INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **50**:845–862
Published online 30 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1080

On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3D flow around a cylinder

Volker John*,†

FR 6.1—Mathematik, Universität des Saarlandes, Postfach 15 11 50, 66041 Saarbrücken, Germany

SUMMARY

This paper studies the efficiency of two ways to treat the non-linear convective term in the time-dependent incompressible Navier–Stokes equations and of two multigrid approaches for solving the arising linear algebraic saddle point problems. The Navier–Stokes equations are discretized by a second-order implicit time stepping scheme and by inf–sup stable, higher order finite elements in space. The numerical studies are performed at a 3D flow around a cylinder. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: incompressible Navier–Stokes equations; fixed point iteration; Newton's method; coupled multigrid method; higher-order finite elements

1. INTRODUCTION

The efficient and accurate simulation of incompressible flows by solving the incompressible Navier–Stokes equations is of importance not only by itself but it is also the core of many complex applications, e.g. like the simulation of crystal growth [1, 2] or of fuel cells [3].

The numerical solution of the Navier–Stokes equations requires basically the choice of discretizations in time and space, the choice of an iterative scheme for treating the characteristic non-linear convective term and the choice of a solver for the arising algebraic saddle point problems. In the last decades, many approaches for solving these equations have been developed, see References [4, 5] for overviews. The challenge nowadays consists in combining accuracy of the solution and efficiency of the solution process.

This paper presents a numerical evaluation of two different ways to deal with the non-linear convective term in the Navier–Stokes equations and of two different coupled

*Correspondence to: Volker John, FR 6.1—Mathematik, Universität des Saarlandes, Postfach 15 11 50, 66041 Saarbrücken, Germany.

†E-mail: john@math.uni-sb.de

Received 4 May 2005
Revised 13 July 2005
Accepted 17 July 2005

multigrid approaches for solving the linear algebraic saddle point problems. This evaluation will be done at a particular example—a 3D time-dependent laminar flow around a cylinder. Flows around obstacles are of high practical interest. The consideration of a laminar flow allows the application of Galerkin discretizations whereas the simulation of turbulent flows requires the application of a turbulence model. Turbulence modelling is an active field of research, see References [6, 7] for recent developments. A turbulence model introduces in general additional non-linear terms and it will also influence the behaviour of the multigrid methods. In the numerical studies presented in this paper, we want to concentrate on basic discretizations of the Navier–Stokes equations without further modelling and therefore we do not consider turbulent flows. The investigation of iteration schemes to treat the non-linearities and multigrid solvers for the simulation of such flows will be subject of a forthcoming study.

The Crank-Nicolson scheme will be used as temporal discretization of the Navier–Stokes equations. This is a second-order implicit scheme. It has been shown that such schemes are necessary for performing accurate simulations of laminar incompressible flows [8–11]. As discretization of the linear saddle point problems which arise after a linearization of the Navier–Stokes equations, we use inf–sup stable pairs of finite element spaces of higher order. The fulfillment of the inf–sup condition guarantees the unique solvability of the algebraic saddle point problems [12], without having to choose an additional pressure stabilization. The use of higher order finite elements (at least second-order velocity and first-order pressure) is necessary to obtain accurate solutions for flows around obstacles, e.g. for the drag and the lift coefficient at the obstacle, see References [8, 10, 13–16]. We will present studies with the popular Taylor-Hood finite element of second and third order on tetrahedral grids and with second- and third-order finite elements with discontinuous pressure approximation on hexahedral grids.

The two ways to treat the non-linear convective term of the Navier–Stokes equations are a fixed point iteration and Newton's method. These ways, their advantages and drawbacks will be presented in detail in Section 3.

In each step of the fixed point iteration and Newton's method, a large linear algebraic saddle point problem has to be solved, see Reference [17] for a state-of-the-art overview on solving such problems. It was found in numerical studies that so-called coupled multigrid methods with local smoothers are among the most efficient approaches [8, 9, 18, 19]. Coupled multigrid methods compute the solution for both types of unknowns, the velocity and the pressure, simultaneously. We will study their behaviour as well as solver and as preconditioner in the Krylov subspace method Flexible GMRES [20, 21].

It is well known that multigrid methods work much more efficient for low order discretizations than for higher order ones. Based on this observation, a so-called multiple discretization multilevel method was developed and analysed in References [13, 14, 22] which uses in the multigrid hierarchy lower-order discretizations on coarser levels. This multigrid approach will be compared in the numerical studies with the standard one. Both multigrid approaches and their components are described in Section 4.

The paper is organized as follows. The problem of the 3D flow around a cylinder is introduced in Section 2. Section 3 describes the discretization of the Navier–Stokes equations and the two ways of linearization which are studied in this paper. The multigrid methods are presented in Section 4. Section 5 contains the results of the numerical studies and final conclusions are presented in Section 6.

2. A TIME-DEPENDENT 3D FLOW AROUND A CYLINDER

We consider the flow around a cylinder which was defined in Reference [8] as a benchmark problem within the DFG high priority research program *Flow Simulation with High-Performance Computers*. The flow is governed by the incompressible Navier–Stokes equations:

$$
\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} - \nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \frac{1}{\rho}\nabla p &= \mathbf{0} \quad \text{in } (0,8]\times\Omega \\
\nabla\cdot\mathbf{u} &= 0 \quad \text{in } [0,8]\times\Omega \\
\mathbf{u} &= \mathbf{g}(t,\mathbf{x}) \quad \text{on } [0,8]\times\partial\Omega_{\text{in}} \\
(-p\mathbf{I} + \nu\nabla\mathbf{u})\mathbf{n} &= \mathbf{0} \quad \text{on } [0,8]\times\partial\Omega_{\text{out}} \\
\mathbf{u} &= \mathbf{0} \quad \text{on } [0,8]\times\partial\Omega\setminus(\partial\Omega_{\text{in}}\cup\partial\Omega_{\text{out}}) \\
\mathbf{u}(0,\cdot) &= \mathbf{0} \quad \text{in } \Omega
\end{aligned}
\tag{1}
$$

Figure 1 presents the domain Ω. The boundary of Ω is denoted by $\partial\Omega$, the outer normal by $\mathbf{n}$ and $\mathbf{x}=(x,y,z)$. The channel has a height of $H=0.41$ m and the diameter of the cylinder is $D=0.1$ m. Parameters of the fluid are the kinematic viscosity $\nu=10^{-3}\,\mathrm{m}^2/\mathrm{s}$ and the density

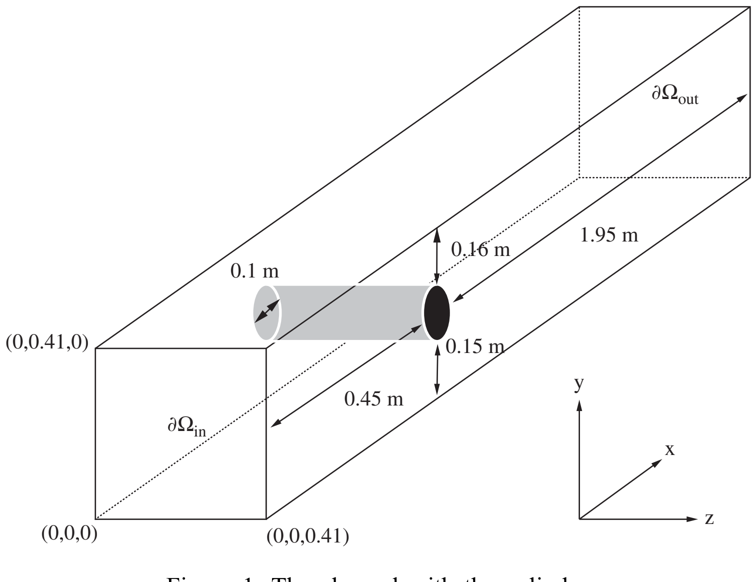

Figure 1. The channel with the cylinder.

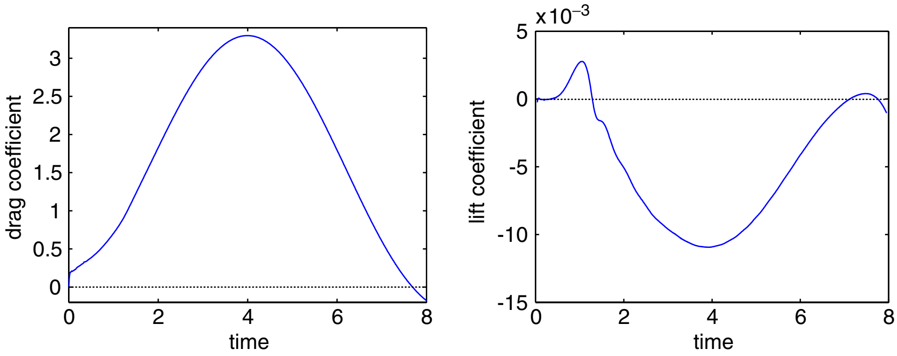

Figure 2. Drag and lift coefficient at the cylinder.

$\rho = 1\,\mathrm{kg/m^3}$. The inflow is prescribed by

$$\mathbf{g}(t,\mathbf{x}) = \begin{pmatrix} 16U \sin\left(\pi \dfrac{t}{8}\right) \dfrac{yz(H-y)(H-z)}{H^4} \\ 0 \\ 0 \end{pmatrix}$$

with $U = 2.25\,\mathrm{m/s}$. The flow has the Reynolds number $0 \leqslant Re(t) \leqslant 100$ based on ν, D and the mean inflow $\bar{U}(t) = \sin(\pi t/8)\mathrm{m/s}$.

Important features of this flow are the drag and the lift coefficient at the cylinder, see Figure 2. Our numerical simulations for the computation of these coefficients were performed using the Crank-Nicolson scheme with an equi-distant time step $\Delta t = 0.01$ s and the Q_2/P_1^{disc} finite element discretization in space on a very fine mesh (6 052 800 velocity degrees of freedom (d.o.f.), 983 040 pressure d.o.f.). The drag and lift coefficients have been computed using volume integrals like, e.g.

$$c_{\mathrm{d}}(t) = -\frac{2}{\rho D \bar{U}_{\max}^2 H} \int_\Omega \left[\rho \left(\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v}_{\mathrm{d}} + \nu \nabla \mathbf{u}(t) : \nabla \mathbf{v}_{\mathrm{d}} + (\mathbf{u}(t)\cdot\nabla)\mathbf{u}(t)\cdot \mathbf{v}_{\mathrm{d}}\right) - p(t)(\nabla \cdot \mathbf{v}_{\mathrm{d}})\right] \mathrm{d}\mathbf{x}$$

with $\bar{U}_{\max} = 1\,\mathrm{m/s}$ and an appropriately defined dimensionless test function $\mathbf{v}_{\mathrm{d}}$, see [13, 14, 10] for details. The maximal drag coefficient $c_{\mathrm{d,max}} = 3.2968$ is obtained at $t = 4.00$ s. The lift coefficient is around zero. A vortex shedding behind the cylinder cannot be observed in this problem.

3. DISCRETIZATION OF THE NAVIER–STOKES EQUATIONS AND APPROACHES FOR THE LINEARIZATION

The discretization and linearization approach for (1) which has been used consists of the following steps:

1. *Semi-discretization of* (1) *in time.* An implicit time stepping scheme is used. The semi-discretization in time leads in each discrete time step to a non-linear system of equations.

2. *Variational formulation and linearization.* The non-linear system of equations is reformulated as variational problem and the non-linear variational problem is linearized by an iterative scheme.
3. *Discretization of the linear saddle point problems in space.* The linear saddle point problems arising in each step of the iterative scheme are discretized by a finite element discretization using an inf–sup stable pair of finite element spaces.

In the first step, the Crank-Nicolson scheme was used as temporal discretization in the numerical studies presented in Section 5. Let Δt_n be the current time step from t_{n-1} to t_n, i.e. $\Delta t_n = t_n - t_{n-1}$. Then, the semi-discretization in time of the dimensionless form of the Navier–Stokes equations (1) with the Crank-Nicolson scheme has the form

$$\begin{aligned} \mathbf{u}_n + 0.5\Delta t_n[-Re^{-1}\Delta\mathbf{u}_n + (\mathbf{u}_n\cdot\nabla)\mathbf{u}_n] + \Delta t_n\nabla p_n \\ = \mathbf{u}_{n-1} - 0.5\Delta t_n[-Re^{-1}\Delta\mathbf{u}_{n-1} + (\mathbf{u}_{n-1}\cdot\nabla)\mathbf{u}_{n-1}] \\ \nabla\cdot\mathbf{u}_n = 0 \end{aligned}$$

The Crank-Nicolson scheme is one of the most popular time-stepping schemes for the incompressible Navier–Stokes equations. In a competitive study of such schemes, [11], the Crank-Nicolson scheme was in general twice as fast as the fractional-step θ-scheme. It is well known that the fractional-step θ-scheme is more stable (strongly A-stable) than the Crank-Nicolson scheme (A-stable). However, we did not encounter stability problems with the Crank-Nicolson scheme in the numerical studies presented in this paper.

To describe the variational formulation in the second step of our approach, some function spaces are introduced: $V_0 = (H_0^1(\Omega))^{\mathrm{d}}$, $V_{\mathbf{g}} = \{\mathbf{v} : \mathbf{v}\in(H^1(\Omega))^{\mathrm{d}}, \mathbf{v}|_{\partial\Omega_{\text{in}}} = \mathbf{g}\}$, and $Q = L^2(\Omega)$. Here, $L^2(\Omega)$ is the Lebesgue space of square integrable functions and $H^1(\Omega)$ the Sobolev space of square integrable functions whose first (weak) derivative is also square integrable. The functions of $H_0^1(\Omega)$ are those of $H^1(\Omega)$ which vanish on $\partial\Omega$.

The variational problem reads as follows: find $(\mathbf{u}_n, p_n)\in V_{\mathbf{g}}\times Q$ such that for all $(\mathbf{v}, q)\in V_0 Q$

$$\begin{aligned} (\mathbf{u}_n,\mathbf{v}) + 0.5\Delta t_n[(Re^{-1}\nabla\mathbf{u}_n,\nabla\mathbf{v}) + ((\mathbf{u}_n\cdot\nabla)\mathbf{u}_n,\mathbf{v})] - \Delta t_n(p_n,\nabla\cdot\mathbf{v}) \\ = (\mathbf{u}_{n-1},\mathbf{v}) - 0.5\Delta t_n[(Re^{-1}\nabla\mathbf{u}_{n-1},\nabla\mathbf{v}) + ((\mathbf{u}_{n-1}\cdot\nabla)\mathbf{u}_{n-1},\mathbf{v})] \\ 0 = (\nabla\cdot\mathbf{u}_n, q) \end{aligned} \tag{2}$$

This is a non-linear system of equations which we solve iteratively starting with an initial guess $(\mathbf{u}_n^0, p_n^0)$. The initial guess is chosen to be the solution of the previous time step $(\mathbf{u}_n^0, p_n^0) = (\mathbf{u}_{n-1}, p_{n-1})$. The comparison of two approaches for this iteration is one of the main topics of the numerical studies presented in Section 5.

The first iterative scheme is a fixed point iteration where the non-linear convective term is approximated by

$$(\mathbf{u}_n^{m+1}\cdot\nabla)\mathbf{u}_n^{m+1} \approx (\mathbf{u}_n^m\cdot\nabla)\mathbf{u}_n^{m+1}$$

with a known velocity field $\mathbf{u}_n^m$. This leads to the following iteration for solving (2): Given $\mathbf{u}_n^m \in V_\mathbf{g}$, the iterate $(\mathbf{u}_n^{m+1}, p_n^{m+1}) \in V_\mathbf{g} \times Q$ is computed by solving

$$\begin{aligned}
(\mathbf{u}_n^{m+1}, \mathbf{v}) + 0.5\Delta t_n[(Re^{-1}\nabla \mathbf{u}_n^{m+1}, \nabla \mathbf{v}) + ((\mathbf{u}_n^m \cdot \nabla)\mathbf{u}_n^{m+1}, \mathbf{v})] - \Delta t_n(p_n^{m+1}, \nabla \cdot \mathbf{v}) \\
= (\mathbf{u}_{n-1}, \mathbf{v}) - 0.5\Delta t_n[Re^{-1}(\nabla \mathbf{u}_{n-1}, \nabla \mathbf{v}) + ((\mathbf{u}_{n-1} \cdot \nabla)\mathbf{u}_{n-1}, \mathbf{v})] \\
0 = (\nabla \cdot \mathbf{u}_n^{m+1}, q)
\end{aligned} \tag{3}$$

for all $(\mathbf{v}, q) \in V_0 \times Q$, $m = 0, 1, 2, \ldots$. Equations (3) are called Oseen equations.

The second approach, a Newton iteration, uses the linearization

$$(\mathbf{u}_n^{m+1} \cdot \nabla)\mathbf{u}_n^{m+1} \approx (\mathbf{u}_n^m \cdot \nabla)\mathbf{u}_n^{m+1} + (\mathbf{u}_n^{m+1} \cdot \nabla)\mathbf{u}_n^m - (\mathbf{u}_n^m \cdot \nabla)\mathbf{u}_n^m$$

of the non-linear convective term. The corresponding iterative scheme is: given $\mathbf{u}_n^m \in V_\mathbf{g}$, compute $(\mathbf{u}_n^{m+1}, p_n^{m+1}) \in V_\mathbf{g} \times Q$ such that

$$\begin{aligned}
(\mathbf{u}_n^{m+1}, \mathbf{v}) + 0.5\Delta t_n[(Re^{-1}\nabla \mathbf{u}_n^{m+1}, \nabla \mathbf{v}) + ((\mathbf{u}_n^m \cdot \nabla)\mathbf{u}_n^{m+1}, \mathbf{v}) + ((\mathbf{u}_n^{m+1} \cdot \nabla)\mathbf{u}_n^m, \mathbf{v})] \\
-\Delta t_n(p_n^{m+1}, \nabla \cdot \mathbf{v}) \\
= (\mathbf{u}_{n-1}, \mathbf{v}) - 0.5\Delta t_n[Re^{-1}(\nabla \mathbf{u}_{n-1}, \nabla \mathbf{v}) + ((\mathbf{u}_{n-1} \cdot \nabla)\mathbf{u}_{n-1}, \mathbf{v})] \\
+0.5\Delta t_n((\mathbf{u}_n^m \cdot \nabla)\mathbf{u}_n^m, \mathbf{v}) \\
0 = (\nabla \cdot \mathbf{u}_n^{m+1}, q)
\end{aligned} \tag{4}$$

for all $(\mathbf{v}, q) \in V_0 \times Q$, $m = 0, 1, 2, \ldots$.

An alternative to the full Newton iteration is a quasi Newton iteration which leaves the left-hand side of (4) unchanged and updates only the right-hand side by the computed iterates. This approach is used, e.g. in Reference [16].

The final step of our approach consists in applying an inf–sup stable finite element method for discretizing (3) and (4) in space. This gives a linear algebraic saddle point problem of the form

$$\mathscr{A}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{5}$$

We use the standard approach that the velocity finite element spaces V_0^h and $V_\mathbf{g}^h$ are products of three scalar finite element spaces, e.g. V_0^h is given by

$$\begin{aligned}
V_0^h &= \mathrm{span}\{\mathbf{v}_i^h\}_{i=1}^{3N_v} \\
&= \mathrm{span}\left\{\left\{\begin{pmatrix} v_i^h \\ 0 \\ 0 \end{pmatrix}\right\}_{i=1}^{N_v} \cup \left\{\begin{pmatrix} 0 \\ v_i^h \\ 0 \end{pmatrix}\right\}_{i=1}^{N_v} \cup \left\{\begin{pmatrix} 0 \\ 0 \\ v_i^h \end{pmatrix}\right\}_{i=1}^{N_v}\right\}
\end{aligned}$$

Here, N_v is the number of d.o.f. for a scalar component of the velocity. Denoting the number of pressure d.o.f. by N_p, the A block of the system matrix in (5) has the dimensions $3N_v \times 3N_v$ and the B block has the dimension $N_p \times 3N_v$. The symbol $A(i,j)$, $i,j=1,\ldots,3N_v$, is used for an entry of the matrix block A. This matrix block can be decomposed into nine subblocks of dimension $N_v \times N_v$. The notation A_{kl}, $k,l \in \{1,2,3\}$, stands for the subblock corresponding to the lth velocity component from the ansatz space and the kth velocity component from the test space. The kth scalar component of the velocity vector $\mathbf{v}_i^h$ is denoted by $(\mathbf{v}_i^h)_k$.

It turns out that the structure of the left upper block A of the system matrix in (5) looks different for the discretization of the Oseen equations (3) and the discrete equations coming from Newton's iteration (4).

Let $\mathbf{u}_{\text{old}}^h$ be a given finite element approximation of the velocity field. In the fixed point iteration (3), the matrix entries are given by

$$A(i,j) = \int_\Omega (\mathbf{u}_{\text{old}}^h \cdot \nabla)\mathbf{v}_j^h \cdot \mathbf{v}_i^h \,\mathrm{d}\mathbf{x} = \sum_{k=1}^{3} \int_\Omega (\mathbf{u}_{\text{old}}^h \cdot \nabla(\mathbf{v}_j^h)_k) \cdot (\mathbf{v}_i^h)_k \,\mathrm{d}\mathbf{x}$$

This entry vanishes if the non-vanishing component of $\mathbf{v}_i^h$ and $\mathbf{v}_j^h$ is different. If this component is the same, the matrix entry is the same independent of the component. Thus, from the discretization of (3), one obtains a matrix block of the form

$$A = \begin{pmatrix} A_{11} & 0 & 0 \\ 0 & A_{11} & 0 \\ 0 & 0 & A_{11} \end{pmatrix}$$

The matrix entries for Newton's iteration (4) are given by

$$A(i,j) = \sum_{k=1}^{3} \int_\Omega \left[(\mathbf{u}_{\text{old}}^h \cdot \nabla(\mathbf{v}_j^h)_k) \cdot (\mathbf{v}_i^h)_k + \left(\sum_{l=1}^{3} (\mathbf{v}_j^h)_l \cdot \nabla(\mathbf{u}_{\text{old}}^h)_l \right) \cdot (\mathbf{v}_i^h)_k \right] \mathrm{d}\mathbf{x}$$

From the second term follows that for all pairs (k,l), the matrix blocks A_{kl} posses in general non-zero entries. Altogether, the matrix which has to be used for Newton's method has the block form

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

where the blocks are in general mutually different since different derivatives of $\mathbf{u}_{\text{old}}^h$ has to be considered in the assembling of each block.

In summary, using the fixed point iteration (3) requires less memory and matrix–vector products with the system matrix need less floating point operations. In addition, much more assembling of matrices has to be performed in Newton's method (4). Note, assembling is quite expensive for higher-order finite elements in 3D. One needs quadrature rules which are sufficiently accurate such that the accuracy of the finite element is not spoiled by quadrature errors. These quadrature rules posses a lot of quadrature points where the finite element basis functions and $\mathbf{u}_{\text{old}}^h$ have to be evaluated, see Reference [23] for details.

The advantages of the fixed point iteration become smaller if instead of the gradient formulation of the viscous term $(\nu\nabla\mathbf{u}^h, \nabla\mathbf{v}^h)$ the deformation tensor formulation $(2\nu\mathbb{D}(\mathbf{u}^h), \mathbb{D}(\mathbf{v}^h))$, $\mathbb{D}(\mathbf{u}^h) = (\nabla\mathbf{u}^h + (\nabla\mathbf{u}^h)^{\mathrm{T}})/2$, is used. The block structure of the matrix corresponding to the deformation tensor formulation is

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{12}^{\mathrm{T}} & A_{22} & A_{23} \\ A_{13}^{\mathrm{T}} & A_{23}^{\mathrm{T}} & A_{33} \end{pmatrix}$$

Thus, at least six blocks have to be stored. The deformation tensor formulation is correct from the physical point of view. It should be used for the simulation of turbulent flows. In addition, there are some boundary conditions, like slip with friction boundary conditions, which require this formulation of the viscous term.

We like to emphasize that there is a crucial difference concerning the iteration schemes between the steady state and the time-dependent Navier–Stokes equations. The initial guess $(\mathbf{u}_n^0, p_n^0) = (\mathbf{u}_{n-1}, p_{n-1})$ in the time-dependent case is very often quite close to the solution in the current discrete time t_n, in particular, if Δt_n is small. Thus, one expects that only very few fixed point iterations or Newton steps are necessary to compute the solution in t_n. For the steady state Navier–Stokes equations such a good initial iterate is in general not available, even if the prolongation of the solution from a coarser grid is used. Thus, the results presented in this paper cannot be simply carried over to the steady state Navier–Stokes equations and these equations require new studies.

4. THE SOLVER OF THE LINEAR ALGEBRAIC SADDLE POINT PROBLEMS

This paper studies also the efficiency of coupled multigrid methods for solving the linear algebraic saddle point problems which arise in the linearization and discretization of the time-dependent flow problem described in Section 2. These coupled multigrid methods are described in detail in References [14, 23]. To keep this paper self explaining, we will give here a short overview on their most important components.

A multigrid method is given by its grid hierarchy, the grid transfer operator, the smoother on the finer levels, the coarse grid solver and the way of going through the grid hierarchy (the kind of multigrid cycle). The grid transfer operator is described in References [22–24]. We used the most common types of cycles, which are the V-, the F- and the W-cycle.

The initial triangulation of a domain, which is not too simple, like the channel with the cylinder, is already rather fine. Since each uniform refinement increases the number of degrees of freedom by the factor eight, the memory limits of present day computers are reached soon. Thus, one characteristic feature of multigrid methods in 3D is that the multigrid hierarchy possesses only few levels.

The multiplicative Vanka smoother was applied as smoother. Each smoothing step of a Vanka-type smoother requires the solution of a number of local problems. The local updates are appropriately combined to a global update. Vanka-type smoothers start by a decomposition of the sets of the velocity d.o.f. $\mathscr{V}^h$ and the pressure d.o.f. $\mathscr{Q}^h$ into J subsets

$$\mathscr{V}^h = \bigcup_{j=1}^{J} \mathscr{V}_j^h, \quad \mathscr{Q}^h = \bigcup_{j=1}^{J} \mathscr{Q}_j^h \tag{6}$$

Then, for each index j, one takes the d.o.f. in $\mathscr{V}_j^h \cup \mathscr{Q}_j^h$ and constructs a small matrix $\mathscr{A}_j$ by the intersection of the rows and the columns of the global matrix $\mathscr{A}$ which correspond to the d.o.f. in $\mathscr{V}_j^h \cup \mathscr{Q}_j^h$. One step of a Vanka-type smoother consists in a loop from $j = 1,\ldots,J$, where for each index j a local system of the form

$$\begin{pmatrix} u \\ p \end{pmatrix}_j := \begin{pmatrix} u \\ p \end{pmatrix}_j + \mathscr{A}_j^{-1} \left(\begin{pmatrix} f \\ g \end{pmatrix} - \mathscr{A} \begin{pmatrix} u \\ p \end{pmatrix} \right)_j$$

is solved. This gives a local update. We refresh the vector which contains the global update immediately with each new local update. This is a block Gauss–Seidel approach which is called *multiplicative Vanka smoother*. The multiplicative Vanka smoother is now completely described if the decompositions (6) are given. Our strategy to obtain them is as follows:

1. Take some pressure d.o.f. which form $\mathscr{Q}_j^h$.
2. The corresponding velocity d.o.f. in $\mathscr{V}_j^h$ are all those which are connected to the pressure d.o.f. in the $\mathscr{Q}_j^h$ by entries in the off diagonal block B.

Thus, $\mathscr{V}_j^h$ is completely described by the choice of $\mathscr{Q}_j^h$ and the connections of the velocity and pressure d.o.f. of the underlying pair of finite element spaces. Concerning the choice of $\mathscr{Q}_j^h$, we distinguish two situations:

1. The pressure is discretized by a discontinuous finite element space. Then, we take for $\mathscr{Q}_j^h$ all pressure d.o.f. which belong to one mesh cell. It turns out that the corresponding velocity d.o.f. in $\mathscr{V}_j^h$ are all velocity d.o.f. which belong to the same mesh cell. Therefore, this approach is called *mesh cell oriented Vanka smoother*. The number of local systems J to solve in each step of the mesh cell oriented Vanka smoother is equal to the number of mesh cells.
2. The pressure is discretized by a continuous finite element space. Then, we take for each $\mathscr{Q}_j^h$ only one pressure d.o.f. This defines the *pressure node oriented Vanka smoother*. Here, the number of local systems J to solve in each step of the pressure node oriented Vanka smoother corresponds to the number of pressure d.o.f.

For detailed illustrations of the mesh cell oriented and the pressure node oriented Vanka smoother, we refer to Reference [14]. Examples for the size of the local systems for different pairs of finite element spaces can be found as well in Reference [14].

The size of the local systems using higher-order finite element discretizations may be rather large. In our numerical studies, systems with dimension lower or equal than 100 have been solved by Gaussian elimination. Larger systems were solved only approximately by applying 10 iterations with GMRES [25].

The systems on the coarsest grid, level 0, are solved approximately by the multiplicative Vanka smoother. The iteration is stopped either after having reduced the Euclidean norm of the residual by the factor 10 or after 100 iterations.

Sometimes, it becomes necessary to damp the smoother iteration. Let (u_l, p_l) be the current iterate on the multigrid level l and $(\delta u_l, \delta p_l)$ be the update computed by one iteration of the smoother. Then, the new iterate is given by $(u_l, p_l) + \omega_l(\delta u_l, \delta p_l)$. Another possibility of damping exists after the prolongation. Let $(\delta u_{l-1}, \delta p_{l-1})$ be the update which was prolongated from level $l-1$ to level l. Instead of accepting this update, one can use $\gamma_l(\delta u_{l-1}, \delta p_{l-1})$ with an appropriately chosen parameter γ_l. The damping parameters can be chosen differently on all levels of the multigrid hierarchy.

The standard multigrid approach assigns to each level of the geometric grid hierarchy exactly one level of the multigrid hierarchy and on each level of the multigrid hierarchy, the same discretization is used. Besides this standard approach, we consider in the numerical studies also two types of so-called multiple discretization multilevel methods (mdml). A multiple discretization multilevel method has at least two levels of the multigrid hierarchy on the finest geometric grid. One of them, which forms the top of the multigrid hierarchy, uses the discretization of interest, e.g. in our studies a higher-order discretization. The discretizations of the other levels of the multigrid hierarchy on the finest geometric grid are of lower order. On the coarser geometric grids, only lowest order discretizations are used. All multigrid approaches which were studied are illustrated in Table I. Type 2 of the multiple discretization multilevel method will be studied the first time in this paper.

For discretizing the Oseen equations (3) and the equations arising in Newton's iteration (4) with a lowest order discretization on the coarser grids, we used the non-conforming Crouzeix–Raviart finite element on tetrahedral grids (P_1^{nc}/P_0), [26], and the non-conforming Rannacher–Turek finite element on hexahedral grids, (Q_1^{rot}/Q_0), [27]. Since (3) and (4) are convection dominated equations, we stabilized these lowest order non-conforming discretizations with a Samarskij upwinding scheme [28].

The saddle point problem (5) has the same abstract form for the steady state and the time-dependent Navier–Stokes equations. However, the properties of the matrix block A are different in both cases. Whereas this block consists of a diffusive and a convective part for the steady state equations, an additional symmetric matrix, the mass matrix, occurs for the time-dependent Navier–Stokes equations. Moreover, the mass matrix is in general dominant since the diffusive and convective part are multiplied with the step size Δt_n which is generally small. This leads to fundamental differences in the properties of the matrix block A in both cases. For this reason, it is not possible simply to carry over the results for the steady state Navier–Stokes equations from Reference [14] to the time-dependent case. The time-dependent Navier–Stokes equations require new studies.

Table I. Correspondence of the levels of the multigrid hierarchy (right) to the levels of the geometric grid hierarchy (left).

Geometric level	Standard multigrid	mdml type 1	mdml type 2	Multigrid level
L	—	—	Third order	L + 2
L	—	Higher order	Second order	L + 1
L	Higher order	Lowest order	Lowest order	L
L − 1	Higher order	Lowest order	Lowest order	L − 1
⋮	⋮	⋮	⋮	⋮
1	Higher order	Lowest order	Lowest order	1
0	Higher order	Lowest order	Lowest order	0

It was found in numerical studies for the steady state Navier–Stokes equations in Reference [14] that all multigrid methods were more efficient and more robust as preconditioner in the Krylov subspace iterative method flexible GMRES (FGMRES), [20, 21], than as solver. We could observe the same in the numerical studies for time-dependent problems. This statement will be supported by presenting one characteristic result of using the multigrid methods as solver. Otherwise, we will concentrate on their behaviour as preconditioner in FGMRES.

5. THE NUMERICAL STUDIES

This section presents numerical studies which compare

- the efficiency of using a fixed point iteration (3) vs Newton's method (4),
- the efficiency of the different multigrid approaches presented in Section 4.

The computations were performed on hexahedral and tetrahedral grids using discretizations of order k for the velocity and $k-1$ for the pressure, $k \in \{2, 3\}$. Since we obtained similar results for the Taylor–Hood finite elements on hexahedral and tetrahedral grids, we present only the results for the tetrahedral grids. The initial grids are shown in Figure 3.

The Crank-Nicolson scheme was applied with an equi-distant time step $\Delta t_n = 0.01$ such that 800 time steps were to be computed.

Results concerning parameters of the flow which were obtained on the grids used in the evaluation of the linearization schemes and the multigrid approaches are presented in Table II. They are compared with results for lowest order finite element discretizations (stabilized with Samarskij upwinding) with considerably more d.o.f. as well as with the reference computation presented at the end of Section 2. The lift coefficient $c_l(t)$ was always rather small. Table II gives its largest absolute value. Another benchmark parameter defined in Reference [8] was the difference of the pressure

$$\Delta p := p(8, (0.45, 0.2, 0.205)) - p(8, (0.55, 0.2, 0.205))$$

at the final time $t = 8$ s between one point at the front and one point at the back of the cylinder. The results in Table II emphasize the superior accuracy of higher-order discretizations, above

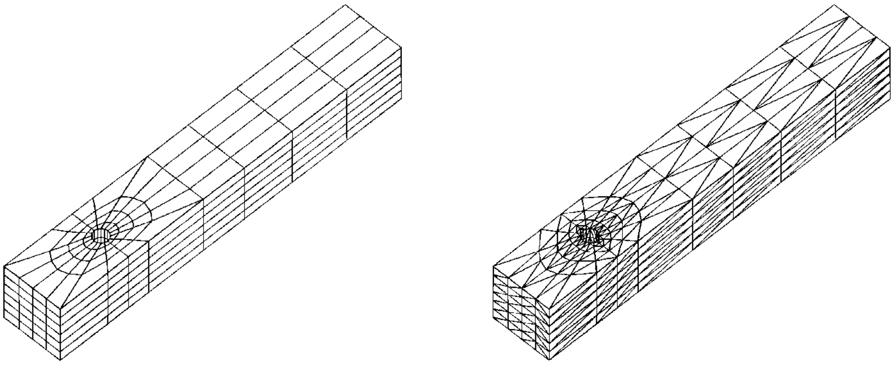

Figure 3. The initial grids.

Table II. Parameters of the flow for the discretizations used in the numerical studies.

Disc	L	d.o.f.	$t_{c_{\mathrm{d}},\max}[s]$	$c_{\mathrm{d},\max}$	$\lvert c_{\mathrm{l},\max}\rvert$	Δp
Q_1^{rot}/Q_0	3	2 496 000	4.06	3.4340	0.0295	−0.0561
P_1^{nc}/P_0	2	1 309 440	3.99	3.6253	0.1046	−0.0777
Q_2/P_1^{disc}	2	899 040	3.96	3.2805	0.0849	−0.1027
P_2/P_1	2	810 160	4.00	3.2847	0.0080	−0.1096
Q_3/P_2^{disc}	1	371 400	4.11	3.3029	0.0611	−0.1023
P_3/P_2	1	367 000	4.00	3.2762	0.0631	−0.1017
Reference: Q_2/P_1^{disc}	3	7 035 840	4.00	3.2968	0.0110	−0.1087

all with respect to the maximal drag coefficient and the difference of the pressure. Almost all computations with fewer d.o.f. than the reference computation showed stronger oscillations of the lift coefficient.

The solution of the linear algebraic saddle point problems arising from the fixed point iteration (3) and Newton's method (4) was stopped after an reduction of the Euclidean norm of the residual by a factor of 10 or after five FGMRES iterations. Iterations (3) and (4) for solving the non-linearity were terminated in each discrete time if the Euclidean norm of the residual was less than 10^{-5}. It happened that this criterion was fulfilled even for the initial iterate in some discrete times t_n. In this case, no iteration was performed in t_n and the simulation of the flow was continued at t_{n+1}.

The computations were performed at a computer with HP PA-RISC 8500 processors (440 MHz, 1760 Mflops/s peak). The computing times are given in seconds. The computations were performed with the code MooNMD [29].

The abbreviations in the tables of this section are the followings:

mg—standard multigrid,
mdml—multiple discretization multilevel method, type 1,
mdml2—multiple discretization multilevel method, type 2,
ite—total number of FGMRES iterations,
t(sol)—time needed by the solver,
t(tot)—total computing time,
%—percentage of solver time with respect to the total computing time,
2lev—2-level method,
Div.—solver diverged,
Slow—computation too slow (much more than 10^6 s).

The three fastest computations with respect to the total computing time and all computations which are within 10% of the fastest one are emphasized.

5.1. Computations with the Q_2/P_1^{disc} finite element on a hexahedral grid

The computations with the second-order velocity and the first order discontinuous pressure were performed on a hexahedral grid which was obtained by two refinements of the initial grid. The numbers of d.o.f. on each level for the standard multigrid method and the multiple discretization multilevel method are given in Table III. Damping within the multigrid methods was not applied in these computations.

Table III. Discretizations and d.o.f. for the computations with the Q_2/P_1^{disc} finite element.

	Standard mg		mdml, type 1	
Level	Disc	d.o.f.	Disc	d.o.f.
3			Q_2/P_1^{disc}	899 040
2	Q_2/P_1^{disc}	899 040	Q_1^{rot}/Q_0	316 800
1	Q_2/P_1^{disc}	117 360	Q_1^{rot}/Q_0	40 800
0	Q_2/P_1^{disc}	15 960	Q_1^{rot}/Q_0	5 400

Table IV. Q_2/P_1^{disc}, multigrid methods as solver.

		Fixed point iteration (3)				Newton's method (4)			
Solver	Cycle	ite	t(sol)	t(tot)	%	ite	t(sol)	t(tot)	%
mg	V(1,1)	Div.	—	—	—	Div.	—	—	—
mg	V(2,2)	762	167 864	*207 570*	80	761	213 059	*363 273*	58
mg	F(1,1)	1300	181 931	*222 182*	81	1285	230 393	*384 315*	59
mg	F(2,2)	758	195 218	234 834	83	755	246 228	*396 645*	62
mg	W(1,1)	1300	182 013	*222 261*	81	1285	228 962	*381 811*	59
mg	W(2,2)	758	195 446	235 064	83	755	244 463	*394 523*	61
mdml	V(1,1)	Div.	—	—	—	Div.	—	—	—
mdml	V(2,2)	1041	200 516	239 851	83	1029	252 121	*396 852*	63
mdml	F(1,1)	Div.	—	—	—	Div.	—	—	—
mdml	F(2,2)	1133	227 255	266 812	85	1107	281 428	426 628	65
mdml	W(1,1)	Slow	—	—	—	Slow	—	—	—
mdml	W(2,2)	Slow	—	—	—	Slow	—	—	—

For the Q_2/P_1^{disc} finite element discretization, we present a comparison of the behaviour of the multigrid approaches as solver and as preconditioner in FGMRES, see Tables IV and V. It can be clearly seen that they are much more efficient and robust as preconditioner. We obtained similar results for all other discretizations and the same observations are reported for the steady state Navier–Stokes equations [14]. For this reason, no further results for the multigrid methods as solver will be presented here.

With respect to the iterative scheme for solving the Navier–Stokes equations, the fixed point iteration (3) was always much more efficient than Newton's method (4). The total number of FGMRES iterations is in general similar such that the large overhead of Newton's method led to considerably longer computing times. In Newton's method, approximately half of the total computing time is needed for other tasks than solving, in particular for assembling matrices.

Both multigrid approaches behaved quite similar for the Q_2/P_1^{disc} finite element. The same was observed for the steady state Navier–Stokes equations in Reference [14].

Table V. Q_2/P_1^{disc}, multigrid methods as preconditioner in FGMRES.

		Fixed point iteration (3)				Newton's method (4)			
Prec.	Cycle	ite	t(sol)	t(tot)	%	ite	t(sol)	t(tot)	%
mg	V(1,1)	861	103 899	*144 838*	71	851	131 534	*288 911*	45
mg	V(2,2)	623	137 334	173 054	79	649	185 788	328 968	56
mg	F(1,1)	689	108 098	147 070	73	708	141 909	*294 550*	48
mg	F(2,2)	618	167 154	202 955	82	640	218 221	357 912	60
mg	W(1,1)	689	107 337	*145 908*	73	708	144 221	299 172	48
mg	W(2,2)	618	166 207	201 704	82	640	219 567	360 197	60
mdml	V(1,1)	955	93 869	*133 613*	70	959	120 967	*269 009*	44
mdml	V(2,2)	703	135 704	173 188	78	723	177 417	319 784	55
mdml	F(1,1)	1010	104 067	*143 137*	72	1038	137 138	*284 523*	48
mdml	F(2,2)	678	136 576	173 729	78	694	177 442	317 777	55
mdml	W(1,1)	1231	127 144	166 545	76	1178	155 536	300 343	51
mdml	W(2,2)	926	186 439	226 372	82	851	218 545	367 006	59

Table VI. Discretizations and d.o.f. for the computations with the Q_3/P_2^{disc} finite element.

	Standard mg		mdml, type 1		mdml, type 2	
Level	Disc	d.o.f.	Disc	d.o.f.	Disc	d.o.f.
3					Q_3/P_2^{disc}	371 400
2			Q_3/P_2^{disc}	371 400	Q_2/Q_1^{disc}	117 360
1	Q_3/P_2^{disc}	371 400	Q_1^{rot}/Q_0	40 800	Q_1^{rot}/Q_0	40 800
0	Q_3/P_2^{disc}	49 260	Q_1^{rot}/Q_0	5 400	Q_1^{rot}/Q_0	5 400

5.2. Computations with the Q_3/P_2^{disc} finite element on a hexahedral grid

The finest geometric level in these computations was level $L = 1$. Thus, the standard multigrid is just a 2-level method. The multiple discretization multilevel method type 1 possesses three levels and the multiple discretization multilevel method type 2 has even four levels. The numbers of degrees of freedom on all levels of the multigrid hierarchies are given in Table VI. All damping parameters in the multigrid methods were set to be 1.0.

Table VII presents the computational results for the multigrid methods as preconditioner in FGMRES. Concerning the fixed point iteration (3) and Newton's method (4), the situation is the same as for the Q_2/Q_1^{disc} finite element discretization. The huge overhead of Newton's method is again remarkable.

The computational results show that type 1 of the multiple discretization multilevel method is often more efficient than type 2 with respect to the total computing time. The number of FGMRES iterations is general smaller for type 2. However, the costs of the additional level in the multigrid hierarchy of this type led finally to longer computing times. The standard multigrid approach was for the Q_3/P_2^{disc} finite element discretization considerably slower than both types of the multiple discretization multilevel method.

Table VII. Q_3/P_2^{disc}, multigrid methods as preconditioner in FGMRES.

		Fixed point iteration (3)				Newton's method (4)			
Prec.	Cycle	ite	t(sol)	t(tot)	%	ite	t(sol)	t(tot)	%
mg	2lev(1,1)	1533	116 890	164 359	71	1566	169 042	347 478	48
mg	2lev(2,2)	869	136 926	183 617	74	867	203 801	378 166	53
mg	2lev(3,3)	700	157 705	202 763	77	713	239 831	409 824	58
mdml	V(1,1)	1597	76 671	*118 979*	64	1601	109 661	*267 474*	40
mdml	V(2,2)	813	77 274	*117 291*	65	818	107 756	*254 027*	42
mdml	V(3,3)	723	102 434	144 316	70	724	136 686	290 409	47
mdml	F(1,1)	13 627	662 765	832 519	79	Slow	—	—	—
mdml	F(2,2)	1173	111 792	155 264	72	1059	142 013	302 136	47
mdml	F(3,3)	725	102 980	144 656	71	715	143 994	299 763	48
mdml	W(1,1)	13 627	665 686	835 680	79	8431	587 006	1 000 750	58
mdml	W(2,2)	1173	112 004	155 556	72	1059	145 155	307 160	47
mdml	W(3,3)	725	102 526	144 127	71	715	144 793	300 989	48
mdml2	V(1,1)	1391	82 389	*129 556*	63	1339	104 483	*280 108*	37
mdml2	V(2,2)	860	99 988	146 700	68	839	127 798	298 488	42
mdml2	V(3,3)	701	121 170	165 826	73	702	159 265	325 834	48
mdml2	F(1,1)	1472	106 282	153 255	69	1404	137 576	318 507	43
mdml2	F(2,2)	851	121 259	167 729	72	838	161 459	337 029	47
mdml2	F(3,3)	702	148 764	193 473	76	694	196 543	365 383	53
mdml2	W(1,1)	2352	171 260	221 784	77	2385	233 047	421 700	55
mdml2	W(2,2)	834	121 016	168 160	71	826	159 321	334 437	47
mdml2	W(3,3)	702	153 410	198 965	77	703	198 795	366 770	54

Table VIII. Discretizations and d.o.f. for the computations with the P_2/P_1 finite element.

	Standard mg		mdml, type 1	
Level	Disc	d.o.f.	Disc	d.o.f.
3			P_2/P_1	810 160
2	P_2/P_1	810 160	P_1^{nc}/P_0	1 309 440
1	P_2/P_1	106 680	P_1^{nc}/P_0	166 080
0	P_2/P_1	14 740	P_1^{nc}/P_0	21 360

5.3. Computations with the Taylor–Hood finite element P_2/P_1 on a tetrahedral grid

The second-order Taylor–Hood finite element was applied on a tetrahedral grid which was obtained after two refinements of the initial grid, see Table VIII for information on the degrees of freedom. A damping was not applied in the multigrid methods.

Table IX presents the results of the computational studies. Again, the simulations were much faster if the fixed point iteration (3) was used compared to Newton's method (4). With respect to the multigrid approaches, the multiple discretization multilevel method was for each type of multigrid cycle considerably more efficient than the standard multigrid approach.

Table IX. P_2/P_1, multigrid methods as preconditioner in FGMRES.

		Fixed point iteration (3)				Newton's method (4)			
Prec.	Cycle	ite	t(sol)	t(tot)	%	ite	t(sol)	t(tot)	%
mg	V(1,1)	780	112 520	*131 108*	85	772	138 247	*210 479*	65
mg	V(2,2)	752	207 991	226 556	91	752	279 555	353 340	79
mg	F(1,1)	774	156 139	174 980	89	776	190 195	262 246	72
mg	F(2,2)	763	282 509	301 123	93	753	341 287	413 595	82
mg	W(1,1)	774	155 090	173 681	89	776	192 108	264 751	72
mg	W(2,2)	763	283 564	302 232	93	753	347 775	420 656	82
mdml	V(1,1)	764	92 379	*113 432*	81	763	116 161	*194 134*	59
mdml	V(2,2)	754	171 265	192 294	89	755	218 143	295 960	73
mdml	F(1,1)	761	108 566	*130 001*	83	763	133 411	*211 034*	63
mdml	F(2,2)	754	193 378	214 292	90	762	246 659	324 150	76
mdml	W(1,1)	762	111 058	132 102	84	763	142 647	221 240	64
mdml	W(2,2)	754	201 288	222 370	90	760	257 596	335 768	76

Table X. Discretizations and d.o.f. for the computations with the P_3/P_2 finite element.

	Standard mg		mdml, type 1		mdml, type 2	
Level	Disc	d.o.f.	Disc	d.o.f.	Disc	d.o.f.
3					P_3/P_2	367 000
2			P_3/P_2	367 000	P_2/P_1	106 680
1	P_3/P_2	367 000	P_1^{nc}/P_0	166 080	P_1^{nc}/P_0	166 080
0	P_3/P_2	49 140	P_1^{nc}/P_0	21 360	P_1^{nc}/P_0	21 360

5.4. Computations with the Taylor–Hood finite element P_3/P_2 on a tetrahedral grid

The numbers of d.o.f. for the Taylor–Hood pair of finite element spaces with third-order velocity and second-order pressure are presented in Table X. In type 2 of the multiple discretization multilevel method, we used the P_2/P_1 finite element discretization on the multigrid level 2. In these computations, the damping factor in the pressure node oriented Vanka smoother was set to $\omega_l = 0.8$ on all levels. Also, the updates after the prolongations were damped with $\gamma_l = 0.8$ on all levels.

Table XI shows that the standard multigrid approach fails completely for this discretization. We found that much smaller damping factors are necessary for its convergence. However, the computing times were much longer than those in Table XI obtained with both types of the multiple discretization multilevel method.

The fixed point iteration (3) was for the P_3/P_2 finite element discretization also more efficient than Newton's method in terms of computing time. However, the superiority was not as large as for the other discretizations. For the higher-order P_3/P_2 finite element discretization, the part of the solver time on the total computing time is considerably larger than for the other discretizations considered in the numerical studies. Similar to the Q_3/P_2^{disc} finite element discretizations, type 2 of the multiple discretization multilevel method was in general somewhat slower than type 1.

Table XI. P_3/P_2, multigrid methods as preconditioner in FGMRES.

		Fixed point iteration (3)				Newton's method (4)			
Prec.	Cycle	ite	t(sol)	t(tot)	%	ite	t(sol)	t(tot)	%
mg	2lev(1,1)	Div.	—	—	—	Div.	—	—	—
mg	2lev(2,2)	Div.	—	—	—	Div.	—	—	—
mdml	V(1,1)	798	276 837	*290 545*	95	753	312 189	*364 314*	85
mdml	V(2,2)	773	525 908	539 657	97	747	600 546	652 481	92
mdml	F(1,1)	790	279 578	*293 253*	95	753	318 737	*370 825*	85
mdml	F(2,2)	819	565 027	578 803	97	749	615 480	667 653	92
mdml	W(1,1)	790	279 422	*293 094*	95	753	317 661	*369 835*	85
mdml	W(2,2)	819	566 809	580 581	97	749	614 295	666 433	92
mdml2	V(1,1)	816	294 634	*309 800*	95	743	319 695	*377 961*	84
mdml2	V(2,2)	922	654 643	669 862	97	793	655 576	713 054	91
mdml2	F(1,1)	858	333 135	348 313	95	744	351 384	409 787	85
mdml2	F(2,2)	898	675 203	690 378	97	800	710 920	768 363	92
mdml2	W(1,1)	829	329 246	344 369	95	744	356 640	414 983	85
mdml2	W(2,2)	892	680 011	695 267	97	802	723 984	781 520	92

6. SUMMARY

The paper studied the efficiency of two approaches for linearizing the incompressible Navier–Stokes equations and of two multigrid approaches being used in the solution of the arising linear algebraic saddle point problems for the simulation of a time-dependent laminar flow around a three-dimensional cylinder. The equations were discretized in time by the Crank-Nicolson scheme and in space by inf–sup stable higher-order Galerkin finite element discretizations.

The fixed point iteration (3) was always much more efficient than Newton's method (4). Using all multigrid methods as preconditioner in FGMRES was much faster and more robust than using them as solver. The multiple discretization multilevel method type 1 was in general the best approach. In particular for Taylor–Hood finite elements, it was considerably better than the standard multigrid method. Only for the Q_2/P_1^{disc} finite element, the efficiency of both multigrid approaches was similar.

REFERENCES

1. Logashenko D, Fischer T, Motz S, Gilles E-D, Wittum G. Simulation of crystal growth in a stirred tank. *Computing and Visualization in Science* 2005, in press.
2. Öncül A, Sundmacher K, Thévenin D. Numerical investigation of the influence of the activity coefficient on barium sulphate crystallization. *Chemical Engineering Science* 2005; **60**(19):5395–5405.
3. Krewer U, Song Y, Sundmacher K, John V, Lübke R, Matthies G, Tobiska L. Direct methanol fuel cell (DMFC): analysis of residence time behaviour of anodic flow bed. *Chemical Engineering Science* 2004; **59**:119–130.
4. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method*. Wiley: Chichester, 2000.
5. Glowinski R. Finite element methods for incompressible viscous flow. In *Numerical Methods for Fluids* (*Part 3*), Ciarlet PG *et al.* (eds), Handbook of Numerical Analysis, vol. IX. North-Holland: Amsterdam, 2003; 3–1176.
6. Hughes TJ, Mazzei L, Jansen KE. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science* 2000; **3**:47–59.
7. Sagaut P. *Large Eddy Simulation for Incompressible Flows* (2nd edn). Springer: Berlin, Heidelberg, New York, 2003.

8. Schäfer M, Turek S. The benchmark problem 'Flow around a cylinder'. In *Flow Simulation with High-Performance Computers II*, Hirschel EH (ed.), Notes on Numerical Fluid Mechanics, vol. 52. Vieweg: Braunschweig, 1996; 547–566.
9. Turek S. *Efficient Solvers for Incompressible Flow Problems*: *An Algorithmic and Computational Approach*. Lecture Notes in Computer Science and Engineering, vol. 6. Springer: New York, 1999.
10. John V. Reference values for drag and lift of a two-dimensional time dependent flow around a cylinder. *International Journal for Numerical Methods in Fluids* 2004; **44**:777–788.
11. John V, Matthies G, Rang J. A comparison of time-discretization/linearization approaches for the time-dependent incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2005, in press.
12. Girault V, Raviart P-A. *Finite Element Methods for Navier–Stokes Equations*. Springer: Berlin, Heidelberg, New York, 1986.
13. John V, Matthies G. Higher order finite element discretizations in a benchmark problem for incompressible flows. *International Journal for Numerical Methods in Fluids* 2001; **37**:885–903.
14. John V. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2002; **40**:775–798.
15. John V, Knobloch P. On non-nested multilevel solvers for the Stokes and Navier–Stokes equations. *Proceedings of the 18th GAMM—Seminar*, Leipzig, 2002.
16. Braack M, Richter T. Solutions of 3D Navier–Stokes benchmark problems with adaptive finite elements. *Computers and Fluids* 2005, in press.
17. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. In *Acta Numerica*, Iserles A (ed.). Cambridge University Press: Cambridge, 2005; 1–137.
18. John V. A comparison of parallel solvers for the incompressible Navier–Stokes equations. *Computing and Visualization in Science* 1999; **4**(1):193–200.
19. John V. On the performance of smoothers in coupled multigrid methods for the solution of the incompressible Navier–Stokes equations on parallel computers. In *Advances in Fluid Mechanics III*, Rahman M, Brebbia CA (eds). WIT Press: 2000; 181–190.
20. Saad Y. A flexible inner–outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* 1993; **14**(2):461–469.
21. Saad Y. *Iterative Methods for Sparse Linear Systems* (2nd edn). SIAM: Philadelphia, PA, 2003.
22. John V, Knobloch P, Matthies G, Tobiska L. Non-nested multi-level solvers for finite element discretizations of mixed problems. *Computing* 2002; **68**:313–341.
23. John V. *Large Eddy Simulation of Turbulent Incompressible Flows. Analytical and Numerical Results for a Class of LES Models*. Lecture Notes in Computer Science and Engineering, vol. 34. Springer: Berlin, Heidelberg, New York, 2004.
24. Schieweck F. A general transfer operator for arbitrary finite element spaces. Preprint 00-25, Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg, 2000.
25. Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**(3):856–869.
26. Crouzeix M, Raviart P-A. Conforming and nonconforming finite element methods for solving the stationary Stokes equations. I. *RAIRO Analyse Numerique* 1973; **7**:33–76.
27. Rannacher R, Turek S. Simple nonconforming quadrilateral Stokes element. *Numerical Methods for Partial Differential Equations* 1992; **8**:97–111.
28. Roos H-G, Stynes M, Tobiska L. *Numerical Methods for Singularly Perturbed Differential Equations*. Springer: Berlin, 1996.
29. John V, Matthies G. MooNMD— a program package based on mapped finite element methods. *Computing and Visualization in Science* 2004; **6**:163–170.