Comput Visual Sci 1: 193–200 (1999)

Computing and
Visualization in Science

Regular article

A comparison of parallel solvers for the incompressible Navier–Stokes equations

V. John

Institut für Analysis und Numerik, Otto-von-Guericke-Universität Magdeburg, Postfach 4120, D-39016 Magdeburg, Germany
(e-mail: volker.john@mathematik.uni-magdeburg.de)

Received: 14 October 1997 / Accepted: 11 February 1998

Communicated by: G. Wittum

Abstract. The paper compares coupled multigrid methods and pressure Schur complement schemes (operator splitting schemes) for the solution of the steady state and time dependent incompressible Navier–Stokes equations. We consider pressure Schur complement schemes with multigrid as well as single grid methods for the solution of the Schur complement problem for the pressure. The numerical tests have been carried out on benchmark problems using a MIMD parallel computer. They show the superiority of the coupled multigrid methods for the considered class of problems.

1 Introduction

The fast and accurate solution of the steady state and time dependent incompressible Navier–Stokes equations is of interest not only as an autonomous problem. The numerical simulation of complex models coming from engineering and industry, such as crystal growth, the study of ferro fluids, and contamination transport in groundwater and estuaries, requires the repeated solution of Navier–Stokes equations. Besides a suitable discretization, a fast and robust solver for these equations is the core of these numerical simulations.

A number of solvers for the Navier–Stokes equations has been developed during the last decades. We consider solution schemes where the nonlinear system of equations is solved by successive solutions of linear systems. The solvers for these linear systems, coming from the linearization and discretization of the Navier–Stokes equations, can be divided in two major groups. The solvers of the first group try to attack the coupled linear system of equations. This group can be subdivided further in multilevel type methods and single grid methods. The second group of solvers splits the operator by eliminating the velocity in the coupled linear system and essentially solves an equation for the pressure. We call these solvers *pressure Schur complement schemes*. Well known representatives are the SIMPLE–algorithm by Patankar and Spalding [10] and the Uzawa-algorithm. A numerical study of pressure Schur complement schemes can be found in a paper by Turek [16].

In the DFG-high-priority program "Flow simulation with high performance computers", a comprehensive study and comparison of discretizations and solvers for the Navier–Stokes equations have been carried out, see Schäfer and Turek [12]. Solvers which use multigrid methods in their essential parts have been proven superior to single grid methods. This will be demonstrated also in the numerical results of this paper. The main goal of this paper is a comparison of (strongly) coupled multigrid methods and pressure Schur complement schemes where multigrid methods are used for solving the systems concerning the degrees of freedom of the velocity as well as of the pressure. In this sense, we can call the latter "weakly coupled multigrid methods".

The huge amount of data, the large size of the arising discrete systems, and the fine resolution of relatively long time intervals require the use of powerful parallel computers. Unfortunately, methods which have a good numerical performance, e.g. multigrid methods in many applications, often show an unsatisfactory parallel efficiency. In Sect. 5, we describe some simple but efficient ways to balance the numerical and parallel performance which result in a considerable increase of the total efficiency, i.e. savings of computing time. We present results computed on a MIMD parallel computer with a moderate number of processors. We think that in the following years this type of computer will be quite common in industry and science.

The paper is organized as follows: in Sect. 2, we give details on the spatial and temporal discretization of the equations. Sects. 3 and 4 are devoted to the description of the coupled multigrid methods and the pressure Schur complement schemes. Parallel aspects of multigrid methods are discussed in Sect. 5. The main part of the paper is Sect. 6, where the numerical studies on a benchmark problem describing the flow through a channel around a cylinder are presented. The results of these studies are summarized in Sect. 7.

2 The problems and their discretization

We consider the steady state incompressible Navier–Stokes equations

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } \Omega \\ \mathbf{u} &= \mathbf{g} && \text{on } \partial\Omega \end{aligned} \tag{1}$$

and the time dependent incompressible Navier–Stokes equations

$$\begin{aligned} \frac{\partial\mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega\times(0,T] \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } \Omega\times(0,T] \\ \mathbf{u} &= \mathbf{g} && \text{on } \partial\Omega\times(0,T] \\ \mathbf{u} &= \mathbf{u}_0 && \text{for } t = 0. \end{aligned} \tag{2}$$

In (1) and (2), Ω is a bounded domain in $\mathbb{R}^2$, $\mathbf{u}$ the velocity, p the pressure, ν the kinematic viscosity of the fluid, $\mathbf{u}_0$ an initial velocity, $\mathbf{g}$ a Dirichlet boundary condition satisfying the compatibility condition

$$\int_{\partial\Omega} \mathbf{g}\cdot n\, d\gamma = 0,$$

T the end of a time interval, and $\mathbf{f}$ represents exterior forces. The first equation in (1) and (2) describes the conservation of momentum and the second equation the conservation of mass.

Equation (1) is linearized using a fixed point iteration. Given a current iterate $(\mathbf{u}^n, p^n)$, in each nonlinear iteration step an Oseen problem of the form

$$\begin{aligned} -\nu\Delta\mathbf{u}^{n+1} + (\mathbf{u}^n\cdot\nabla)\mathbf{u}^{n+1} + \nabla p^{n+1} &= \mathbf{f} && \text{in } \Omega \\ \nabla\cdot\mathbf{u}^{n+1} &= 0 && \text{in } \Omega \\ \mathbf{u}^{n+1} &= \mathbf{g} && \text{on } \partial\Omega \end{aligned}$$

is solved which is discretized using the nonconforming P_1/P_0-finite element from Crouzeix and Raviart [4]. Let $\mathcal{T}_h$ be an admissible triangulation of Ω into triangles. Then the discrete velocity is computed in the space V_h with

$$V_h := \left\{ \begin{array}{l} \text{space of 2d linear functions on the triangles} \\ \text{which are continuous in the midpoints of the} \\ \text{edges of the triangles} \end{array} \right\}$$

and the discrete pressure is computed in the space Q_h with

$$Q_h := \left\{ \begin{array}{l} \text{space of constant functions on the triangles} \\ \text{which belong to } L^2_0(\Omega) \end{array} \right\}.$$

This pair of finite element spaces guarantees the inf-sup stability condition uniformly with respect to the mesh size and the shape regularity constant of the mesh. Besides this favourable analytical property, there are also advantages from the point of view of an implementation on a parallel computer. It can be seen in Fig. 1 that the degrees of freedom of the velocity can be connected to the edges of the triangles. Thus, the degrees of freedom on the interfaces (boundary of a subdomain of Ω which is stored on a processor) have to be stored on two processors only. That's why, only one communication is sufficient to interchange information between the same degrees of freedom on different processors. This is not true for degrees of freedom which are connected to corners of triangles. In this case, there exist some unknowns which are stored on more than two processors, so-called *crosspoints*.

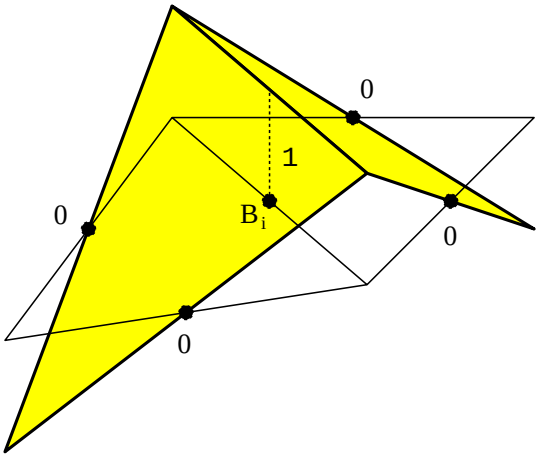

Fig. 1. Scalar component of a basis function of V_h

In the case of dominant convection, a stabilization of the discrete problems is necessary. We use Samarskij upwinding which has been analyzed by Schieweck and Tobiska [15]. Another technique that may improve the accuracy of the discrete solution is the so-called *pressure separation* which has been introduced by Schieweck [14] and Dorok [5]. Instead of (1), we solve

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla(p-\tilde{p}) &= \mathbf{f} - \nabla\tilde{p} && \text{in } \Omega \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } \Omega \\ \mathbf{u} &= \mathbf{g} && \text{on } \partial\Omega, \end{aligned} \tag{3}$$

where $\tilde{p} \in H^1(\Omega)$ is chosen such that the H^1-norm $\|p-\tilde{p}\|_1$ is less than $\|p\|_1$. This technique leads to a better a priori error estimate for the solution of the discrete system obtained with (3) in comparison to that coming from (1). There are several possibilities to choose $\tilde{p}$, for a discussion see [6]. In the numerical studies of this paper, we have applied the projection operator proposed by Oswald [9] to map the solution $p_h \in Q_h$ of the discrete problem without pressure separation to a piecewise linear conforming function $\tilde{p}$. This function is used as separated pressure.

The time derivative in (2) is discretized using the Crank–Nicolson scheme with a modification, which is employed e.g. by Turek [16]. The velocity and the external forces of the previous time step but not the pressure are used to compute the right hand side in this scheme. In the time t_{k+1}, this leads to a system of the form

$$\begin{aligned} \frac{\mathbf{u}-\mathbf{u}(t_k)}{\tau} + \frac{1}{2}\left(-\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p & \\ = \frac{1}{2}\mathbf{f}(t_{k+1}) + \frac{1}{2}\mathbf{f}_1(t_k) && \text{in } \Omega \\ \nabla\cdot\mathbf{u} = 0 && \text{in } \Omega \\ \mathbf{u} = \mathbf{g}(t_{k+1}) && \text{on } \partial\Omega \end{aligned}$$

with $\mathbf{u} = \mathbf{u}(t_{k+1})$, $p = p(t_{k+1})$, $\tau = t_{k+1} - t_k$, $\mathbf{f}_1(t_k) = \nu\Delta\mathbf{u}(t_k) - (\mathbf{u}(t_k)\cdot\nabla)\mathbf{u}(t_k) + \mathbf{f}(t_k)$. The linearization and discretization in space is carried out analogously to the steady

state problem including the usage of Samarskij upwinding and pressure separation. The modified Crank–Nicolson scheme has produced more accurate results than the BDF(2) scheme in a comparative study, see [6].

Thus, the linearization and discretization of the incompressible Navier–Stokes equations leads to large saddle point problems of the form

$$\mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} r \\ s \end{pmatrix}. \tag{4}$$

3 Coupled multigrid methods

Coupled multigrid methods compute the solution for both types of unknowns (velocity and pressure) of (4) simultaneously. Here, we want to give some details on the components of the multigrid methods we have used, which are the grid transfer operations, the smoothers, the coarsest grid solver, and a step length control.

The grid transfer operations. There are no canonical grid transfer operations since the discrete spaces belonging to the different levels of the multigrid are not nested in the nonconforming finite element discretization. We use L^2-projections for the restriction and prolongation.

The smoothers. The smoother influences the efficiency of multigrid methods considerably. Because of the zero block of $\mathcal{A}$, standard smoothers, like point-wise Gauss–Seidel, cannot be applied to (4) without modification. We use the *Vanka smoother*, see e.g. Vanka [17] or Schieweck [13], which is a block-wise Gauss–Seidel method. A block consists of the degrees of freedom which are connected to a triangle T. That are three two-dimensional vectors of the discrete velocity and one degree of freedom of the pressure. Using the Vanka-smoother, local (7×7)-systems with a matrix of the form

$$\begin{pmatrix} \mathrm{diag}(A_T) & B_T \\ B_T^T & 0 \end{pmatrix}$$

have to be solved. Schieweck [14] pointed out that for the reason of stability it is sometimes better to use a full upper left block in the (7×7)-systems

$$\begin{pmatrix} A_T & B_T \\ B_T^T & 0 \end{pmatrix}.$$

We call this smoother *stabilized Vanka smoother*. The solution of its local (7×7)-system needs about three times the numerical work (floating point operations) compared to the Vanka smoother.

On a parallel computer, communications within the smoothing process should be avoided. That's why, we use a block Jacobi smoother where a block consists of all unknowns which are stored on one processor. In the blocks, the Vanka smoother or stabilized Vanka smoother is used. After each smoothing iteration, a communication is necessary in order to obtain a consistently stored result via averaging the values.

The coarsest grid solvers. The linear system on the coarsest grid often is too large to solve it efficiently with a direct method. That's why, we use an iterative method, namely the same iterative method which is used as smoother on the finer levels. A *standard approach* is to reduce the Euclidean norm of the residual by a prescribed factor. However, this approach has some disadvantages on parallel computers, compare Sect. 5. An *alternative approach* is to prescribe a small number of iteration steps and break the iteration after these steps without computing the norm of the residual.

The step length control. In addition to standard multigrid methods, we apply a step length control after each multigrid cycle. Let $\left(u^T, p^T\right)^i$ be the current iterate and $\left(\delta u^T, \delta p^T\right)^{i+1}$ the update proposed by the multigrid method. This update is accepted only as a direction of update and the next iterate is set to be

$$\left(u^T, p^T\right)^{i+1} = \left(u^T, p^T\right)^i + \kappa \left(\delta u^T, \delta p^T\right)^{i+1}.$$

The factor $\kappa \in \mathbb{R}$ is chosen such that the Euclidean norm of the residual

$$\left[\left(\begin{pmatrix} r \\ s \end{pmatrix} - \mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix}^{i+1}\right)^T \left(\begin{pmatrix} r \\ s \end{pmatrix} - \mathcal{A}\begin{pmatrix} u \\ p \end{pmatrix}^{i+1}\right)\right]^{1/2}$$

is minimal. This one-dimensional optimization problem has the solution

$$\kappa = \frac{\left((r^T, s^T) - \mathcal{A}(u^T, p^T)^i\right)^T \left(\mathcal{A}(\delta u^T, \delta p^T)^{i+1}\right)}{\left(\mathcal{A}(\delta u^T, \delta p^T)^{i+1}\right)^T \left(\mathcal{A}(\delta u^T, \delta p^T)^{i+1}\right)}. \tag{5}$$

Numerical tests [6] show the important rôle of the step length control to improve the efficiency of all multigrid methods.

4 Pressure Schur complement schemes

Pressure Schur complement schemes start with eliminating u by means of the momentum equation in (4). The arising equation for p, the so-called *Schur complement equation*, is solved by an iteration of the form

$$\begin{aligned} p^0 &= 0, \\ p^{m+1} &= p^m + \alpha\left(B^T C^{-1} B\right)^{-1}\left(-B^T A^{-1} B p^m + B^T A^{-1} r - s\right), \end{aligned} \tag{6}$$

$m = 0, 1, \ldots, \mathsf{M}-1$. Here, C^{-1} is a matrix approximating A^{-1} and α is a damping factor. A linear system of the form

$$B^T C^{-1} B x = -B^T A^{-1} B p^m + B^T A^{-1} r - s \tag{7}$$

has to be solved in each iteration step. Having finished the iteration with step $m = \mathsf{M}-1$, an approximation u^{M} of u can be computed with

$$u^{\mathsf{M}} = A^{-1}\left(r - B p^{\mathsf{M}}\right) \tag{8}$$

or

$$u^{\mathsf{M}} = A^{-1}\left(r - B p^{\mathsf{M}-1}\right) + \frac{1}{\alpha} C^{-1} B\left(p^{\mathsf{M}} - p^{\mathsf{M}-1}\right). \tag{9}$$

The choice of (8) puts the emphasis on the fulfillment of the momentum equation whereas (9) guarantees a discrete divergence free solution, see Turek [16]. In (9), no inversion of A

is necessary since $A^{-1}(r - Bp^{M-1})$ has been computed in the last step of iteration (6). The main cores of the pressure Schur complement schemes are the solution of system (7) in each iteration step and the approximation of A^{-1} in (6)–(8).

To approximate the product $A^{-1}b$, we apply a multigrid method for solving a linear system for the velocity degrees of freedom of the form $Au = b$. As restriction and prolongation, we have chosen L^2-projections. As smoother, we use ILU$_\beta(0)$-iterations, see e.g. Wittum [18]. This smoother is also employed to solve the system on the coarsest grid approximately. After each multigrid cycle, we apply a step length control of form (5), now for the system $Au = b$.

To solve (7) efficiently, we must distinguish between the steady state and the time dependent case. In the time invariant case, the matrix A has the form

$$A = \nu L + K,$$

where L is the discretization of the Laplacian and K the Samarskij upwind discretization of the linearized convective term. Thus, A generally does not have a dominant diagonal and is nonsymmetric. The choice of C^{-1} has to take into account these properties. The solution of (7) requires products of the form $C^{-1}b$. These products are defined by the approximation of the solution of $Au = b$ which is obtained with a multigrid method for the velocity degrees of freedom. The in general nonsymmetric linear system (7) is solved approximately by *gmres(restart)* from Saad and Schultz [11]. In our computations, we have used the same multigrid scheme to approximate $A^{-1}b$ and to compute $C^{-1}b$, i.e. we solve the pressure Schur complement system with an unpreconditioned gmres method.

For the time dependent problem, especially for small time steps, the discretization of the time derivative leads to a diagonal dominance in A, since

$$A = \tau^{-1}M + \nu L + K.$$

Therefore, $C^{-1} = I$ (identity matrix), $C^{-1} = M^{-1}$ (mass matrix, which is diagonal due to the L^2-orthogonality of the standard basis functions of V_h) or $C^{-1} = \mathrm{diag}(A)^{-1}$ are good approximations of A^{-1}. The linear system (7) becomes symmetric and positive definite after fixing the constant in the pressure. It is solved either by a *preconditioned conjugate gradient method (pcg)* or by a multigrid method for the pressure degrees of freedom. The matrix $B^T C^{-1} B$ can be computed and stored explicitely if C^{-1} is a diagonal matrix, see [6]. Thus, standard multigrid techniques can be applied, even for piecewise constant functions. We use Gauss–Seidel iterations as smoother and as iterative solver on the coarsest grid, L^2-projections as grid transfer operations, and a step length control of form (5), now for the linear system (7).

5 Parallel aspects of multigrid methods

Multigrid methods show in many situations a good numerical efficiency which is based on global transport of information. However, this *global* transport contradicts the optimal condition of high parallel efficiency, which is locality.

Especially, *pure multigrid methods* (= one smoothing step on each level and exact solution of the coarsest grid system) show a bad parallel performance on all grids which are coarser than the finest one. The numerical work (flops) on these grids is in general small compared to the amount of communications. Even idleness of processors will occur. The situation is worst on the coarsest grid, where the global transport of information takes place with the solution of the coarsest grid system. If the V-cycle is used, the losses of parallel and total efficiency (computing time) are less dramatic, provided the problems are large (with respect to the number of processors) and the number of levels is moderate (≤ 10), e.g. see McBryan et al. [8]. But employing the W-cycle, for the reasons of numerical efficiency and robustness, can lead to a considerable loss of parallel and total efficiency on parallel computers. Here, it is necessary to increase the parallel efficiency in order to obtain a better total performance. However, the methods of increasing the parallel efficiency may decrease the numerical efficiency. Thus, there is the need of a balance of numerical and parallel efficiency. This balance should preserve the core of the multigrid algorithm. It should strengthen the multigrid components which show a good parallel performance (these are of single grid character) and weaken the components which are responsible for the bad parallel efficiency.

Concerning the last point, Axelsson and Neytcheva [1] have introduced the *short multigrid cycle*, where for deeply refined grids the level of the coarsest grid is determined as a function of the level of the finest grid. Thus, the short multigrid cycle has a reduced depth compared to the pure multigrid cycle (which can be interpreted as a compromise between pure multigrid and single grid). The coarsest grid in the short multigrid cycle possesses enough degrees of freedom such that the ratio of numerical work to communications is much better than on the original coarsest grid. In addition, idleness of processors is avoided.

We have proposed some methods to increase the parallel as well as the total efficiency in [7]. On less deep refinements of a multigrid, the work on the coarsest grid should be small. Often, the coarsest grid system is solved iteratively. A widely used approach, which works in general well on workstations, is the iteration up to a prescribed reduction of the norm of the residual (e.g. 0.1). However, the computation of the norm requires global communication on parallel computers, which is expensive. Therefore, the coarsest grid system should be solved approximately using a prescribed small maximal number of iterations (e.g. 10 using the smoother) without computing the norm of the residual. The number of cycles for the solution of the linear system will slightly increase in general. But the savings on the coarsest grid are often so large that the computing times decrease considerably. A second way to improve the parallel efficiency consists in increasing the number of smoothing steps, especially on finer grids, which gives a better ratio of numerical work to communications. This approach can be viewed again as a compromise between pure multigrid and single grid methods.

An improvement of the efficiency of multigrid methods, which is possible for all types of cycles and on workstations as well as on parallel computers, results from the current hardware architecture. It is based on the fact that a large number of flops can be applied extremely fast on data which are stored in the cache memory. For multigrid methods, this means to prefer smoothers which perform a lot of flops on a small set of data (small = fits into the cache) instead of

smoothers which perform only a small amount of flops. Of course, the more expensive smoothers should enhance the numerical efficiency and the robustness of the multigrid method in comparison to the cheaper ones.

Specifically, for this reason we use the stabilized Vanka smoother instead of the Vanka smoother in the coupled multigrid methods. The number of flops in each smoothing step increases by a factor of three. However, we have observed an increase of the computing time per smoothing step only by factor of approximately 1.5; for a comparison of both smoothers see also [6, 7].

In Table 2, we present results which demonstrate the improvements of the total efficiency on a parallel computer using the modifications described above. On the one hand, we have used a coupled multigrid method with Vanka smoother, 2 pre and post smoothing steps, and the standard approach of the coarsest grid iteration with a reduction factor of 0.1. On the other hand, the computations have been carried out with the stabilized Vanka smoother, 6 pre and post smoothing steps, and the alternative approach on the coarsest grid with 10 iterations. On the workstation, both schemes need almost the same computing time for the solution of the problem whereas on 16 processors the second scheme is faster by a factor of 2.6. The amount of communications and the time spent on the coarsest grid are considerably smaller in the second scheme.

The modifications described above can be interpreted as a compromise between pure multigrid and single grid, but the resulting algorithm is still far closer to the pure multigrid method than to a single grid scheme.

6 Numerical results

The test problems

The numerical studies have been performed on benchmark problems of the DFG priority research program "Flow simulation with high-performance computers", see Schäfer and Turek [12]. These problems describe flows in a channel around a cylinder, see Fig. 2. The steady state problem is to find a solution of (1) with $\nu = 10^{-3}$, $f = \mathbf{0}$, the parabolic

Table 1. Data of the grid using uniform refinement

level	velocity	pressure	edges on S
0	532	192	8
4	146 752	49 152	128
5	588 416	196 608	256

Table 2. Steady state problem, standard and modified coupled multigrid methods, level 5, above 1 processor, below 16 processors

solver	M	sm.	it.	time	comm.	lev0.
Vanka-W	10	2	11/103	7270	0.0	4.5
stVan-W	10	6	5/32	7190	0.0	0.3
Vanka-W	10	2	13/123	4091	20.0	48.0
stVan-W	10	6	5/33	1575	4.4	5.2

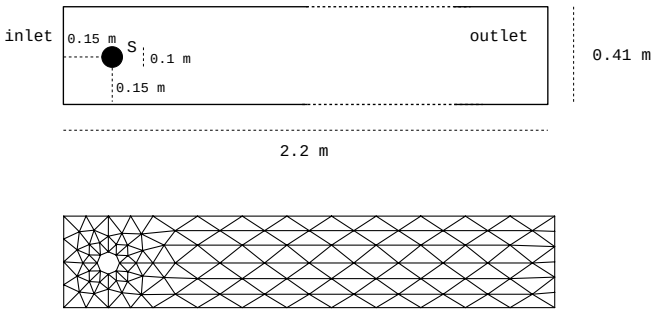

Fig. 2. Domain and coarsest grid (level 0) for the numerical tests

inflow and outflow profile

$$\mathbf{u}(0, y) = \mathbf{u}(2.2, y) = 0.41^{-2}(1.2y(0.41 - y), 0),$$

$0 \le y \le 0.41$ and no-slip conditions on the other boundaries. The Reynold's number of this flow is $\mathrm{Re} = 20$. Benchmark reference values are the drag coefficient c_D, the lift coefficient c_L, and the difference of the pressure between the front and the back of the cylinder Δp.

The time dependent problem (2) describes a Kármán vortex street by $\nu = 10^{-3}$, $\mathbf{f} = \mathbf{0}$, the steady state parabolic inflow and outflow profile

$$\mathbf{u}(0, y, t) = \mathbf{u}(2.2, y, t) = 0.41^{-2}(6y(0.41 - y), 0),$$

$0 \le y \le 0.41$, $0 \le t \le T$ and no-slip conditions on the other boundaries. The benchmark reference values are $c_D(t)$, $c_L(t)$ and $\Delta p(t)$. The initial time corresponds to a state where c_L has a maximum. The length of a period is denoted by T. We found in the numerical tests $T \approx 0.335$ s. Additional benchmark coefficients are the maximal values of $c_D(t)$ and $c_L(t)$, $\Delta p(t_0 + T/2)$, and the Strouhal number St. This flow has the Reynold's number $\mathrm{Re} = 100$.

We have used the coarsest grid (level 0) depicted in Fig. 2. The data of the grids which result from uniform refinement can be seen in Table 1. In [6, 7], we have given results concerning the benchmark reference values using the nonconforming P_1/P_0-finite element discretization on this grid. The low order of the local polynomials of the discrete spaces makes it necessary to use level 4 or more often even level 5 to obtain good results for all reference values.

Next, we explain the abbreviations in the tables given below:

Vanka-V(F,W)	coupled multigrid method with V-cycle, (F-cycle, W-cycle), Vanka smoother,
stVan-V(F,W)	coupled multigrid method with V-cycle, (F-cycle, W-cycle), stabilized Vanka smoother,
pr-cor gm	pressure Schur complement scheme with gmres(restart) for the solution of (7),
pr-cor mg	pressure Schur complement scheme with multigrid for the solution of (7),
pr-cor pcg	pressure Schur complement scheme with pcg for the solution of (7),
M	maximal number of multigrid cycles or iterations (6) per nonlinear iteration step,
sm.	number of pre and post smoothing steps in the multigrid cycles,

it. number of nonlinear iterations/number of multigrid cycles,
time cpu time in seconds,
comm. communication in the cpu time in %,
lev0. iteration on the coarsest grid in the cpu time in %,
τ fixed length of the time step in the time dependent problem,
t/per. computing time for one period in the time dependent problem,
t/step average computing time for one time step in the time dependent problem.

The number of pre- and post-smoothing steps has been chosen equal on all levels.

The computations have been performed on a Parsytec GCPowerPlus (80 MHz, 9.2 MFlops/processor (LINPACK), 35 MB/s communication, 5 μs message setup time, 60 μs minimal network latency). The implementation of the algorithms has been done using parts of the program package *ugp1.0* by Bastian and co-workers [2, 3]. In particular, data structures, load balancing routines and the parallel environment of this program have been used. However, the data structures had to be extended in order to handle multidimensional data and nonconforming finite elements.

The only criterion for the comparison of the different solvers is the total computing time!

The steady state case

At first, we consider the stationary problem. We compare the coupled multigrid methods (V-, F-, and W-cycle) with a pressure Schur complement scheme. We have done comprehensive studies on this subject, see [6, 7]. The results presented here are representative ones. The computations have been carried out on level 5 using 16 processors. The starting point of the iterations has been the interpolated solution from level 4 and the stopping criterion has been chosen as an Euclidean norm of the residual less than 10^{-7}.

The Schur complement system (7) is solved approximately with 10 gmres iterations without restart. The inverse of the matrix A is approximated by one W-cycle.

The components of the coupled multigrid methods have been chosen accordingly to the considerations in the previous section. Thus, we have used the stabilized Vanka smoother, the alternative approach of the coarsest grid iteration, and a large number of smoothing steps has been applied, see Table 3. The maximal number of iterations on the coarsest grid has been prescribed by 10.

Table 3 shows clearly the superiority of the coupled multigrid methods over the pressure Schur complement scheme. Using gmres(restart), there are no problems to damp the high frequent modes of the error. However, this method almost stagnates in the damping of the low frequent error modes which leads to a very slow rate of convergence and a large number of iterations. Using multigrid for solving (7) instead of gmres(restart), the iteration for the solution of the nonlinear system did not converge. Within the coupled multigrid methods, the W–cycle has been proven best for the considered example. Using the modifications described in the previous section, the W-cycle has an acceptable communication overhead and computing time for the solution of the coarsest grid systems. The parallel performance of the V- and F-cycle is very good which comes essentially from the absence of crosspoints in the nonconforming P_1/P_0-finite element discretization.

Table 3. Steady state problem, coupled multigrid methods and pressure Schur complement schemes

solver	M	sm.	it.	time	comm.	lev0.
stVan-V	15	2	28/407	4730	2.4	1.3
stVan-V	15	4	14/200	4066	2.2	0.8
stVan-V	15	6	10/133	3859	2.1	0.5
stVan-V	15	8	8/107	4034	2.0	0.4
stVan-V	15	10	7/79	3673	2.1	0.3
stVan-V	15	12	6/71	3916	2.0	0.3
stVan-V	15	14	6/66	4211	1.9	0.2
stVan-F	10	2	15/144	2356	3.2	4.7
stVan-F	10	4	10/86	2463	2.8	2.7
stVan-F	10	6	6/50	2046	2.6	1.9
stVan-F	10	8	5/41	2178	2.5	1.4
stVan-F	10	10	5/38	2485	2.5	1.2
stVan-W	10	2	11/101	2016	5.4	12.4
stVan-W	10	4	6/51	1722	4.7	7.3
stVan-W	10	6	5/33	1575	4.4	5.2
stVan-W	10	8	4/28	1722	4.3	4.0
stVan-W	10	10	4/27	2039	4.2	3.3
pr-cor gm	10	1	222	14 260	15.2	10.2
pr-cor gm	10	2	199	17 840	15.3	4.1
pr-cor gm	10	3	178	21 050	15.2	2.2
pr-cor gm	10	4	154	22 310	15.5	1.5

The time dependent case

The results given in Table 5 and Table 6 are computed without using pressure separation. The use of pressure separation has improved the accuracy of the computed benchmark reference values but also has led to an increase of the computing times by a factor of 2 to 3. The stopping criterion of the iteration in each time step has been chosen such that the Euclidean norm of the residual is less than 10^{-5}. The maximal number of iterations per time step has been prescribed by 30. The numbers in Table 5 and Table 6 are given in the form "from – to".

In the pressure Schur complement schemes, we have chosen $C^{-1} = \mathrm{diag}\,(A)^{-1}$, the diagonal preconditioner $\mathrm{diag}(B^T C^{-1} B)^{-1}$ for pcg, and the matrix A^{-1} has been approximated by one V-cycle. These schemes have been very sensitive on the choice of the damping parameter α. A good choice depends not only on the length τ of the time step but also on the refinement in space. Even small increases of α have led to a divergence of the method whereas small decreases resulted in a much slower rate of convergence. The parameters we have chosen are given in Table 4.

The pressure Schur complement scheme using 50 steps of pcg for the solution of (7) reaches the prescribed norm

Table 4. Choice of the damping factor α for the pressure Schur complement schemes

τ	0.01	0.005	0.0025
Table 5, level 4	0.15	0.25	0.4
Table 6, level 5	0.05	0.1	0.2

Table 5. Time dependent problem, coupled multigrid methods and pressure Schur complement schemes, time steps $\tau = 0.01, 0.005, 0.0025$ (top to bottom), level 4, 8 processors

solver	it.	t/per.	t/step	comm.	lev0.
Vanka-V	8	14 100	382	4-6	3-6
Vanka-F	6	10 400	281	4.4	3.7
Vanka-W	5-6	9010	244	6.3	8.2
pr-cor mg (8)	8	21 200	574	9-13	3-7
pr-cor mg (9)	8	20 200	546	8-12	3-7
pr-cor pcg (8)	30	44 000	1200	7.0	0.2
Vanka-V	4-5	14 700	204	4-6	2-6
Vanka-F	3-4	11 600	161	4.6	3.7
Vanka-W	3-4	10 500	146	6.4	8.2
pr-cor mg (8)	4	21 100	289	8-12	3-8
pr-cor mg (9)	4	19 700	274	8-13	3-8
pr-cor pcg (8)	30	84 800	1178	6.9	0.2
Vanka-V	3	16 700	117	5.0	3-5
Vanka-F	3	16 400	114	4.4	3.6
Vanka-W	3	14 600	102	5.9	8.1
pr-cor mg (8)	3	25 800	180	9-13	3-8
pr-cor mg (9)	3	24 600	172	9-14	3-8
pr-cor pcg (8)	16-30	144 900	1013	5.9	0.1

Table 6. Time dependent problem, coupled multigrid methods and pressure Schur complement schemes, time steps $\tau = 0.01, 0.005, 0.0025$ (top to bottom), level 5, 16 processors

solver	it.	t/per.	t/step	comm.	lev0.
Vanka-V	12-17	48 900	1359	2.9	1.1
Vanka-F	9-12	37 600	1045	3.6	2.7
Vanka-W	9-10	32 300	896	6.7	9.2
pr-cor mg (9)	19-22	106 000	2960	8-11	2-5
Vanka-V	8-10	56 500	807	2.4	1.1
Vanka-F	6-7	42 300	605	3.6	2.7
Vanka-W	6	38 000	543	7.1	9.0
pr-cor mg (9)	10	95 900	1370	8-11	2-5
Vanka-V	4-5	56 800	411	2.9	1.1
Vanka-F	3-4	43 100	312	3.6	2.7
Vanka-W	3	39 400	285	6.6	9.1
pr-cor mg (9)	5	86 500	627	7-10	2-4

of the residual only for small time steps. The convergence is very slow and this method is not competitive with the other methods. The behaviour of the pressure Schur complement scheme with multigrid for the solution of (7) is much better. The both possibilities (8) and (9) of the computation of the velocity have not led to significant differences in the benchmark reference values. Using (9) saves roughly 5% of computing time. The main interesting aspect is the decrease of the computing times for smaller time steps, see Table 6. However, for the time steps chosen, which are small enough to obtain good benchmark reference values, the coupled multigrid methods with the Vanka smoother have been superior to the pressure Schur complement schemes. The W-cycle has performed again best in all tests.

7 Summary

Coupled multigrid methods have been proven superior to the pressure Schur complement schemes described in Sect. 4 in the computation of benchmark problems for steady state and time dependent incompressible Navier–Stokes equations. The superiority has been very large in the stationary case. Even for the time dependent problem, they have performed better than the pressure Schur complement scheme with multigrid for the solution of (7). However, the coupled multigrid methods do not profit by a decreasing length of the time step. The total computing times for a fixed time interval have increased slightly with the decrease of the time step. In contrary, the computing times using the pressure Schur complement scheme with multigrid have decreased for smaller time steps. This has been observed also by Turek [16]. Thus, it can be expected that in problems which require very fine temporal resolution, the pressure Schur complement scheme with multigrid may be superior to the coupled multigrid methods. Pressure Schur complement schemes with single grid methods for the solution of (7) have shown a very poor performance in all tests. A serious drawback of the pressure Schur complement schemes seems to us the sensitivity to the damping factor α. As far as we know, there is neither a general guideline nor an a posteriori choice so far presented in the literature. The coupled multigrid methods with the modifications described in Sect. 5 have shown in all tests a smaller communication overhead than the pressure Schur complement schemes.

Acknowledgements. The author wishes to acknowledge Dr. F. Schieweck and Prof. L. Tobiska for the discussions on the subject of this paper.

References

1. Axelsson, O., Neytcheva, M.: Scalable algorithms for the solution of Navier's equations of elasticity. J. Comput. Appl. Math., 63:149–178, 1995
2. Bastian, P. Parallele adaptive Mehrgitterverfahren. Teubner Skripten zur Numerik. Teubner-Verlag, 1996
3. Bastian, P, Birken, K., Johannsen, K., Lang, S., Neuß, N., Rentz-Reichert, H., Wieners, C.: UG – a flexible software toolbox for solving partial differential equations. Comput. Visual. Sci., 1:27–40, 1997
4. Crouzeix, M., Raviart, P.-A.: Conforming and nonconforming finite element methods for solving the stationary Stokes equations I. R.A.I.R.O. Analyse Numérique, 7:33–76, 1973
5. Dorok, O.: Improved accuracy of a finite element discretization for solving the Boussinesq approximation of the Navier–Stokes equations. In Proceedings of the Second Summer Conference: Numerical Modelling in Continuum Mechanics (Theorie, Algorithms, Applications), Prague, August 22–25, 1994
6. John, V.: Parallele Lösung der inkompressiblen Navier–Stokes Gleichungen auf adaptiv verfeinerten Gittern. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Fakultät für Mathematik, 1997
7. John, V.: On the parallel performance of coupled multigrid methods for the solution of incompressible Navier–Stokes equations. In Griebel, M., Iliev, O.P., Margenov, S.D., Vassilevski, P.S. (Eds.). Large-Scale Scientific Computations of Engineering and Environmental Problems, Vol. 62 of Notes on Numerical Fluid Mechanics, pages 269–280. Vieweg, 1998
8. McBryan, O.A., Frederickson, P.O., Linden, J., Schüler, A., Solchenbach, K., Stüben, K., Thole, C.-A., Trottenberg, U.: Multigrid methods on parallel computers – a survey of recent developments. Impact Comp. Sci. Engrg., 3:1–75, 1991
9. Oswald, P.: On a hierarchical basis multilevel method with nonconforming p1 elements. Numer. Math., 62:189–212, 1992
10. Patankar, S.V., Spalding, D.B.: A calculation procedure for heat and mass transfer in three-dimensional parabolic flows. Int. J. Heat Mass Transfer, 15:1787–1806, 1972
11. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 7(3):856–869, 1986
12. Schäfer, M., Turek, S.: The benchmark problem "Flow around a cylinder". In Hirschel, E.H. (Ed.). Flow Simulation with High-Performance

Computers II, Vol. 52 of Notes on Numerical Fluid Mechanics. Vieweg, 1996
13. Schieweck, F.: A parallel multigrid algorithm for solving the Navier–Stokes equations. Impact Comp. Sci. Engrg., 5:345–378, 1993
14. Schieweck, F.: Parallele Lösung der stationären inkompressiblen Navier–Stokes Gleichungen. Otto-von-Guericke Universität Magdeburg, Fakultät für Mathematik, 1996. Habilitation
15. Schieweck, F., Tobiska, L.: An optimal order error estimate for an upwind discretization of the Navier–Stokes equations. Num. Meth. Part. Diff. Equ., 12:407–421, 1996
16. Turek, S.: A comparative study of time-stepping techniques for the incompressible Navier–Stokes equations: From fully implicit nonlinear schemes to semi-implicit projection methods. Int. J. Num. Meth. Fluids, 22:987–1011, 1996
17. Vanka, S.: Block-implicit multigrid calculation of two-dimensional recirculating flows. Comp. Meth. Appl. Mech. Eng., 59(1):29–48, 1986
18. Wittum, G.: Multi-grid methods for Stokes and Navier–Stokes equations. Transforming smoothers: Algorithms and numerical results. Numer. Math., 54:543–563, 1989