_2^1. \tag{20}$$

Repeating this step we end up with a set of prolongation matrices P_{i+1}^i, $i = 1, \ldots, N-1$, where $P_{i+1}^i : \mathbb{R}^{n_{i+1}} \to \mathbb{R}^{n_i}$, $n_1 > n_2 > \ldots > n_N$, a set of restriction matrices R_i^{i+1} and a set of coarse level matrices K_i and auxiliary matrices H_i with

$$K_{i+1} = R_i^{i+1} K_i P_{i+1}^i \text{ and } H_{i+1} = R_i^{i+1} H_i P_{i+1}^i. \tag{21}$$

Completing the AMG method we need on each level $i = 1, \ldots, N-1$ an iterative method for the problem $K_i x_i = b_i$,

$$x_i^{j+1} = \mathcal{S}_i(x_i^j, b_i), \tag{22}$$

the smoothing operator.

Algorithm 1. Basic AMG iteration for the system $K_l x_l = b_l$.
Let m_{pre} be the number of presmoothing steps, m_{post} of postsmoothing steps. Suppose we have chosen a starting solution x_l^0 on level l.

for $k \leftarrow 1$ **to** m_{pre} **do** $x_l^k = \mathcal{S}(x_l^{k-1}, b_l)$; (presmoothing)
$b_{l+1} \leftarrow R_l^{l+1}(b_l - K_l x_l^{m_{\text{pre}}})$; (restriction)
if $l + 1 = N$
  begin
    Compute the exact solution $\bar{x}_N$ of $K_N \bar{x}_N = b_N$;
  end
else
  begin
    Apply algorithm 1 (μ times) on

$K_{l+1}x_{l+1} = b_{l+1}$
(with starting solution $x^0_{l+1} = 0$)
and get $\bar{x}_{l+1}$;

end

$x_l^{m_{\text{pre}}+1} = x_l^{m_{\text{pre}}} + P^l_{l+1}\bar{x}_{l+1}$; (prolongation and correction)

for $k \leftarrow 1$ **to** m_{post} **do** $x_l^{m_{\text{pre}}+k+1} = \mathcal{S}(x_l^{\text{pre}+k}, b_l)$; (postsmoothing)

return $\bar{x}_l = x_l^{m_{\text{pre}}+m_{\text{post}}+1}$;

The part from (restriction) to (prolongation and correction) will be referred to as "coarse grid correction". For $\mu = 1$ the iteration is called a 'V-Cycle', for $\mu = 2$ 'W-Cycle'.

Repeated application of this algorithm until fulfillment of some convergence criterion yields a basic AMG method.

For scalar symmetric positive definite (spd) systems this method (construction of prolongators and smoothers, convergence analysis) is well studied (cf. for example Ruge, Stüben [15], Stüben [19] or Reitzinger [14]). Unfortunately the problem we are interested in is neither positive definite, scalar nor symmetric (for the Navier-Stokes case), thus we have to think about some other techniques, or, more precisely, of a correct application of the standard methods.

3 AMG for the Oseen Problem

If we use AMG for the indefinite saddle point system

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ g \end{pmatrix}, \tag{23}$$

we have to adopt the basic ingredients. The prolongation has to be chosen to avoid a mixture of velocity components and pressure, thus

$$P^i_{i+1} = \begin{pmatrix} \tilde{I}^i_{i+1} & \\ & J^i_{i+1} \end{pmatrix}, \tag{24}$$

with $\tilde{I}^i_{i+1} = \begin{pmatrix} I^i_{i+1} & \\ & I^i_{i+1} \end{pmatrix}$ resp. $\tilde{I}^i_{i+1} = \begin{pmatrix} I^i_{i+1} & & \\ & I^i_{i+1} & \\ & & I^i_{i+1} \end{pmatrix}$ in 2D resp. 3D. $I^i_{i+1} : \mathbb{R}^{n_{i+1}} \to \mathbb{R}^{n_i}$ is the prolongation matrix for one velocity component, $J^i_{i+1} : \mathbb{R}^{m_{i+1}} \to \mathbb{R}^{m_i}$ for pressure. We denote the corresponding restriction matrices with $\tilde{I}^{i+1}_i$, I^{i+1}_i and J^{i+1}_i and use $\tilde{I}^{i+1}_i = (\tilde{I}^i_{i+1})^T$, $I^{i+1}_i = (I^i_{i+1})^T$, $J^{i+1}_i = (J^i_{i+1})^T$. The system matrix on the level i is denoted by

$$\begin{pmatrix} A_i & B_i^T \\ B_i & -C_i \end{pmatrix}. \tag{25}$$

Most parts of the block-system can now be projected to the coarser level as described in section 2.2. We only have to take care of the following two terms, where we have a non-standard h-dependence. First there is the factor β_h in the convection-stabilization (12), which should be of order $\mathcal{O}(h)$. This introduces a mesh-dependence of the stabilizing term different from a differential operator. Thus we propose to assemble and coarsen the corresponding matrix A_2 separately, multiply it on each level with a suitable scaling factor and then add it to the matrix A. Numerical experiments have shown that a non-differentiated strategy may cause problems.

The second non-standard term is the stabilizing lower right matrix in the P1-P1stab case. For reasons which will become apparent in the proof of lemma 6 we propose the following. Set

$$\tilde{C}_1 = C_1, \quad \tilde{C}_{i+1} = J_i^{i+1} \tilde{C}_i J_{i+1}^i, \text{ for } i \geq 1 \tag{26}$$

and

$$C_{i+1} = \frac{\lambda_{\max}(D_i^{-1} M_i)}{h^2} \tilde{C}_{i+1}, \text{ for } i \geq 1, \tag{27}$$

where M_i is the Galerkin projection of the mass matrix M_1 to level i and D_i the diagonal of one (component-) block of the Galerkin projection of the vector-Laplace matrix A_L. For practical computation we will use very rough estimates for $\lambda_{\max}(D_i^{-1} M_i)$.

Apart from the coarsening we also have to ensure that the smoother is a suitable iterative method for saddle point problems.

The following subsections describe these main components, the construction of the coarse level systems and the smoother, in detail.

3.1 The Coarse Level Systems

For scalar systems many possibilities for the construction of the coarse levels can be found (see Stüben [19]). We exemplarily show the generalization of one method from this field, the red-black-coloring algorithm. Other techniques could be applied accordingly.

Details of this algorithm can be found in Kickinger [9], we will only give a brief overview. Assume we have constructed a fine level graph G_i (induced by the auxiliary matrix H_i) representing the connections of the fine-level nodes.

Algorithm 2. Red-Black Coloring

repeat until all the nodes are colored
    begin
        choose an uncolored node
            (e.g. with minimal node number);
        this node is colored black;
        all uncolored neighboring nodes are colored red;
    end
return coloring

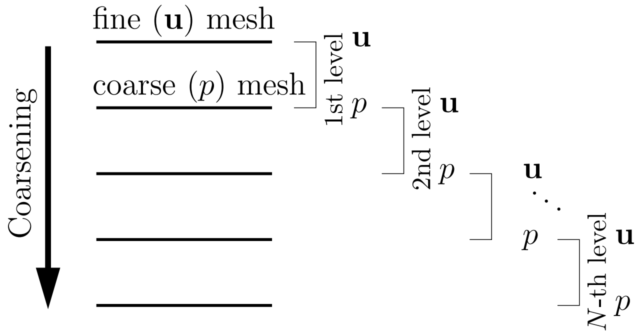

Figure 1: Using red-black-coloring coarsening we get a hierarchy of $\mathbf{u}$-p-levels by shifted selection of the “scalar levels”.

The black nodes are used as coarse-level nodes, the prolongation $\tilde{P}^i_{i+1}$ is constructed (in the scalar case) as

$$(\tilde{P}^i_{i+1})_{i,j} = \begin{cases} 1 & \text{if the black node } j \text{ has the number } i \\ & \text{in the fine level graph,} \\ \frac{1}{m_i} & \text{if } i \text{ is fine level number of a red} \\ & \text{node, } j \text{ a neighboring black node,} \\ 0 & \text{otherwise,} \end{cases} \tag{28}$$

where m_i is the number of neighboring black nodes of fine level node i.

A nice property of this algorithm is the fact that for a hierarchical uniformly refined grid (where the numbering of the coarse grid nodes remains unchanged in the fine grid) the resulting coarse grid systems and prolongations are exactly the same as if a geometric multigrid (GMG) method for this grid-hierarchy was used.

Consider now a GMG method for the P1isoP2-P1 discretized mixed problem. If we use uniform refinement, then the shape functions for the velocities on one level are exactly the pressure shape functions on the next finer level.

Reversing this observation yields a first idea for an AMG method. We start with some scalar auxiliary matrix H_1, and generate a set of prolongation matrices $\tilde{P}^i_{i+1}$. Then we use $J^i_{i+1} = \tilde{P}^i_{i+1}$ and $I^i_{i+1} = \tilde{P}^{i-1}_i$ as illustrated in figure 1 ($\tilde{P}^0_1$ is defined as the trivial interpolation from the fine “pressure mesh” to the fine “velocity mesh”).

Remark 3. In real life problems it is often not possible to apply the classical P1isoP2-P1 element, the necessary refinement would consume too much memory. We can overcome this problem by using the given mesh as “velocity mesh” and take the first coarsened level as first pressure mesh. We only have to take care about a possible loss of stability (section 3.3).

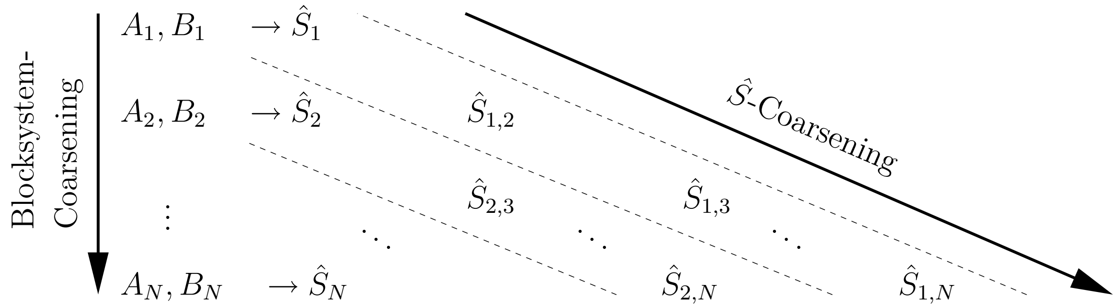

Figure 2: If we use Galerkin projections for the inner AMG, we have to build and keep all the matrices $\hat{S}_{i,j}$

3.2 Smoothing

We present the application of two well known smoothers for saddle point systems to the AMG case.

3.2.1 Braess-Sarazin Smoothers

This smoother (presented by Braess and Sarazin [2] and generalized by Zulehner [26]) is applicable to saddle point problems of the form (23) and consists of repeated application of

$$\hat{A}(\hat{\mathbf{u}}^{j+1} - \mathbf{u}^j) = \mathbf{f} - A\mathbf{u}^j - B^T p^j, \tag{29}$$

$$\hat{S}(p^{j+1} - p^j) = B\hat{\mathbf{u}}^{j+1} - Cp^j - g, \tag{30}$$

$$\hat{A}(\mathbf{u}^{j+1} - \hat{\mathbf{u}}^{j+1}) = -B^T(p^{j+1} - p^j), \tag{31}$$

where $\hat{A}$ is a preconditioner for A and $\hat{S}$ for $C + B\hat{A}^{-1}B^T$ (for details we refer to the articles mentioned above). Concretely we perform one iteration of a symmetric Gauss-Seidel or ω-Jacobi procedure as preconditioner for A in the first and third step. For the solution of (30) we use an (inner) AMG method.

The application of this smoother in the algebraic context is straightforward as no geometric information is needed.

Remark 4. We emphasize that if we want to have Galerkin projected matrices as coarse matrices for the inner AMG method, it is not possible (at least not straightforward) to use the Galerkin projected A_i and B_i from the outer AMG. What we need is $J_i^{i+1} B_i \hat{A}_i^{-1} B_i^T J_{i+1}^i$, what we get from the outer process are $\tilde{I}_i^{i+1} B_i^T J_{i+1}^i$ and $\tilde{I}_i^{i+1} A_i \tilde{I}_{i+1}^i$. Thus we have to perform the coarsening for $\hat{S}$ on each level, which unfortunately leads to an increase of memory usage, illustrated in figure 2.

3.2.2 Schwarz-type Smoothers

A widely used type of smoother is the class of multiplicative Vanka-iterations, where smaller local problems are solved and combined via a multiplicative Schwarz method (MSM). In

Schöberl, Zulehner [17] an additive variant (ASM) of this smoother is analyzed.

We define the sub-problems on a fixed level i via a set of linear prolongation operators $U_j^i : \mathbb{R}^{n_{i,j}} \to \mathbb{R}^{n_i}$ for each velocity component and $Q_j^i : \mathbb{R}^{m_{i,j}} \to \mathbb{R}^{m_i}$ for pressure, where $n_{i,j} \ll n_i$ and $m_{i,j} \ll m_i$.

Now for the smoothing of a saddle point system (23) we successively solve the local residual systems (we drop the indices indicating level i)

$$\begin{pmatrix} \hat{A}^j & {B^j}^T \\ B^j & B^j {\hat{A}^{j}}{}^{-1} {B^j}^T - \hat{S}^j \end{pmatrix} \begin{pmatrix} \mathbf{w}^j \\ r^j \end{pmatrix} = \begin{pmatrix} U_j^T(\mathbf{f} - A\mathbf{u} - B^T p) \\ Q_j^T(g - B\mathbf{u} + Cp) \end{pmatrix}, \tag{32}$$

with $\hat{S}^j = \frac{1}{\tau}(C^j + B^j {\hat{A}^{j}}{}^{-1} {B^j}^T)$ for some relaxation parameter $\tau > 0$, where we update $\mathbf{u}$ and p after each solution of a local problem: $\mathbf{u} \leftarrow \mathbf{u} + U_j \mathbf{w}^j$, $p \leftarrow p + Q_j r^j$.

Although this method in its original form uses mesh information, it can be applied to AMG methods. The crucial properties needed in [17] are

$$\sum_j U_j U_j^T = I, \tag{33}$$

$$U_j^T \hat{A} = \hat{A}^j U_j^T \quad \text{and} \tag{34}$$

$$Q_j^T B = B^j U_j^T, \tag{35}$$

with a suitable preconditioner $\hat{A}$ for A.

These properties can be fulfilled by using one pressure node for each subproblem plus the velocity nodes which are connected via matrix B. Then B^j consists of the non-zero entries of the j-th row of B (plus scaling), and U_j, $\hat{A}$ and $\hat{A}^j$ have to be chosen accordingly.

3.3 Stability

In classical multigrid analysis the convergence of the method is determined by the smoothing property and the approximation property. For scalar equations the latter could roughly be translated to the requirement of a coarse level not being "too far away" from the next finer level, i.e. the coarsening being not too strong. In the case of saddle point systems we also have to take care of a possible lack of stability of the coarse level systems destroying the approximation property.

3.3.1 Stability of the P1-P1stab Element

In the geometric case Franca and Stenberg [6] proved the stability of this element using the following estimate by Verfürth [22]

$$\sup_{\mathbf{0} \neq \mathbf{v} \in \mathbf{U}_h} \frac{(\operatorname{div} \mathbf{v}, p)}{\|\mathbf{v}\|_1} \geq \gamma \|p\|_0 - \delta \left(\sum_{K \in \mathcal{C}_h} h_K^2 \|\nabla p\|_{0,K}^2 \right)^{\frac{1}{2}} \quad \forall p \in Q_h. \tag{36}$$

We will now show, that a similar result holds also in the algebraic case.

Definition 5. A positive definite matrix $H = (h_{ij})$ is said to be of essentially positive type if there exists a constant $\omega > 0$ such that, for all e,

$$\sum_{i,j}(-h_{ij})(e_i - e_j)^2 \geq \omega \sum_{i,j}(-h_{ij}^-)(e_i - e_j)^2, \tag{37}$$

with

$$h_{ij}^- = \begin{cases} h_{ij} & \text{if } h_{ij} < 0, \\ 0 & \text{otherwise.} \end{cases}$$

We denote the Galerkin projection of the Laplace matrix A_L to the i-th level by A_{Li}, it's diagonal by D_i. We will use identical notation for the vector and the scalar variant of these matrices.

Lemma 6. *Assume that*

$$\underline{\alpha} h \leq h_K \leq \bar{\alpha} h \quad \forall K \in \mathcal{C}_h, \tag{38}$$

and further that A_{Li} is symmetric and of essentially positive type and that for all $\underline{\mathbf{v}}_i \in \underline{\mathbf{U}}_i$ we can find a $\underline{\Pi}_i^{i+1}\underline{\mathbf{v}}_i \in \underline{\mathbf{U}}_{i+1}$ such that

$$\|\underline{\mathbf{v}}_i - \tilde{I}_{i+1}^i \underline{\Pi}_i^{i+1}\underline{\mathbf{v}}_i\|_{D_i}^2 \leq \beta_1 \|\underline{\mathbf{v}}_i\|_{A_{Li}}^2. \tag{39}$$

Then for all levels i

$$\sup_{0 \neq \underline{\mathbf{v}} \in \underline{\mathbf{U}}_i} \frac{\underline{\mathbf{v}} B_i^T \underline{p}}{\|\underline{\mathbf{v}}\|_{A_{Li}}} \geq \gamma \|\underline{p}\|_{M_i} - \delta \left(\underline{p}^T C_i \underline{p}\right)^{\frac{1}{2}} \quad \forall p \in Q_i. \tag{40}$$

Proof. We follow the ideas presented by Franca and Stenberg [6]. Obviously

$$\|\underline{\mathbf{x}}_i\|_{M_i}^2 \leq \lambda_{\max}(D_i^{-1}M_i)\|\underline{\mathbf{x}}_i\|_{D_i}^2, \tag{41}$$

hence

$$\|\underline{\mathbf{v}}_i - \tilde{I}_{i+1}^i \underline{\Pi}_i^{i+1}\underline{\mathbf{v}}_i\|_{M_i}^2 \leq \lambda_{\max}(D_i^{-1}M_i)\beta_1 \|\underline{\mathbf{v}}_i\|_{A_{Li}}^2,$$

therefore there exists $\Pi_i^{i+1}\mathbf{v}_i \in \mathbf{U}_{i+1}$, such that

$$\|\mathbf{v}_i - \Pi_i^{i+1}\mathbf{v}_i\|_0^2 \leq \lambda_{\max}(D_i^{-1}M_i)\beta_1 \|\mathbf{v}_i\|_1^2. \tag{42}$$

From (37) we can infer that

$$\frac{2}{\omega}\underline{\mathbf{x}}^T D_i \underline{\mathbf{x}} \geq \underline{\mathbf{x}}^T A_{Li} \underline{\mathbf{x}}. \tag{43}$$

Because of

$$\begin{aligned} \|\tilde{I}_{i+1}^i \underline{\Pi}_i^{i+1}\underline{\mathbf{v}}_i\|_{A_{Li}} - \|\underline{\mathbf{v}}_i\|_{A_{Li}} &\leq \sqrt{\frac{2}{\omega}} \|\tilde{I}_{i+1}^i \underline{\Pi}_i^{i+1}\underline{\mathbf{v}}_i - \underline{\mathbf{v}}_i\|_{D_i} \\ &\leq \sqrt{\frac{2\beta_1}{\omega}} \|\underline{\mathbf{v}}_i\|_{A_{Li}} \end{aligned}$$

we see that

$$\|\tilde{I}^i_{i+1}\underline{\Pi}^{i+1}_i \underline{\mathbf{v}}_i\|_{A_{L_i}} \le \underbrace{\left(1+\sqrt{\frac{2\beta_1}{\omega}}\right)}_{=:\beta_2} \|\underline{\mathbf{v}}_i\|_{A_{L_i}}. \tag{44}$$

Now assume that for a level i for all $p_i \in Q_i$

$$(\operatorname{div}\mathbf{w}_i, p_i) \ge c_i\|\mathbf{w}_i\|_1\|p_i\|_0 - d_i\|\mathbf{w}_i\|_1\left(\underline{p}_i^T C_i \underline{p}_i\right)^{1/2} \tag{45}$$

Set $\mathbf{w}_{i+1} = \Pi^{i+1}_i \mathbf{w}_i$. Then

$$\begin{aligned}(\operatorname{div}\mathbf{w}_{i+1}, p_{i+1}) &= (\operatorname{div}(\mathbf{w}_{i+1}-\mathbf{w}_i), p_{i+1}) + (\operatorname{div}\mathbf{w}_i, p_{i+1})\\ &= (\mathbf{w}_i - \mathbf{w}_{i+1}, \nabla p_{i+1}) + (\operatorname{div}\mathbf{w}_i, p_{i+1})\\ &\ge -\left(\sum_K h_K^{-2}\|\mathbf{w}_i-\mathbf{w}_{i+1}\|^2_{0,K}\right)^{1/2}\cdot\left(\sum_K h_K^2\|\nabla p_{i+1}\|^2_{0,K}\right)^{1/2}\\ &\quad + (\operatorname{div}\mathbf{w}_i, p_{i+1}).\end{aligned}$$

Now because of (38) and (42)

$$\begin{aligned}\sum_K h_K^{-2}\|\mathbf{w}_i-\mathbf{w}_{i+1}\|^2_{0,K} &\le (\underline{\alpha}h)^{-2}\|\mathbf{w}_i-\mathbf{w}_{i+1}\|^2_0\\ &\le \frac{\beta_1}{\underline{\alpha}^2}\frac{\lambda_{\max}(D_i^{-1}M_i)}{h^2}\|\mathbf{w}_i\|_1^2,\end{aligned}$$

thus

$$\begin{aligned}(\operatorname{div}\mathbf{w}_{i+1}, p_{i+1}) &\ge -\frac{\sqrt{\beta_1}}{\underline{\alpha}}\|\mathbf{w}_i\|_1\left(\underline{p}_{i+1}^T C_{i+1}\underline{p}_{i+1}\right)^{1/2} + (\operatorname{div}\mathbf{w}_i, p_{i+1})\\ &\ge -\left(\frac{\sqrt{\beta_1}}{\underline{\alpha}} + d_i\sqrt{\frac{\lambda_{\max}(D_i^{-1}M_i)}{\lambda_{\max}(D_{i-1}^{-1}M_{i-1})}}\right)\cdot\|\mathbf{w}_i\|_1\left(\underline{p}_{i+1}^T C_{i+1}\underline{p}_{i+1}\right)^{1/2}\\ &\quad + c_i\|p_{i+1}\|_0\|\mathbf{w}_i\|_1.\end{aligned}$$

With (44) we get

$$\begin{aligned}\frac{(\operatorname{div}\mathbf{w}_{i+1}, p_{i+1})}{\|\mathbf{w}_{i+1}\|_1} &\ge \beta_2^{-1}\frac{(\operatorname{div}\mathbf{w}_{i+1}, p_{i+1})}{\|\mathbf{w}_i\|_1}\\ &\ge c_i/\beta_2\|p_{i+1}\|_0\\ &\quad -\left(\frac{\sqrt{\beta_1}}{\beta_2\underline{\alpha}} + \frac{d_i}{\beta_2}\sqrt{\frac{\lambda_{\max}(D_i^{-1}M_i)}{\lambda_{\max}(D_{i-1}^{-1}M_{i-1})}}\right)\cdot\left(\underline{p}_{i+1}^T C_{i+1}\underline{p}_{i+1}\right)^{1/2},\end{aligned}$$

hence, with

$$c_{i+1} := c_i/\beta_2 \quad \text{and}$$

$$d_{i+1} := \frac{\sqrt{\beta_1}}{\beta_2 \underline{\alpha}} + \frac{d_i}{\beta_2}\sqrt{\frac{\lambda_{\max}(D_i^{-1}M_i)}{\lambda_{\max}(D_{i-1}^{-1}M_{i-1})}}$$

and with

$$\gamma := \min_{i=1\ldots N} c_i \quad \text{and}$$

$$\delta := \max_{i=1\ldots N} d_i$$

we complete the proof. □

Remark 7. Assumption (39) is fulfilled e.g. for prolongators introduced by Ruge, Stüben [15] or Vaněk et al. [20].

3.3.2 Stability of the Modified Taylor-Hood Element

The situation is worse for this mixed element. We know that in a geometric setting the P1isoP2-P1 element is LBB stable, c.f. Bercovier, Pironneau [1] (thus the systems are stable on all levels). The key point here is, that for each edge, providing interaction between two pressure nodes, we have an additional velocity node, "stabilizing this edge".

In the algebraic case we have little control on the formation of the coarser levels, especially not on the interaction of velocity and pressure unknowns there. Experiments have shown, that this really results in a loss of regularity (mostly in the three dimensional case) which causes trouble.

At the moment we see two general possibilities for solving this problem. Firstly we could use a smoother, mighty enough to compensate the lack of approximation. In numerical tests we have discovered, that the Braess-Sarazin smoother can be used for this purpose while the local Schwarz-type smoother fails.

The second possibility is to change the gridtransfer operators for pressure and/or velocity components in some way to gain stability.

A more restrictive coarsening can lead to more stability but for the cost of a worse approximation. The convergence of the AMG method remains poor in some situations.

Thus we dismiss the idea of having the same level-hierarchy for velocity and pressure. We suppose that the hierarchy for the pressure unknowns has been built and try to adapt the velocity coarsening to gain regular systems. As in the geometric case we want the pressure nodes to form a subset of the velocity nodes, the question is how to select the additional nodes.

A first idea was the introduction of "algebraic edge nodes", i.e. for each connection of pressure nodes we introduce an additional velocity node. The (open) problem is the construction of a good prolongation.

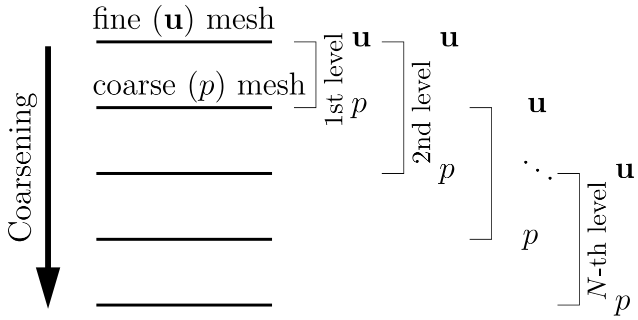

Figure 3: The strategy illustrated in figure 1 is adapted as in the first step the velocity mesh is not coarsened.

A successful strategy is to go the other way round, not to introduce additional nodes but keep more fine nodes as before. Even a first try, where at the first coarsening step we keep all the velocity nodes and do standard red-black coarsening for all subsequent steps (as illustrated in figure 3), works quite well, what will be shown in the numerical results. We will refer to the hierarchy generated by this strategy as "doubly shifted hierarchy".

Now as even this brute method has to be preferred to the original one, there remains the question about coarse levels that lie in between, i.e. if we can find a criterion determining which subset of the velocity nodes can be coarsened and which should stay untouched. This question is subject to further research.

3.4 The SIMPLE Algorithm

In the section about the numerical results we will compare our methods with a variant of the SIMPLE algorithm. In this alternative to the coupled AMG method an iterative decoupling of velocity and pressure equations is performed, and the resulting systems are solved with scalar AMG methods.

Algorithm 8. SIMPLE
Choose an approximation D of A, define the modified Schur complement $\hat{S} := C+BD^{-1}B^T$.

choose starting solutions $\mathbf{u}_0$, p_0;
$k \leftarrow 0$;
repeat until convergence
  begin
    solve $A\tilde{\mathbf{u}} = \mathbf{f} - B^T p_k$;
    solve $\hat{S}\tilde{p} = B\tilde{\mathbf{u}} - Cp_k - g$;
    choose a damping factor γ;
    $p_{k+1} = p_k + \gamma\tilde{p}$;
    $\mathbf{u}_{k+1} = \tilde{\mathbf{u}} - D^{-1}B^T\tilde{p}$;
    $k \leftarrow k + 1$;
  end

Remark 9. Algorithm 8 represents a simple version of this class of algorithms. Often the nonlinear iteration is included in the iteration, for details see Patankar [10] or Griebel et al. [7].

We use $C = \operatorname{diag} A$ (therefore $\hat{S}$ can be computed explicitely), one AMG iteration for the first (A) equation and an AMG method for a reduction (about 0.01) in the residual of the second ($\hat{S}$) equation. The factor γ is chosen as large as possible but small enough for not destroying convergence.

4 The Navier Stokes Problem

Our goal is the solution of the nonlinear Navier-Stokes problem, thus we have to develop some process dealing with the nonlinearity. This should not be a main subject of this paper, but we will discuss shortly our strategies.

We have already stated, that we solve Oseen-linearized systems originating in some (damped) fixed point iteration. In the situation of small viscosity ν and stationary equations even a damped process is hard to control. As this is less the case when solving the instationary problem, we introduce a pseudo time term, i.e. we obtain an iterative process where $\mathbf{u}_i$ and p_i satisfy

$$\begin{pmatrix} A(\mathbf{u}_{i-1}) + \alpha \bar{M} & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \mathbf{u}_i \\ p_i \end{pmatrix} = \begin{pmatrix} \mathbf{f} + \alpha \bar{M} \mathbf{u}_{i-1} \\ g \end{pmatrix}, \tag{46}$$

where $\bar{M}$ is the mass matrix or (as we are not interested in the correct reconstruction of a non-stationary process) a lumped mass matrix and α a (small) parameter.

Besides the stabilization of the nonlinear process, this method has the nice property that it increases the symmetry of the linear systems.

Another possibility to moderate problems of the linear solver and gain some efficiency (with respect to computing time) in situations with dominating convection is to use the AMG method not as linear solver but as preconditioner for a Krylov-space method, e.g. GMRES(m) (i.e. GMRES, restarted after m steps).

5 Numerical Results

All numerical tests were performed using the software package AMuSE (<u>A</u>lgebraic <u>Mu</u>ltigrid for <u>S</u>tokes-type <u>E</u>quations) on a standard (Linux-) PC. AMuSE was developed by Ferdinand Kickinger (until 2000), Christoph Reisinger (until 2000) and the author (at the moment sole developer). It is written in C++ and makes extensive use of the object oriented features of this programming language.

In the following examples the convergence rates are defined as $\|\mathbf{r}_{i+1}\|_2 / \|\mathbf{r}_i\|_2$, where $\mathbf{r}_i$ is the residual of the whole system, and i is large enough, such that we measure the asymptotic behavior. The asymptotic efficiency was measured analogously, here we measured the reduction of the residual per minute of computation time.

Schwarz smoother			Braess smoother		
cycle	assympt. factor	assympt. red./min.	cycle	assympt. factor	assympt. red./min.
W-2-1	no convergence		W-5-5	no convergence	
W-3-3	0.40	2.0e-10	W-6-6	0.67	1.5e-5
W-4-3	0.34	9.5e-11	W-7-7	0.69	8.1e-4
W-5-5	0.30	1.1e-8	W-10-10	0.50	8.4e-5
W-6-6	0.27	7.1e-8	W-12-12	0.47	1.8e-4
GMRES(m)+Schwarz smoother			GMRES(m)+Braess smoother		
m	cycle	red./min.	m	cycle	red./min.
3	W-4-3	1.1e-11	3	W-10-10	2.9e-4
4	W-4-3	1.1e-11	4	W-10-10	1.9e-6

Table 1: 2D valve, stationary Stokes problem

5.1 2D Example

In two space dimensions we regard the problem of the flow through a half opened valve (geometry provided by the AVL List GmbH, Graz, Austria). The mesh was generated by Ferdinand Kickinger's NAOMI (see [8]), it consists of 11621 nodes, which we used for the velocity unknowns, the 3088 pressure unknowns live on the next coarser level. Three levels were generated, the coarsest level consists of 589 "velocity nodes" and 108 "pressure nodes".

In table 1 we show the behavior of different kinds of smoothers and solvers for the stationary Stokes problem on this geometry.

The more interesting case is the Navier-Stokes problem, here with $\nu = 0.0005$, mean velocity at inlet 0.93, distance between the walls at the narrowest part 0.003. In table 2 we show the convergence results for the Oseen problem with convection speed near the solution and $\alpha = 1$ in (46). In figure 4 we compare the convergence history of different solvers for the Oseen and the Stokes Problem, in figure 5 the solution is visualized.

5.2 3D Examples

5.2.1 Modified Taylor-Hood element

The first example with a three dimensional domain is a flow entering the geometry through two separated supplying pipes, passing two valves and entering a cylinder which it leaves again at two holes in the bottom (geometry provided by the AVL List GmbH). The mesh was generated with Joachim Schöberl's Netgen (see [16]). The finest level consists of 687645 velocity unknowns and 29483 pressure unknowns. The stationary Navier-Stokes problem is solved for $\nu = 0.001$, velocity at inlet 0.5, distance between walls at narrowest part 0.03.

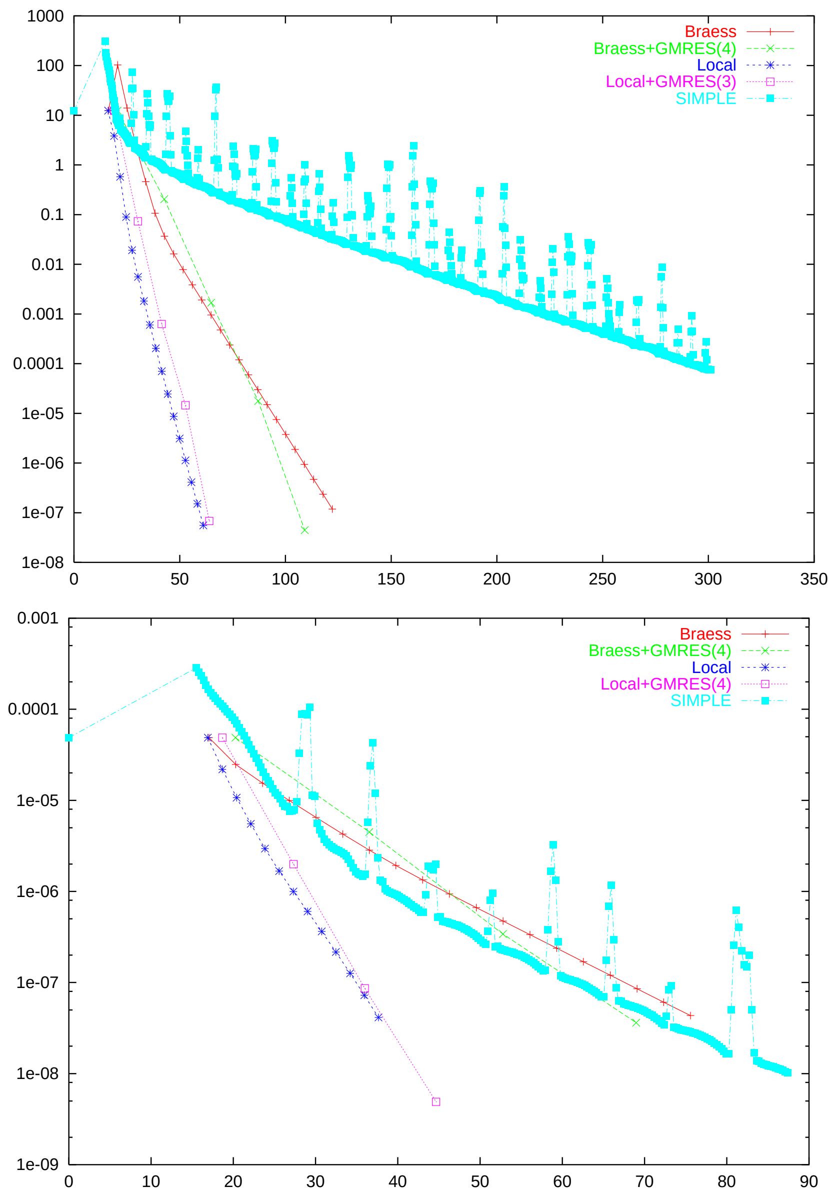

Figure 4: 2D valve, comparison of different solvers/smoothers first for the Stokes Problem and then for the Oseen-Problem with $\nu = 0.0005$ (x-axis: CPU time [s], y-axis: norm of residual).

Schwarz smoother			Braess smoother		
cycle	assympt. factor	assympt. red./min.	cycle	assympt. factor	assympt. red./min.
W-1-1	no convergence		W-5-5	no convergence	
W-2-2	0.60	2.2e-8	W-6-5	0.80	4.8e-3
W-3-2	0.55	4.5e-8	W-7-7	0.73	3.4e-3
W-3-3	0.50	6.9e-8	W-8-8	0.71	4.5e-3
W-6-6	0.37	3.9e-6	W-12-12	0.65	8.8e-3
GMRES(m)+Schwarz smoother			GMRES(m)+Braess smoother		
m	cycle	red./min.	m	cycle	red./min.
3	W-2-2	6.0e-9	3	W-7-7	9.8e-4
4	W-2-2	3.7e-10	4	W-7-7	1.1e-4

Table 2: 2D valve, Oseen problem

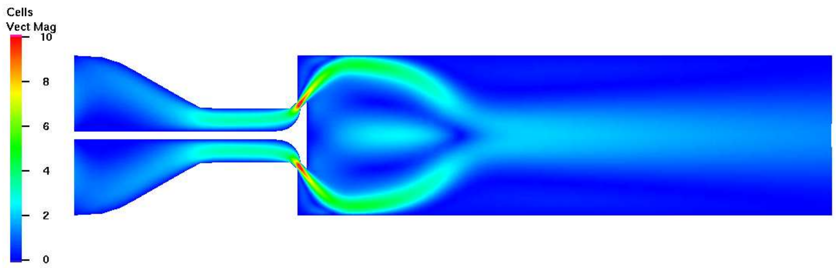

Figure 5: Norm of the velocity. For the computation only one half of the geometry shown (with symmetry boundary conditions) was used as domain.

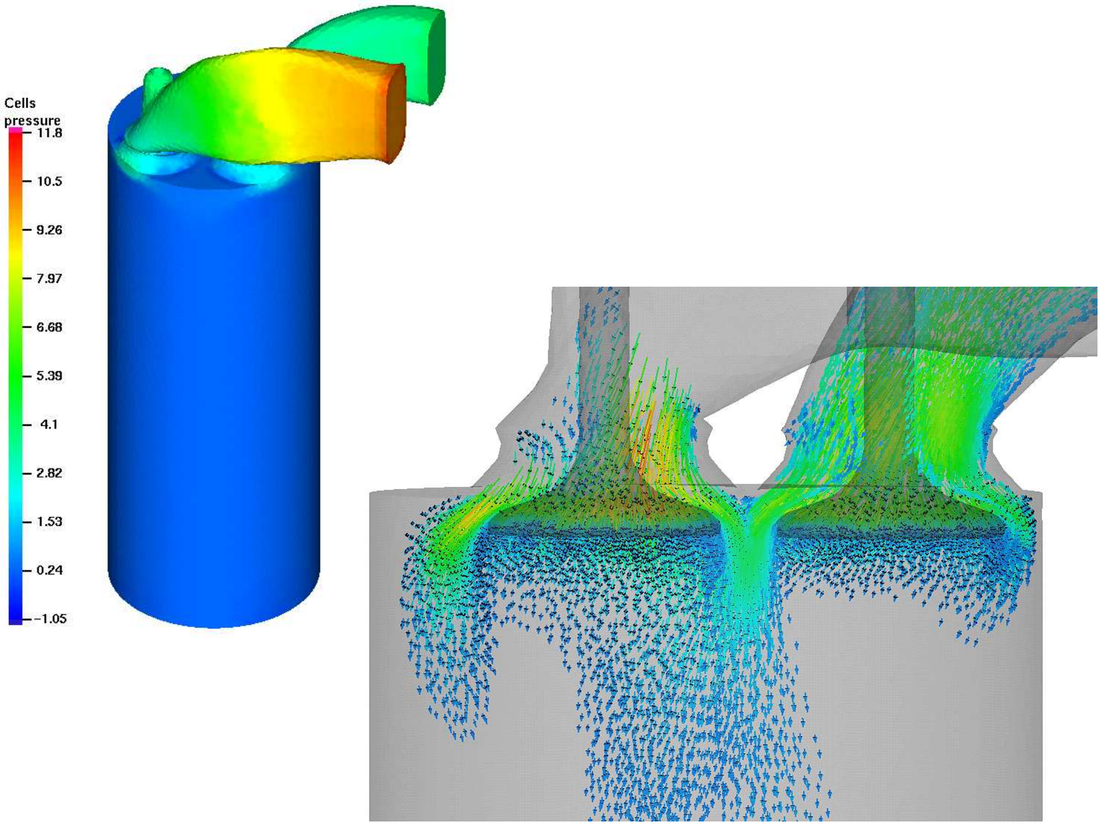

Figure 6: The surface of the domain colored by the pressure values and a part of the flow passing the valves, visualized by vectors colored by their magnitude (blue means slow, red fast).

In figure 6 parts of the solution are shown, tables 3, 4 and figure 7 compare the different solution strategies for the Oseen problem (with convection speed near the solution and $\alpha = 0$ in (46)) and for the Stokes problem.

One could suppose that the better performance of the doubly shifted hierarchy originates only in the increase of smoothing steps on the fine (velocity-) level. In figure 7 and table 4 we illustrate that this is not the case, as we plot for comparison the convergence history of the standard method with twice the number of smoothing steps on the finest level.

We omit the rates for the AMG method with Schwarz smoother, as the standard method does not converge (at least not with a sensible number of smoothing steps), and for the doubly shifted hierarchy the size of the local problems explodes, thus the method becomes very inefficient.

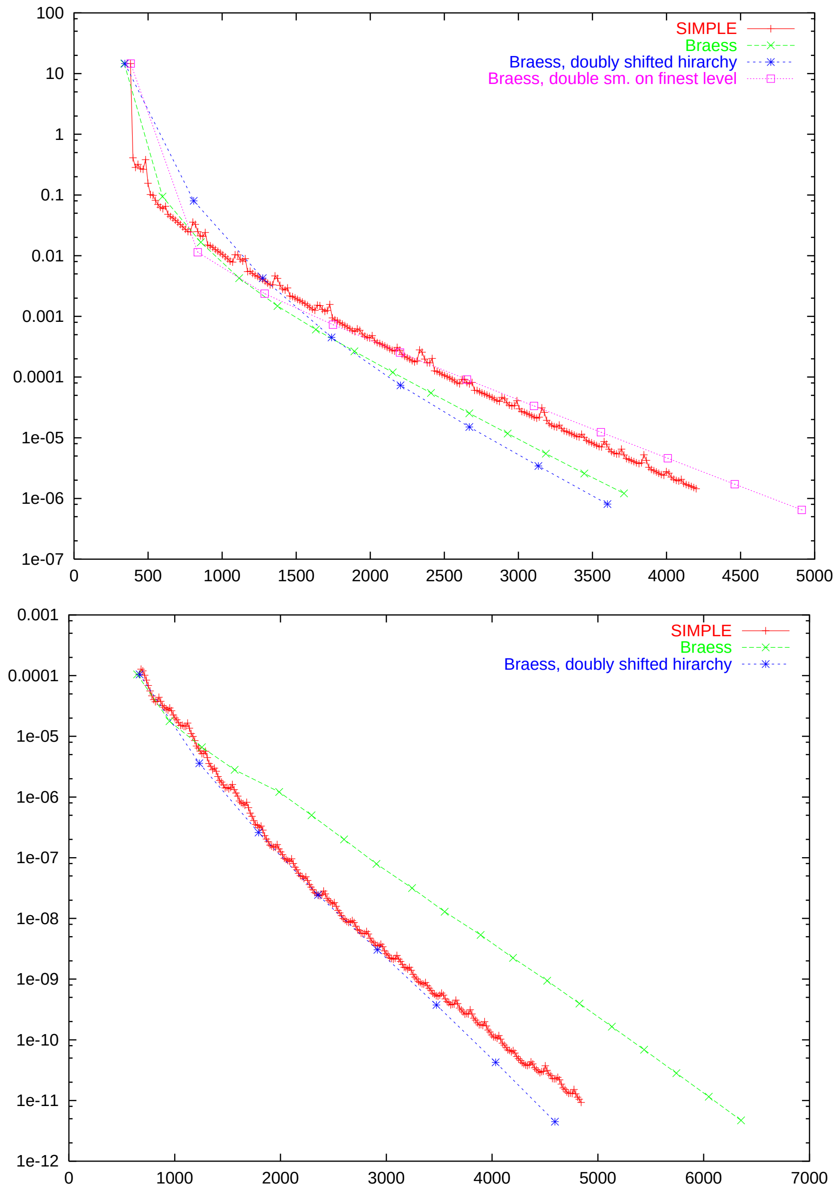

Figure 7: 3D valve, comparison of different solvers/smoothers first for the Stokes Problem and then for the Oseen-Problem with $\nu = 0.001$ (x-axis: CPU time [s], y-axis: norm of residual).

smoother/solver	cycle	assympt. factor	assympt. red./min.
Braess, shif. red/black	W-14-13	0.41	0.84
Braess, double shifted r/b	W-12-11	0.12	0.76
Braess, double shifted r/b	W-14-13	0.10	0.79

Table 3: 3D valves, Oseen problem

smoother/solver	cycle	assympt. factor	assympt. red./min.
Braess, shif. red/black	W-11-11	0.43	0.84
Braess, double shifted r/b	W-11-11	0.21	0.81
Braess, shif. red/black	W-11-11 (22-22 on 1st level)	0.37	0.88

Table 4: 3D valves, Stokes problem

5.2.2 Stabilized P1-P1 Element

The second three dimensional example is a geometry with two inlet ports in an angle of $\frac{\pi}{2}$ and two smaller outlets, a so called rotax (mesh provided by the AVL List GmbH). Here, a multi-element mesh is used, consisting of 302 tetrahedra, 142339 hexahedra, 5095 pyramids and 10019 prisms with triangular basis. The discretization on this kind of mesh is done in that way, that the elements are split to tetrahedra, there the standard discretization is applied, and the unknowns on the resulting additional nodes are locally eliminated.

Here we have used the P1-P1stab element.

In figure 9 we show the convergence of the nonlinear process solving the Navier-Stokes problem for $\nu = 0.0005$ (with $\alpha = 5$ in (46)), velocity at inlet 0.05, diameter of outlet 0.045. In figure 8 we compare the convergence-history for the coupled approach and SIMPLE for the solution of the Oseen-problem near the solution, $\alpha = 0$ in (46), and see that for the P1-P1stab element our method performs nicely. The rates can be found in table 5, figure 10 illustrated the solution.

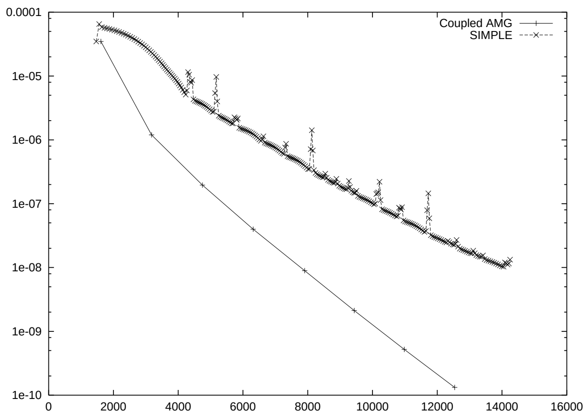

Figure 8: 3D rotax, residuals of the solution of the Oseen-Problem. Comparison of coupled versus SIMPLE approach (x-axis: CPU time [s], y-axis: norm of residual).

Solver	assympt. red./min.
Coupled (with Braess sm.)	0.95
SIMPLE	0.97

Table 5: 3D rotax, P1-P1-stab element, Oseen problem

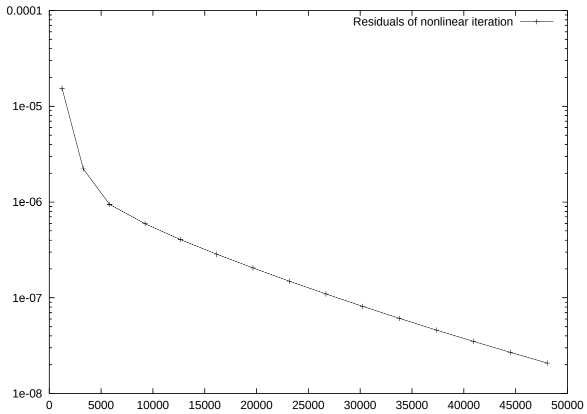

Figure 9: 3D rotax, residuals of the nonlinear iteration (x-axis: CPU time [s], y-axis: norm of residual). Starting solution was the Stokes flow through this geometry.

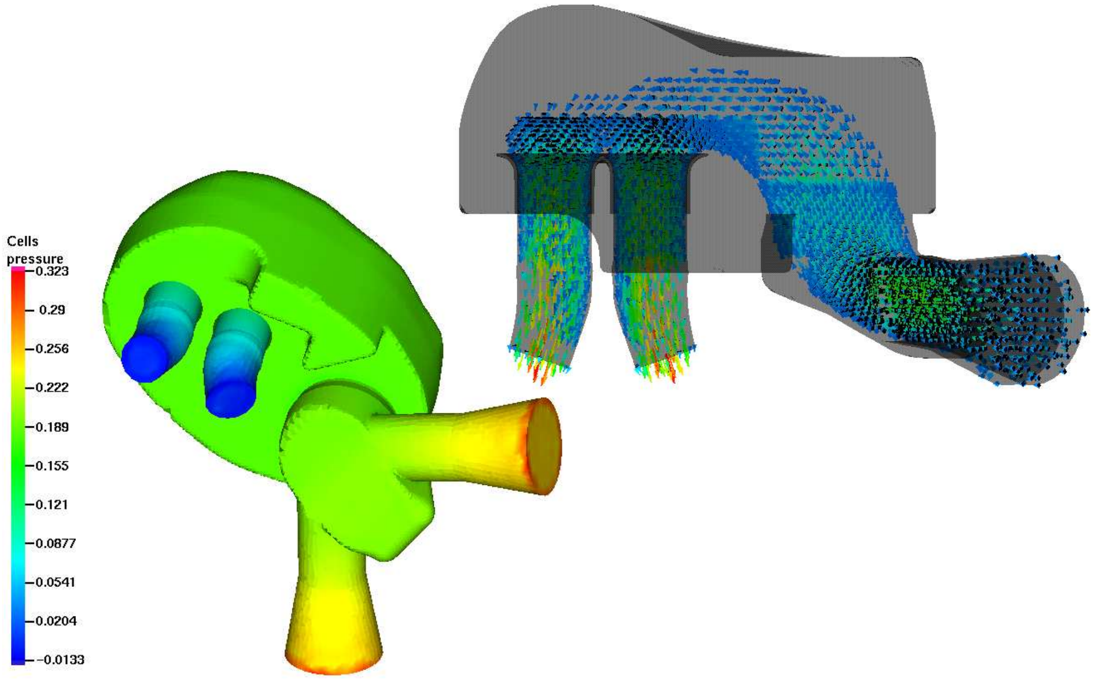

Figure 10: Pressure on the surface of the geometry and main part of the flow.

6 Conclusions

In this paper we have investigated several applicable components for an AMG solver for the Oseen problem. We have seen, that it is hardly possible to construct a 'black-box' method, as the behavior of the solver depends for example strongly on the chosen finite element pairing and its stability properties.

Although there is still much potential in the development of an optimal AMG solver for this type of problems, we have provided a strategy which already can compete with methods using the classical framework of SIMPLE or similar algorithms and performs better in most situations.

Acknowledgements

The author would like to thank Ferdinand Kickinger from AVL List GmbH for providing him with industrial test cases, and Walter Zulehner, J.K. University, Linz, for many fruitful discussions.

References

[1] M. Bercovier and O. Pironneau. Error estimates for finite element method solution of the Stokes problem in the primitive variables. *Num. Math.*, 33:211–224, 1979.

[2] D. Braess and R. Sarazin. An efficient smoother for the Stokes problem. *Applied Numerical Math.*, 23:3–20, 1997.

[3] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D.J. Evans, editor, *Sparsity and its applications*, pages 257–284. Cambridge University Press, Cambridge, 1984.

[4] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer series in computational mathematics. Springer, New York, 1991.

[5] L.P. Franca, T.J.R. Hughes, and R. Stenberg. Stabilized finite element methods for the Stokes problem. In M. Gunzburger and R.A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, chapter 4, pages 87–107. Cambridge University Press, 1993.

[6] L.P. Franca and R. Stenberg. Error analysis of some galerkin least squares methods for the elasticity equations. *SIAM J. Numer. Anal.*, 28(6):1680–1697, 1991.

[7] M. Griebel, T. Neunhoeffer, and H. Regler. Algebraic multigrid methods for the solution of the Navier-Stokes equations in complicated geometries. *Int. J. Numer. Methods Fluids*, 26:281–301, 1998.

[8] F. Kickinger. Automatic microscaling mesh generation. Institusbericht No. 525, Department of Mathematics, Johannes Kepler University, Linz, 1997.

[9] F. Kickinger. Algebraic multigrid for discrete elliptic second-order problems. In Wolfgang Hackbusch, editor, *Multigrid Methods V, Proceedings of the 5th European Multigrid conference*, volume 3 of *Springer Lecture Notes in Computational Science and Engineering*, pages 157–172, 1998.

[10] S. Patankar. *Numerical heat transfer and fluid flow.* Series in Computational Methods in Mechanics and Thermal Sciences. McGraw-Hill, New York, 1980.

[11] O. Pironneau. *Finite Element Methods for Fluids.* John Wiley & Sons, Chichester, 1989.

[12] R. Rannacher. Finite Element Methods for the Incompressible Navier-Stokes Equations. In Giovanni Galdi et al., editors, *Fundamental directions in mathematical fluid mechanics*, pages 191–293. Birkhäuser, Basel, 2000.

[13] M.J. Raw. A coupled algebraic multigrid method for the 3D Navier-Stokes equations. In W. Hackbusch and G. Wittum, editors, *Fast Solvers for Flow Problems, Proc. 10th GAMM-Seminar*, volume 49 of *Notes on Numerical Fluid Mechanics*, pages 204–215, Wiesbaden, 1995. Vieweg.

[14] S. Reitzinger. *Algebraic Multigrid Methods for Large Scale Finite Element Equations.* PhD thesis, Johannes Kepler University, Linz, 2001.

[15] J.W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. McCormick, editor, *Multigrid Methods*, volume 5 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, 1986.

[16] J. Schöberl. Netgen — an advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41–52, 1997. Netgen homepage: `www.sfb013.uni-linz.ac.at/~joachim/netgen/`.

[17] J. Schöberl and W. Zulehner. On additive schwarz-type smoothers for sadddle point problems. SFB-Report No. 01-20, SFB F013, Johannes Kepler University, Linz, 2001.

[18] K. Stüben. An introduction to algebraic multigrid. In U. Trottenberg, C. Oosterlee, and A. Schüller, editors, *Multigrid*, pages 413–532. Academic Press, New York, 2001.

[19] K. Stüben. A review of algebraic multigrid. *Computational and Applied Mathematics*, 128:281–309, 2001.

[20] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996.

[21] S. Vanka. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Comp. Meth. Appl. Mech. Eng.*, 59(1):29–48, 1986.

[22] R. Verfürth. Error estimates for a mixed finite element approximation of the Stokes equations. *R.A.I.R.O. Num. Anal.*, 18(2):175–182, 1984.

[23] R. Verfürth. A multilevel algorithm for mixed problems. *SIAM J. Numer. Anal.*, 21:264–271, 1984.

[24] R. Webster. An algebraic multigrid solver for Navier-Stokes problems. *Int. J. Numer. Meth. Fluids*, 18:761–780, 1994.

[25] G. Wittum. On the convergence of multi-grid methods with transforming smoothers. *Numer. Math.*, 57:15–38, 1990.

[26] W. Zulehner. A class of smoothers for saddle point problems. *Computing*, 65(3):227–246, 2000.