Applied Numerical Math. 23, 3–20 (1997)
with a note added after publication

An Efficient Smoother for the Stokes Problem

D. Braess* and R. Sarazin*

Abstract. We present an efficient smoother for the solution of the Stokes problem by multigrid methods. The smoother is obtained from a variant of the pressure correction steps in SIMPLE type algorithms. On the other hand the complete SIMPLE iteration does not provide good smoothers. The differences give rise to a distinction of two types of iterations although the types are seldom encountered in their pure form. The differences are less significant in practical computations due to additional approximations.

A smoothing and convergence rate $O(m^{-1})$ is achieved where m denotes the number of smoothing steps. Here the usual pressure correction step in SIMPLE iterations is understood as a Jacobi type iteration. A Gauss-Seidel version has turned out to be more robust and preferable. This is obvious from our numerical results for stationary and instationary problems.

1. Introduction

When multigrid algorithms for the Stokes problem (or the Navier Stokes equations) are constructed, the design of the smoothing procedure is not an easy task.

The well-known procedures for the smoothing of scalar elliptic problems cannot be applied here in the standard way since the matrices are not positive definite when they arise from the discretization of saddle point problems. For this reason the early multigrid procedures delt with the squared systems [7, 18, 14]. Although the convergence of that multigrid method has been proven in [18], the method has not become popular due to its low efficiency.

Therefore, now two other methods are used in most cases. The first method refers to the distributed iterations [5, 20, 21, 19], and the second one refers to an approximate decoupling of the system

$$\begin{aligned} Au + B^T p &= f, \\ Bu \qquad\quad &= 0, \end{aligned} \tag{1.1}$$

see [11, 16]. The starting point is often an approximation of the block Cholesky decomposition of (1.1). Moreover, the matrix A and its Schur complement $S := BA^{-1}B^T$ are positive definite.

We will establish a smoothing procedure which is implicitly also related to a splitting process. The solution of (1.1) characterizes the point on a subspace at which a quadratic

* Mathematisches Institut, Ruhr-Universität, 44780 Bochum, Germany
e-mail: braess@num.ruhr-uni-bochum.de; Regina.Sarazin@quantum.de

form attains its minimum. In the iteration process the quadratic form A will be replaced by a simpler one. If we had applied our procedure formally to variational problems without restrictions [these are here two Poisson equations], we had obtained the classical multigrid procedure with smoothing by the Jacobi or Gauss-Seidel iteration. We have to solve an equation of Poisson type in each smoothing step. This is no drawback since there are efficient Poisson solvers, and Poisson solvers are already used in many iterative solvers for the Stokes problem. Moreover, in this context the Poisson equations need only be solved with low accuracy.

We also hope to shed some light into the methods which are related to the SIMPLE method introduced by Patankar and Spalding [10, 9]. Those authors split the iteration into two steps. In the first step only new values of the velocities are computed. In the second step which is called "pressure correction step" the pressures are adapted and the velocities are changed such that $\operatorname{div} u = 0$ holds. There are different variants of the SIMPLE method, and we will see that not all variants are suitable in multigrid algorithms.

Although we focuss only on the stationary Stokes problem, the algorithm is also suited for the treatment of instationary problems and the Navier-Stokes equations.

2. Notations

Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, or 3. We will write the Stokes problem in its weak form.

Find $u \in X := H_0^1(\Omega)^d$ and $p \in Y := L_{2,0}(\Omega)$ such that

$$\begin{aligned} a(u, v) + b(v, p) &= \langle f, v \rangle \quad \text{for all } v \in X, \\ b(u, q) \qquad\quad &= 0 \qquad\quad \text{for all } q \in Y. \end{aligned} \tag{2.1}$$

Here, $f \in X'$ is given, and

$$\begin{aligned} a(u, v) &:= \int_\Omega \nabla u \nabla v dx, \\ b(v, q) &:= -\int_\Omega \operatorname{div} v q dx. \end{aligned} \tag{2.2}$$

The spaces $H^s(\Omega)$ and $H_0^s(\Omega)$ denote the usual Sobolev spaces endowed with the Sobolev norms $\| \, . \, \|_s$. Furthermore $L_{2,0}(\Omega) := \{q \in L_2(\Omega);\ \int_\Omega q dx = 0\}$ refers to the space of L_2-functions, when we do not distinguish between functions which only differ by a constant.

We assume that Ω is polygonal and that T_h is a triangulation of Ω with triangles whose angles are bounded from below and whose diameters are bounded by h. Let u_h and p_h be the finite element solutions of u and p for the finite element spaces of Taylor and Hood, say $u_h \in X_h$ and $p_h \in Y_h$, cf. [6]. We will only refer to the usual approximation property

$$\|u - u_h\|_1 + h^{-1}\|u - u_h\|_0 + \|p - p_h\|_0 \leq ch \cdot \|f\|_0, \tag{2.3}$$

and the inverse property

$$\|u_h\|_1 \leq ch^{-1}\|u_h\|_0 \quad \text{for } u_h \in X_h. \tag{2.4}$$

Moreover, the stability of the saddle point problem will also be postulated.

When we consider multigrid methods, we will have sequences of nested triangulations with $h_0 > h_1 > \ldots > h_k$, see Section 4.

Throughout the paper we use the symbol c to denote a generic positive constant that is independent of the meshsize h and of the level ℓ. Note that c may take different values at different places.

3. The Basic Iteration

In the analysis of multigrid methods the smoothing property is often established by purely algebraic considerations, and the connection of the algebraic system to the elliptic problem is not needed. This is the situation here, and we will study the basic iteration for the smoothing in abstract form.

Let A be a symmetric and positive definite matrix. The linear system

$$\begin{pmatrix} A & B^T \\ B & \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{3.1}$$

characterizes the solution of the minimum problem

$$\frac{1}{2}u'Au - f'u \to \min!$$
$$Bu = g.$$

Let α be a real number which is not smaller than the maximal eigenvalue of A, i.e.,

$$v'Av \leq \alpha v'v \tag{3.2}$$

holds. The solution of the similar system

$$\begin{pmatrix} \alpha I & B^T \\ B & \end{pmatrix} \begin{pmatrix} v \\ q \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix} \tag{3.3}$$

is more easily determined,

$$\begin{pmatrix} \alpha I & B^T \\ B & \end{pmatrix}^{-1} = \begin{pmatrix} \frac{1}{\alpha}(I - B^T S_1 B) & B^T S_1 \\ S_1 B & -\alpha\, S_1 \end{pmatrix} \tag{3.4}$$

where $S_1 := (BB^T)^{-1}$. In actual computations the vectors are obtained by implementing the formulas $(BB^T)q = Bd - \alpha e$ and $v = \frac{1}{\alpha}(d - B^T q)$. Specifically, an equation of the Poisson type has to be solved when the Stokes problem is treated. We note that the

parameter α is encountered in two entries of the inverse matrix (3.4) and that only one of them refers to α^{-1}, cf. Remark 6.1 below.

In the multigrid analysis we will restrict ourselves to the auxiliary system (3.3) and the iteration (3.6). We recommend replacing the multiple of the unit matrix αI in actual computations by a preconditioner αC of A. Note that the solution of

$$\begin{pmatrix} \alpha C & B^T \\ B & \end{pmatrix} \begin{pmatrix} v \\ q \end{pmatrix} = \begin{pmatrix} d \\ e \end{pmatrix} \tag{3.5}$$

is obtained by the solution of $(BC^{-1}B^T)q = BC^{-1}d - \alpha e$ and $v = \frac{1}{\alpha}C^{-1}(d - B^T q)$. The algebraic considerations of this section remain true. In particular C will often be a diagonal matrix of the form $C := \mathrm{diag}(A)$.

Let $(u_j, p_j)'$ be the current approximation in the iteration process. If the equation for the correction is solved with the simplified matrix, we obtain the procedure

$$\begin{pmatrix} u_{j+1} \\ p_{j+1} \end{pmatrix} = \begin{pmatrix} u_j \\ p_j \end{pmatrix} - \begin{pmatrix} \alpha I & B^T \\ B & \end{pmatrix}^{-1} \left\{ \begin{pmatrix} A & B^T \\ B & \end{pmatrix} \begin{pmatrix} u_j \\ p_j \end{pmatrix} - \begin{pmatrix} f \\ g \end{pmatrix} \right\}. \tag{3.6}$$

Note that u_{j+1} satisfies the restriction which is defined with the original variational problem, i.e., $Bu_{j+1} = g$ holds.

In our analysis we assume that the iteration (3.6) is performed exactly. In actual computations we will be content with approximate solutions. In this way we come closer to other approaches. Obviously, one has

$$\begin{pmatrix} C & B^T \\ B & \end{pmatrix}^{-1} = \begin{pmatrix} I & -C^{-1}B^T \\ & I \end{pmatrix} \begin{pmatrix} C & \\ B & -S \end{pmatrix}^{-1}$$

where $S := BC^{-1}B^T$. If C and S are approximated by incomplete LU decompositions (or diagonal matrices), then the iterations of [21] are obtained. In [2] there is also an analysis of the iteration with nonexact solutions for the Schur complement. They prove convergence in suitable energy-norms.

A straightforward calculation shows that we have for the iteration (3.6)

$$\begin{pmatrix} u - u_{j+1} \\ p - p_{j+1} \end{pmatrix} = \begin{pmatrix} \alpha I & B^T \\ B & \end{pmatrix}^{-1} \begin{pmatrix} \alpha I - A & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u - u_j \\ p - p_j \end{pmatrix}. \tag{3.7}$$

More interesting is the formula for the u-component which follows from (3.7) and (3.4).

$$u - u_{j+1} = (I - B^T(BB^T)^{-1}B)\,(I - \frac{1}{\alpha}A)\,(u - u_j). \tag{3.8}$$

The eigenvalues of the iteration matrix in (3.7) are obtained from the eigenvalue problem

$$\begin{pmatrix} \alpha I - A & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} w \\ r \end{pmatrix} = \lambda \begin{pmatrix} \alpha I & B^T \\ B & \end{pmatrix} \begin{pmatrix} w \\ r \end{pmatrix}.$$

Nonzero eigenvalues are only possible with $w \neq 0$, $Bw = 0$. By multiplying the first equation by w we obtain in this case

$$w'(\alpha I - A)w = \lambda \alpha w'w.$$

With this it follows that (cf. (3.2))

$$0 \leq \lambda < 1, \quad 1 - \lambda = \frac{w'Aw}{\alpha w'w}. \tag{3.9}$$

Recalling, that A stands for a discretization of the Laplacian, we see that the eigenvalues close to 1 are associated to smooth eigenfunctions.

We emphasize that the error of the new approximate solution is independent of the p-component of the approximation from the previous step. This is obvious from (3.7).

The Euklidean norm of vectors will be denoted as $\| \, . \, \|_{\ell_2}$.

Lemma 3.1. *Let the vectors u_j be computed by the iteration (3.6). If $\alpha \geq \lambda_{\max}(A)$, then*

$$\|u - u_{j+1}\|_{\ell_2} \leq \|u - u_j\|_{\ell_2}. \tag{3.10}$$

Proof. Since A is symmetric and positive definite, the linear mapping $I - \frac{1}{\alpha}A$ is a contraction. Moreover,

$$P := I - B^T(BB^T)^{-1}B \tag{3.11}$$

is an orthogonal projector, and (3.10) is a direct consequence of (3.8). □

Now we are in a position to prove the main result on the smoothing property.

Lemma 3.2. *Assume that $\alpha \geq \lambda_{\max}(A)$ and that $Bu_0 = g$. If u_m is computed by the iteration (3.6), then there is a (pressure) vector q such that*

$$\|A(u - u_m) + B^T q\|_{\ell_2} \leq \frac{\alpha}{em} \|u - u_0\|_{\ell_2}. \tag{3.12}$$

If the assumption $Bu_0 = g$ is not satisfied, the statement holds if m is replaced by $m - 1$ in the denominator on the right hand side of (3.12).

Proof. The matrix $M := P(I - \frac{1}{\alpha}A)P$ is symmetric, and its spectrum is contained in the intervall [0,1]. From the usual spectral decomposition argument we conclude that

$$\|(I - M)M^m\| \leq \sup_{0 \leq t \leq 1} \{t^m(1 - t)\} \leq \frac{1}{em}. \tag{3.13}$$

Since the recursion formula (3.7) is linear, we may assume without loss of generality that $f = 0$, $g = 0$, and $u = 0$. Since P is a projector and $Bu_m = 0$ for $m \geq 1$, we have

$$\begin{aligned}(I - M)u_m &= u_m - P(I - \frac{1}{\alpha}A)u_m \\ &= \frac{1}{\alpha}PAu_m = \frac{1}{\alpha}(Au_m - B^T(BB^T)^{-1}BAu_m) \\ &= \frac{1}{\alpha}(Au_m + B^T q),\end{aligned} \tag{3.14}$$

if we put $q := -(BB^T)^{-1}BAu_m$.

If $Bu_0 = g$, then $u_m = M^m u_0$, and the desired estimate follows from (3.13) and (3.14).

Otherwise, it follows from Lemma 3.1 that $\|u - u_1\|_{\ell_2} \leq \|u - u_0\|_{\ell_2}$ and $u_m = M^{m-1}u_1$. The estimate is obtained with m replaced by $m - 1$. □

Remark 3.3. We note that the extra term q in (3.14) vanishes if the matrices A and B commute. Since A and B are here discretizations of the differential operators $-\Delta$ and $-\operatorname{div}$, respectively, we expect that $B^T q$ is small and that its largest terms are located near the boundary of the domain. When the concept of transforming smoothers as in [20] is used, then the iteration matrix also contains terms which vanish in the commutative case.

The observation that (u_{j+1}, p_{j+1}) is independent of p_j and that the multiplication by the projector P may be compensated by adding an appropriate pressure term (cf. [12]) are the crucial points of this paper.

4. The Multigrid Algorithm

As usually in the multigrid framework, we have a sequence of meshsizes $h_0 > h_1 > \ldots > h_k$ with $h_{l-1} = 2h_\ell$ for $\ell = 1, 2, \ldots, k$. The corresponding finite element spaces (by Taylor and Hood) are nested

$$X_{l-1} \subset X_\ell, \quad Y_{l-1} \subset Y_\ell \quad \text{for } \ell = 1, 2, \ldots, k. \tag{4.1}$$

Here, as usual we write X_ℓ and Y_ℓ instead of X_{h_ℓ} and Y_{h_ℓ}, respectively. Actually, one is interested in the solution of (2.1) for the finest spaces X_k and Y_k. The other spaces are auxiliary ones, and one has to solve problems of the form

$$\begin{aligned} a(u_\ell, v) + b(v, p_\ell) &=< F_\ell, v > \quad \text{for } v \in X_\ell, \\ b(u_\ell, q) \qquad\quad &= 0 \qquad\quad\;\; \text{for } q \in Y_\ell. \end{aligned} \tag{4.2}$$

Without changing the symbols for the velocity and the pressure variables we will write (4.2) in matrix-vector form

$$\begin{pmatrix} A_\ell & B_\ell^T \\ B_\ell & \end{pmatrix} \begin{pmatrix} u_\ell \\ p_\ell \end{pmatrix} = \begin{pmatrix} F_\ell \\ 0 \end{pmatrix}. \tag{4.3}$$

Multigrid Algorithm (one iteration loop at level ℓ, $1 \le \ell \le k$, with m pre-smoothing steps.)
Let $(u_\ell^0, p_\ell^0) \in X_\ell \times Y_\ell$ be a given approximation of the solution of problem (4.2).

1. *Pre-Smoothing.* For $j = 0, 1, \ldots, m-1$ compute u_ℓ^{j+1}, p_ℓ^{j+1} as the solution of

$$\begin{aligned} \alpha_\ell \cdot (u_\ell^{j+1} - u_\ell^j, v)_{\ell_2} &+ b(v, p_\ell^{j+1} - p_\ell^j) \\ &= \langle F_\ell, v\rangle - a(u_\ell^j, v) - b(v, p_\ell^j) \quad \text{for } v \in X_\ell, \\ b(u_\ell^{j+1} - u_\ell^j, q) &= -b(u_\ell^j, q) \qquad\qquad\qquad\quad \text{for } q \in Y_\ell. \end{aligned}$$

2. *Intermediate pressure step.* Compute the pressure p_ℓ^{m+1} such that the ℓ_2-norm of the functional

$$\|a(u_\ell^m, .) + b(., p_\ell^{m+1}) - \langle F_\ell, .\rangle\|_{\ell_2}$$

is minimal. Moreover set $u_\ell^{m+1} = u_\ell^m$.

3. *Coarse grid correction.* Set

$$F_{\ell-1}(v) := F_\ell(v) - a(u_\ell^{m+1}, v) - b(v, p_\ell^{m+1}) \quad \text{for } v \in X_{\ell-1} \tag{4.4}$$

and let (u_{l-1}, p_{l-1}) be the solution of (4.2) for $\ell - 1$ and the functional (4.4).

If $\ell = 1$, then compute the exact solution and set $(\tilde{u}_{\ell-1}, \tilde{p}_{\ell-1}) = (u_{\ell-1}, p_{\ell-1})$.
If $\ell > 1$, compute an approximation $(\tilde{u}_{\ell-1}, \tilde{p}_{\ell-1})$ of $(u_{\ell-1}, p_{\ell-1})$ by applying $\mu = 2$ iteration steps of the multigrid algorithm on level $\ell - 1$ with zero starting values.

Set

$$u_\ell^{m+2} = u_\ell^{m+1} + \tilde{u}_{\ell-1}, \quad p_\ell^{m+2} = p_\ell^{m+1} + \tilde{p}_{\ell-1}.$$

4. *Post-Smoothing.* Compute u_ℓ^{m+3}, p_ℓ^{m+3} as the solution of

$$\begin{aligned} \alpha_\ell \cdot (u_\ell^{m+3} - u_\ell^{m+2}, v)_{\ell_2} &+ b(v, p_\ell^{m+3} - p_\ell^{m+2}) \\ &= \langle F_\ell, v\rangle - a(u_\ell^{m+2}, v) - b(v, p_\ell^{m+2}) \quad \text{for } v \in X_\ell, \\ b(u_\ell^{m+3} - u_\ell^{m+2}, q) &= -b(u_\ell^{m+2}, q) \qquad\qquad\qquad\quad \text{for } q \in Y_\ell. \end{aligned}$$

(u^{m+3}, p^{m+3}) is the result of the loop. □

We note that the intermediate pressure step is only given in order to have a complete proof. This step can be abandoned in actual computations without a substantial loss of efficiency, cf. Remark 3.3. The update of the pressure would require the solution of the equation $BB^T q_\ell = Br_\ell$ with $r_\ell = F_\ell - Av_\ell^m - B^T p_\ell^m$. So the computing effort for this step is almost the same as for one smoothing step.

The equations of Poisson type which arise in the smoothing steps need only to be solved approximately. The numerical results show that large errors can be tolerated and that the multigrid algorithm is very robust (cf. Section 7).

5. Multigrid Analysis

We will prove convergence of the 2-level iteration for sufficiently many smoothing steps. The extension to the multilevel procedure with a W-cycle is standard [1]. At the moment we are not able to prove convergence of the V-cycle since the iteration matrix is a contraction for the ℓ_2-norm, and this property has not yet been verified for the H^1-norm.

In order to avoid double indices, the consideration for the 2-level procedure will refer to the h-grid as the fine grid and the $2h$-grid as the coarse one.

As usual the analysis of the multigrid method will use discrete norms [1, 4, 18]. We will follow [18] when choosing appropriate norms for the Stokes problem. In this way we get the approximation property already from [18] although the smoothing property has to be established by different considerations.

The scale of norms will be based on a weighted L_2-norm in the product space $X_h \times Y_h$:

$$\|(v_h, q_h)\|_h := \{\|v_h\|_0^2 + h^2\|q_h\|_0^2\}^{1/2}. \tag{5.1}$$

Since $a(u, v)+b(v, p)+b(u, q)$ is a symmetric bilinear form on the finite element space $X_h \times Y_h$, there is a complete set of eigenfunctions $\{(\phi_h^i, \psi_h^i)\}$ which satisfy

$$a(\phi_h^i, v) + b(v, \psi_h^i) + b(\phi_h^i, q) = \lambda_i\big[(\phi_h^i, v)_0 + h^2(\psi_h^i, q)_0\big] \tag{5.2}$$

for all $(v, q) \in X_h \times Y_h$. In view of (5.1) the eigenfunctions may be normalized according to

$$(\phi_h^i, \phi_h^j)_0 + h^2(\psi_h^i, \psi_h^j)_0 = \delta_{ij}. \tag{5.3}$$

Given $(v_h, q_h) \in X_h \times Y_h$, there is a spectral decomposition $(v_h, q_h) = \sum_i c_i(\phi_h^i, \psi_h^i)$. With this decomposition a scale of discrete norms is defined:

$$|||(v_h, q_h)|||_s := \Big\{\sum_i |\lambda_i|^s |c_i|^2\Big\}^{1/2}. \tag{5.4}$$

Obviously,

$$|||(v_h, q_h)|||_0 = \|(v_h, q_h)\|_h. \tag{5.5}$$

For convenience, we may assume that $u_h = 0$, $p_h = 0$ is the solution of the given equations. This assumption implies that u_h^j and p_h^j coincide with the error.

The computations are performed with the matrix vector representation of the equations (2.1). As mentioned in the previous section, we will use the symbols u_h and p_h both for the functions in the finite element spaces and their vector representations. By a standard scaling argument it follows that the Euklidean norms of the vectors are equivalent to the L_2-norms multiplied by $h^{-\frac{d}{2}}$, see e.g. [4]. In order to be consistent with (5.1), without changing the notations we will choose a normalization such that

$$\begin{aligned} c^{-1}\|u_h\|_{\ell_2} \le \|u_h\|_0 \le c\|u_h\|_{\ell_2}, \\ c^{-1}\|p_h\|_{\ell_2} \le h\|p_h\|_0 \le c\|p_h\|_{\ell_2}. \end{aligned} \tag{5.6}$$

Of course, the reader will find a *smoothing property* and an *approximation property* in our proof. On the other hand, this will not be sufficient for the treatment of the mixed problem. The coarse grid correction will satisfy the restriction $\operatorname{div} u = 0$ only with respect to the $2h$-grid. The next step will bring it back to the subset of vectors which satisfy the restriction on the fine grid. A *projection property* will guarantee that this operation is performed without a deterioration of the approximation.

The smoothing property and the approximation property refer to two norms in a reciprocal manner. These norms are

$$\|u_h\|_{\ell_2} \quad \text{and} \quad |||(u_h, p_h)|||_2 \,.$$

We have chosen the $\|.\|_{\ell_2}$-norm in order to avoid an additional constant in (5.8). The equivalent L_2-norm will be more convenient in the proof. As usually, the $|||.|||_2$-norm refers to the result of a multiplication with the operator, i.e., it follows from (5.4) and (5.6) that

$$|||(u_h, p_h)|||_2 \le c^2 \left\| \begin{pmatrix} A & B^T \\ B & \end{pmatrix} \begin{pmatrix} u_h \\ p_h \end{pmatrix} \right\|_{\ell_2} . \tag{5.7}$$

Theorem 5.1. *Let $u_h = 0$, $p_h = 0$. Assume that the number of pre-smoothing steps is $m \ge 2$. Then we have the projection property*

$$\|u_h^1\|_{\ell_2} \le \|u_h^0\|_{\ell_2} \,, \quad \|u_h^{m+3}\|_{\ell_2} \le \|u_h^{m+2}\|_{\ell_2} \,, \tag{5.8}$$

the smoothing property

$$|||(u_h^{m+1}, p_h^{m+1})|||_2 \le \frac{ch^{-2}}{m-1} \|u_h^1\|_{\ell_2} \,, \tag{5.9}$$

and the approximation property

$$\|u_h^{m+2}\|_{\ell_2} \le ch^2 |||(u_h^{m+1}, p_h^{m+1})|||_2 \,. \tag{5.10}$$

Proof. The projection property (5.8) is a direct consequence of Lemma 3.1.

By construction the velocity u_h^1 satisfies $B_h u_h^1 = 0$. For this reason, we can apply Lemma 3.2 to u_h^1 and obtain the estimate

$$\|A_h u_h^{m+1} + B_h^T p_h^{m+1}\|_{\ell_2} \le \frac{\lambda_{\max}(A_h)}{e(m-1)} \|u_h^1\|_{\ell_2}, \tag{5.11}$$

since $u_h^{m+1} = u_h^m$ and p_h^{m+1} is chosen such that the left hand side of (5.11) is minimal. From the inverse property (2.4) we have $\lambda_{\max}(A_h) \le c \cdot h^{-2}$. Moreover

$$B u_h^{m+1} = 0.$$

Combining (5.7) and (5.11) we obtain (5.9).

Finally, we obtain the approximation property directly from Verfürth's results. The Taylor-Hood element satisfies the general properties which are listed in [18]. Specifically, by (4.2) in [18] the coarse-grid corrections u_{2h} and p_{2h} yield improvements such that

$$|||(u_h^{m+1} - u_{2h}\,,\, p_h^{m+1} - p_{2h})|||_0 \le ch^2 |||(u_h^{m+1},\, p_h^{m+1})|||_2. \tag{5.12}$$

Obviously, $\|u_h^{m+1} - u_{2h}\|_0$ is bounded by the left hand side of (5.12), and the equivalence of the ℓ_2-norm and the L_2-norm yields the estimate (5.10). □

The convergence of the pressure vectors follows from the convergence of the velocities and (3.7).

We emphasize that the smoothing property yields an $O(m^{-1})$-behaviour while only $O(m^{-1/2})$ was obtained for the smoothers in [18, 14, 21]. — There is also a difference between our smoothing procedure and the smoothing of the squared matrices which refers to a computational aspect. The normalization of the p-component has been chosen in (5.6) such that $\|p\|_{\ell_2} \approx h\|p\|_0$. From the theoretical point of view the choice with a normalization, say $\|p\|_{\ell_2} \approx 5h\|p\|_0$, would be equally good since the quotient $\frac{5h}{h}$ is only a constant. This change of the normalization has no influence on the computed values u_h^j and p_h^j in our algorithm, but it has when an algorithm with the squared matrices is used.

6. Iterations of SIMPLE Type

When we started our investigations, we first tried to use the SIMPLE iteration of Patankar and Spalding [10,9] as a smoother. The procedure, however, turned out to be not successful. We will give some arguments for the failure and provide a motivation for the iteration considered in Section 3. We hope to shed also some light into the variants of the SIMPLE method which are often encountured in computational fluid mechanics.

Since the algorithms deal with the algebraic equations, we assume that the given equations have the form (3.1). In all algorithms the loop for the computation of $(u_{j+1},\, p_{j+1})$ from $(u_j,\, p_j)$ consists of 2 steps.

SIMPLE Algorithm.

1. Consider the equation
$$A\tilde{u} = f - B^T p_j \tag{6.1}$$
and compute an approximation $u_{j+1/2}$ of $\tilde{u}$ [e.g., by applying the Gauss-Seidel iteration to equation (6.1)].
2. Let D be the diagonal part of A. With $C = D$ solve the equation
$$\begin{pmatrix} \alpha C & B^T \\ B & \end{pmatrix} \begin{pmatrix} v \\ q \end{pmatrix} = \begin{pmatrix} 0 \\ g - Bu_{j+1/2} \end{pmatrix}$$
approximately. To this end apply several steps of an iteration to the equation
$$(BC^{-1}B^T)q = -\alpha(g - Bu_{j+1/2}). \tag{6.2}$$

Let $\tilde{q}$ be the approximate solution. Set

$$u_{j+1} = u_{j+1/2} - \alpha^{-1} C^{-1} B^T \tilde{q}, \quad p_{j+1} = p_j + \tilde{q}. \qquad \square$$

The first step is based on the assumption that p_j is a reasonable approximation such that an improved velocity can be obtained from equation $(3.1)_1$. The second step is called *pressure correction step* and intends to decrease the residue of the equation $Bu = g$.

Remark 6.1. When the equation (3.1) is replaced by (3.3) in an iterative process, the factor α is to be chosen such that the correction does not overshoot the solution. [The same holds if we have αD instead of αI]. To obtain an appropriate damping, a value $\alpha \approx \lambda_{\max}(A)$ is a reasonable choice, if $g = 0$, i.e., if the p-component of the right hand side vanishes. This is obvious from the representation (3.4) for the inverse matrix. Similarly, a value $\alpha \approx \lambda_{\min}(A)$ is appropriate, if $f = 0$. For this reason one often finds two different relaxation parameters in variants of SIMPLE iterations.

In 1984 the SIMPLEC algorithm was introduced by Van Dormaal and Raithby [VDR]. The concept was the same as for the SIMPLE algorithm, but the approximative solution of auxiliary problems was performed by the more efficient ILU solvers. Another improvement is motivated by Remark 6.1.

In essence, there are two differences between SIMPLE and SIMPLEC. In the algorithm SIMPLEC the auxilliary diagonal matrix C is not simply derived from the diagonal part of A, but from appropriately weighted row sums of A. Therefore the entries of C are substantially smaller than the entries in the diagonal of A and underrelaxation is not any longer necessary. Moreover the approximative solutions of (6.1) and (6.2) are computed by ILU iterations, and the number of iterations varies between 1 and 20 in the literature. Similarly there are big differences in the possible relaxation parameters. — We will not be more specific since the differences have no impact on our considerations.

Remark 6.2. The SIMPLE algorithm (and the SIMPLEC algorithm, resp.) is only a poor smoother. One can recognize this by a study of the one-dimensional case. The Stokes problem on the interval $\Omega = (0, 2\pi)$ has the form

$$\begin{aligned} \gamma u - u'' + p' &= f \quad && \text{in } (0, 2\pi), \\ -u' &= 0 && \text{in } (0, 2\pi), \\ u &= u_R && \text{on } \{0, 2\pi\}, \\ \int_0^{2\pi} p\,dx &= 0. \end{aligned} \tag{6.3}$$

Here we have $\gamma = 0$ in the stationary case and $\gamma > 0$ if (6.3) arises from the time discretization of an instationary problem. Although the problem has the pathological solution $u = u_R =$ constant, the application of the algorithm above to the continuous problem gives an insight into the situation of the general case. Let $A \hat{=} \gamma\,\mathrm{Id} - \Delta$, $B^T \hat{=} \mathrm{grad}$, $B \hat{=} -\mathrm{div}$.

The SIMPLE algorithm contains a Poisson equation or Helmholtz equation with Dirichlet boundary conditions for the temporary update of the velocity (cf. step 1 of the algorithm)

$$\begin{aligned} (\gamma\,\mathrm{Id} - \Delta)\, u_{j+1/2} &= f - \nabla p_j \quad && \text{in } \Omega, \\ u_{j+1/2} &= u_R && \text{on } \Gamma := \partial\Omega. \end{aligned} \tag{6.4}$$

Another Poisson equation with "Neumann" boundary is encountered in the pressure correction:

$$\begin{aligned} -\Delta q_{j+1} &= -\alpha \operatorname{div} u_{j+1/2} \quad && \text{in } \Omega, \\ \frac{\partial q_{j+1}}{\partial n} &= 0 && \text{on } \Gamma, \\ \int_\Omega q_{j+1} dx &= 0. \end{aligned} \tag{6.5}$$

We note that the diagonal part of the matrix A is a multiple of the unit matrix if equidistant nodes are used. Since one needs an underrelaxed diagonal matrix for stability reasons, we write $\alpha C = \alpha I$ and obtain the iteration

$$\begin{aligned} u_{j+1} &= u_{j+1/2} - \frac{1}{\alpha}\nabla q_{j+1}, \\ p_{j+1} &= p_j + q_{j+1}. \end{aligned} \tag{6.6}$$

For convenience we consider the iteration with the differential operators of the original equations and not with their discretizations. In order to understand the spectral behavior of the error $(u - u_j,\, p - p_j)$, we start with a pressure p_0 such that $\int_\Omega p_0 dx = 0$ and that the error functions contain only one frequency

$$p_0(x) = p(x) + \cos kx, \quad k \in \mathbb{N}. \tag{6.7}$$

A straightforward integration of (6.4) and (6.5) leads to the intermediate velocity

$$u_{1/2}(x) = u(x) + \frac{k}{\gamma + k^2} \sin kx,$$

the pressure correction

$$q_1(x) = -\frac{\alpha}{\gamma + k^2} \cos kx,$$

and the final result of the iteration step

$$\begin{aligned} u_1(x) &= u(x) = u_R, \\ p_1(x) &= p(x) + \left(1 - \frac{\alpha}{\gamma + k^2}\right) \cos kx. \end{aligned} \tag{6.8}$$

It follows from (6.8) that we have to choose $\alpha = \gamma + 1$ or smaller if an overshooting of the correction is to be avoided. Therefore, the damping factor for $\cos kx$ in (6.8) cannot be better than

$$1 - \frac{\gamma + 1}{\gamma + k^2}. \tag{6.9}$$

In particular, the damping factor is close to 1 for the oscillating terms.

In the case of $\gamma > 0$ the situation is better than for $\gamma = 0$. Since the parameter γ is proportional to $1/\Delta t$, the influence of the frequencies is diminished if the timesteps Δt are small. This is the reason why SIMPLE has been used as a smoother in time dependent problems, if Δt is small (cf. [16]).

We emphasize that our smoothing procedure costs as much as the pressure correction step of the SIMPLE algorithm. Although it is cheaper than a complete SIMPLE cycle, it is better suited as a smoother.

The difference between the SIMPLE type algorithms and the basic iterations in Section 3 is emphasized by the following definition.

6.3 Definition

(1) An iterative method is called *u-dominant* or *direct* if (u_{j+1}, p_{j+1}) mainly depends on the value of u_j.

(2) An iterative method is called *p-dominant* or a *Schur complement iteration* if (u_{j+1}, p_{j+1}) mainly depends on the value of p_j.

In each case the other variable is called a *slave*.

Obviously, the iteration (3.6) is u-dominant, on the other hand the algorithms of SIMPLE type are p-dominant whenever good approximations of (6.1) are computed. Thus the concept of both iterations is different. On the other hand the difference is reduced in actual computations when only approximations of these iterations are performed. This is illustrated e. g. in [15] — The well known Uzawa algorithm is also p-dominant. It may be even interpreted as a Jacobi iteration for the equation for p with the Schur complement $BA^{-1}B^T p = BA^{-1}f - g$.

Let K denote the inverse matrix given in (3.4).

The positive results in section 5 and the negative result in Remark 6.1 indicate the following conjecture. If an iteration of the form (3.6) is used such that the block matrix K_{11} and not K_{22} enters, then only α^{-1} enters into the correction, and the iteration has a smoothing effect. This is found in u-dominant iterations. If, on the other hand, K_{22} enters, a term proportional to α is encountered, and the iteration is not a good smoother. The latter undesired behaviour is often connected with p-dominance.

7. Numerical Results

Numerical results are presented for finite element discretizations of 2D-Stokes problems on a square

$$\begin{aligned} \gamma u - \Delta u + \nabla p &= f \quad && \text{in } \Omega := (0,1) \times (0,1), \\ -\operatorname{div} u &= 0 && \text{in } \Omega, \\ u &= u_R && \text{on } \Gamma, \\ \int_\Omega p \, dx &= 0 \end{aligned} \tag{7.1}$$

with $\gamma \geq 0$ (cf. (6.3)). We have used three different grids for the discretization of Ω, a structured one (see Fig. 1.), an unstructured one (see Fig. 2.), and following [13] a grid which is compressed in one direction, (see Fig. 3). The finest grids are obtained by 4 regular refinements, and one refinement yields the coarset grid.

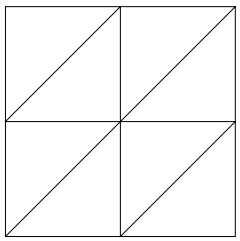

Fig. 1. Structured grid

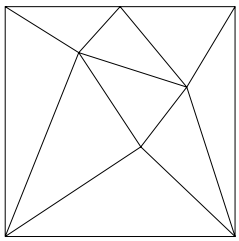

Fig. 2. Unstructured grid

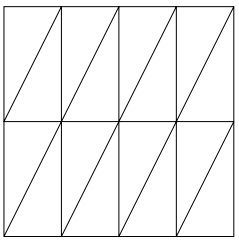

Fig. 3. Distorted grid

The finite element calculations refer to the modified Taylor-Hood-element. Here p_h lives on grids of level ℓ with meshsize $h = h_\ell$ and u_h on grids with meshsize $\frac{h}{2}$. The program is based upon a finite element library, FEAT, cf. [3].

We have investigated the following test problems.
Example 1: A stationary one with

$$\begin{aligned} u(x, y) &= (\sin x \sin y, \cos x \cos y)', \\ p(x, y) &= 2 \cos x \sin y + C, \\ f(x, y) &= (0, 4 \cos x \cos y)', \\ u_R(x, y) &= u|_\Gamma(x, y). \end{aligned}$$

Example 2: A time dependent problem which has already been discretized in time with

$$\begin{aligned} u(x, y) &= (\sin 2\pi xt \sin 2\pi yt, \cos 2\pi xt \cos 2\pi yt)', \\ p(x, y) &= \left(xy - \frac{1}{4}\right) \sin t, \\ f(x, y) &= \gamma u - \Delta u + \nabla p, \quad \gamma = \frac{c}{\Delta t}, c > 0, \\ u_R(x, y) &= u|_\Gamma(x, y). \end{aligned}$$

Throughout this section all errors are measured by the ℓ_2-norm of the residues.

The finite element discretization of (7.1) leads to a system of the form (3.1). The auxiliary linear systems are solved approximately by a cg-solver with accuracies $\epsilon_u = 10^{-4}$ for the error of (6.1) and $\epsilon_q = 10^{-2}$ for the error of the pressure-correction equation. All computations start with $p^0 \equiv 0$ and $u_i^0 = 0$ for the inner nodes and $u_i^0 = u_R(x_i)$ for the nodes x_i on the boundary. Therefore, the error of the initial guess contains both smooth and oscillating parts.

For the sake of comparison we first have used the pure iterations to solve problem (7.1) for example 1 on grids with meshsize $h = 1/32$ for p and $1/64$ for u, resp.. To

Table 1. Results of multigrid algorithm for stationary problem.
W-cycle, 2 pre-smoothing steps, 2 post-smoothing steps.

grid	smoother	C^{-1}	α	rate	steps	final error	CPU
1	SIMPLE	I	const.	0.535	20	$5.7 \cdot 10^{-4}$	185
	BASIC	I	const.	0.120	6	$< 10^{-5}$	42
		I	adapt.	0.105	6	$< 10^{-5}$	46
		SSOR	const.	0.025	4	$< 10^{-5}$	50
		SSOR	adapt.	0.014	4	$< 10^{-5}$	45
2	SIMPLE	I	const.	0.515	20	$1.1 \cdot 10^{-3}$	250
		D^{-1}	const.	0.545	20	$9.8 \cdot 10^{-4}$	248
	BASIC	I	const.	0.477	20	$7.8 \cdot 10^{-4}$	209
		I	adapt.	0.353	14	$< 10^{-5}$	175
		D^{-1}	const.	0.403	19	$< 10^{-5}$	213
		D^{-1}	adapt.	0.250	10	$< 10^{-5}$	133
		SSOR	const.	0.106	7	$< 10^{-5}$	112
		SSOR	adapt.	0.082	6	$< 10^{-5}$	107
3	BASIC	D^{-1}	adapt.	0.173	8	$< 10^{-5}$	194
		SSOR	adapt.	0.033	5	$< 10^{-5}$	165

achieve an error less then 10^{-5} we needed more than 500 iteration steps and 700 CPU-seconds on a workstation with 40 MFLOPS.

To solve the discrete problem on the finest grid we have used a multigrid iteration on four levels with the W-cycle. The computations stopped when the error bound 10^{-5} was achieved, but at most 20 MG-steps were carried out. We have used the BASIC iteration from Section 3 and the SIMPLE iteration from Section 6 as smoothers. For the matrix C we have chosen the identity, or the diagonal part of A, or the preconditioning matrix for A which is usually found in SSOR preconditioner of cg codes (Here it is implemented with relaxation parameter $\omega = 1$). Moreover we have chosen either constant scaling factors α, or α was adapted in each step ($n \geq 1$) such that $\| f - Au^{n+1} - B^T p^{n+1} \|_{\ell_2}$ was minimized. The choices of α are related with the smallest eigenvalue of A in the SIMPLE-case and with the largest one in the BASIC-case on each level. Table 1 shows the results for the different smoothing iterations and for the stationary example with the three grids and 2 pre- and 2 post-smoothing steps. The convergence rates are the geometrical averages over the first 10 MG-steps, and the CPU-times are given in seconds. The parameter α was chosen nearly optimal and differs in each case where α was kept.

Obviously all variants of the multigrid iterations are much faster than the pure iterations. There are also significant differences between the multigrid convergence rates for SIMPLE- and BASIC-type smoothers. This was expected. In the case of the structured grid BASIC-type variants are much better than SIMPLE-type ones. The convergence rates are as good as expected in multigrid algorithms. In the case of the unstructured grid the

situation is more involved. There are great differences between variants of BASIC. Only the SSOR-method yields a high efficiency. Further tests show that the more regular a grid is, the better are the results for $C = I$ or $C = D$ in the BASIC-case.

Table 2. Multigrid iteration for stationary problem on grid 1.
W-cycle, 2 pre-smoothing steps, 2 post-smoothing steps, α = constant.

step	1	2	3	4	5	10
smoother	SIMPLE, $C = I$.					
reduction	0.014	0.451	0.750	0.820	0.843	0.924
error	$6.1 \cdot 10^{-3}$	$2.8 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	$8.3 \cdot 10^{-4}$
smoother	BASIC, $C = I$.					
reduction	0.119	0.117	0.119	0.122	0.126	
error	$3.3 \cdot 10^{-1}$	$3.9 \cdot 10^{-2}$	$4.6 \cdot 10^{-3}$	$5.6 \cdot 10^{-4}$	$7.1 \cdot 10^{-5}$	
smoother	BASIC, $C =$ SSOR.					
reduction	0.008	0.041	0.047	0.053		
error	$9.0 \cdot 10^{-2}$	$3.7 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$	$9.4 \cdot 10^{-6}$		

Table 2 presents the history of a multigrid iteration for the structured grid with the changes of the error reduction factors. In the SIMPLE-case there is a good error reduction only in the first step, but afterwards the error reduction factors become almost as bad as in the single grid SIMPLE iteration. On the other hand the smoothing by the BASIC iteration leads to good and almost constant convergence factors.

The results for different numbers of pre-smoothing and post-smoothing steps are listed in Table 3 for the stationary case and the worst grid. Here we have used the BASIC-iteration with $C^{-1} =$ SSOR(A) and the constant value $\alpha = 1.0$ for smoothing in W-cycle and V-cycle. As expected only 1 pre- and 1 post-smoothing step leads to unsatisfactory convergence rates since the first step in the post-smoothing sequence acts more as a projection than as a smoother. If different numbers in the pre-smoothing and post-smoothing are chosen, then it is more advantageous to choose a larger number in the post-smoothing. This means that the first post-smoothing step only ensured the freedom of divergence. In view of CPU-time there are not great differences between W- and V-cycle, 1 pre- and 2 post-smoothing steps are the best case.

In time dependent problems multigrid iteration with SIMPLE for the smoothing yields good convergence rates, but only if the time steps Δt are small and the condition number of the matrix A is not large. Table 4 shows the results for example 2 on the structured grid. The time discretization is irrelevant for our considerations. Again we took the W-cycle with four levels and 2 pre- and 2 post-smoothing steps.

We conclude that the multigrid algorithms with our smoothing procedure provide good convergence rates both for stationary and for time dependent problems. The deteri-

Table 3. Results of multigrid algorithm for stationary problem on grid 2. Different combinations of pre-smoothing- and post-smoothing steps with the BASIC iteration. $C^{-1} = \text{SSOR}, \alpha = 1.0 = \text{constant}$.

steps		W-cycle			V-cycle		
pre	post	rate	steps	CPU	rate	steps	CPU
2	2	0.144	8	138	0.170	9	134
1	1	0.467	> 20		0.400	> 20	
3	3	0.095	6	162	0.118	7	157
1	2	0.178	10	122	0.212	11	117
2	1	0.213	10	121	0.240	11	122
3	1	0.169	8	144	0.195	9	142
1	3	0.134	9	142	0.196	9	137
2	3	0.106	7	152	0.130	8	146
3	2	0.127	7	155	0.137	7	148

Table 4. Multigrid convergence rate for instationary problem and grid 1. W-cycle, 2 pre-smoothing steps, 2 post-smoothing steps. $C = I, \alpha = \text{constant}, c = 1$.

Δt	smoother	time step				
		1	2	3	4	5
0.0001	SIMPLE	0.062	0.062			
0.001		0.232	0.229	0.620	0.633	0.633
0.005		0.382	0.359	0.786	0.816	0.854
0.001	BASIC	0.104	0.016	0.062	0.063	0.064
0.01		0.122	0.112	0.113	0.116	0.116
0.1		0.125	0.122	0.125	0.124	0.124

oration of the multigrid convergence rate which is observed for distorted or compressed grids, can be compensated by an SSOR preconditioning in the smoothing iteration.

Acknowledgement. The authors want to thank Prof. Heribert Blum for his support in the use of the FEM package FEAT.

References

[1] Bank, R.E. and T. Dupont (1981) An optimal order process for solving finite elements equations. Math. Comp. 36, 35–51

[2] Bank, R.E., Welfert, B. and Yserentant, H. (1990) A Class of Iterative Methods for Solving Saddle Point Problems. Numer. Math. 56, 645-666

[3] Blum, H., J. Harig, S. Müller, and S. Turek (1992) Feat 2D, Finite Element Analysis Tools, User Manual, Release 1.3, Preprint University Heidelberg
[4] Braess, D. and W. Hackbusch (1983) A new convergence proof for the multigrid method including the V-cycle. SIAM J. Numer. Anal. 20, 967–975
[5] Brandt, A. and N. Dinar (1979) Multigrid solutions to flow problems. In "Numerical Methods for Partial Differential Equation", S. Parter (ed.) Academic Press, New York, 53–147
[6] Gireault,V. and P.A. Raviart (1986) Finite Element Methods for Navier-Stokes Equations. Springer, Berlin–Heidelberg
[7] Hackbusch, W. (1980) Analysis and multigrid solutions of mixed finite element and mixed difference equations. Report, University Bochum
[8] Hackbusch, W. (1985) Multi-grid Methods and Applications. Springer, Berlin–Heidelberg
[9] Patankar, S.V. (1980) Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York–Cambridge
[10] Patankar, S.V. and D.B. Spalding (1972) A calculation procedure for heat and mass transfer in three-dimensional parabolic flows. Int. J. Heat Mass Transfer 15, 1787–1806
[11] Peric, M., R. Kessler, and G. Scheuerer (1988) Comparison of finite volume numerical methods with staggered and colocated grids. Comput. Fluids 16, 389–403
[12] Rannacher, R. (1992) On Chorin's projection method for the incompressible Navier-Stokes equations. In "Navier-Stokes Equations: Theory and Numerical Methods" (R: Rautmann et al, eds.) Springer, Berlin–Heidelberg
[13] Reichert, H. and G. Wittum (1994) Robust multigrid methods for the incompressible Navier-Stokes equations. In "Fast Solvers for Flow Problems", (W. Hackbusch et al., eds). Notes Numer. Fluid Mech. 49, 216–228 (1995). Vieweg, Wiesbaden.
[14] Shaidurov, V.V. (1989) Multigrid Methods for Finite Elements. Kluwer Academic Publishers, Dordrecht
[15] Shaw, G.J. and Sivaloganathan, S. (1988) On the Smoothing Properties of the SIMPLE Pressure-Correction Algorithm. Int. J. Numer. Meth. Fluids 8, 441–461
[16] Turek, S. (1994) On discrete projection methods for the incompressible Navier-Stokes equations: An algorithmical approach. Preprint, University Heidelberg
[17] VanDoormaal, J. P. and G. D. Raithby (1984) Enhancements of the SIMPLE method for predicting incompressible fluid flows. Numerical Heat Transfer 7, 147–163
[18] Verfürth, R. (1984) A multilevel algorithm for mixed problems. SIAM J. Numer. Anal. 21, 264–271
[19] Wesseling, P. (1992) An Introduction to Multigrid Methods. John Wiley & Sons, Chichester–New York
[20] Wittum, G. (1989) Multi-grid methods for Stokes and Navier-Stokes equations with transforming smoothers: Algorithms and numerical results. Numer. Math. 54, 543–563
[21] Wittum, G. (1990) On the convergence of multi-grid methods with transforming smoothers. Numer. Math. 57, 15–38

Notes added after publication

1. The intermediate pressure step in the Multigrid Algorithm (Section 4) can be abandoned. It has been proven in [22] that (3.12) holds for the actual pressure vector and not only for an *appropriate* one. For details we refer to Section 5 in [22].

2. Another modification of the Multigrid Algorithm is suggested which increases its numerical stability. The smoothing iteration is usually performed with the residue in the momentum equation being large compared to the residue in the continuity equation. Therefore a relaxation factor $\alpha \approx \lambda_{\max}$ makes sense. Note that the different influence of the parameter α on the u-component and the p-component is obvious from (3.4). Thus the choice is not good if the actual velocity does not satisfy the discrete continuity equation.

For this reason we rather suppress the update of p in the first step of the smoothing step and choose $p_\ell^1 = p_\ell^0$ if

$$Bu_\ell^0 \neq 0.$$

Similarly we have $Bu_\ell^{m+2} \neq 0$, i.e., the velocity is not in the kernel after the coarse-grid correction. Therefore we better set $p_\ell^{m+3} = p_\ell^{m+2}$ or we multiply the correction of the p-variable by a relaxation factor such that the residue

$$\| f - Au_\ell^{m+3} - B^T p_\ell^{m+3} \|_{\ell_2}$$

is minimal. This is done in the same way as the computation of adaptive factors described in section 7. With this modification the computation of adaptive factors in the other smoothing steps becomes less important.

[22] D. Braess, W. Dahmen, and C. Wieners, A multigrid algorithm for the mortar finite element method. SIAM J. Numer. Anal. (to appear)