INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:775–798 (DOI: 10.1002/fld.377)

Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations

Volker John*,†

Otto–von–Guericke–Universität Magdeburg, Institut für Analysis und Numerik, Postfach 4120, 39016 Magdeburg, Germany

SUMMARY

This paper presents a numerical study of the 3D flow around a cylinder which was defined as a benchmark problem for the steady state Navier–Stokes equations within the DFG high-priority research program *flow simulation with high-performance computers* by Schäfer and Turek (Vol. 52, Vieweg: Braunschweig, 1996). The first part of the study is a comparison of several finite element discretizations with respect to the accuracy of the computed benchmark parameters. It turns out that boundary fitted higher order finite element methods are in general most accurate. Our numerical study improves the hitherto existing reference values for the benchmark parameters considerably. The second part of the study deals with efficient and robust solvers for the discrete saddle point problems. All considered solvers are based on coupled multigrid methods. The flexible GMRES method with a multiple discretization multigrid method proves to be the best solver. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: incompressible Navier–Stokes equations; higher order finite element methods; coupled multigrid methods; multiple discretization multigrid method; flexible GMRES

1. INTRODUCTION

A systematic study of discretizations and solvers for the incompressible Navier–Stokes equations was started within the DFG high-priority research program *flow simulation with high-performance computers*. A number of benchmark problems describing flows around obstacles were defined by Schäfer and Turek [1]. The benchmark parameters which should be computed, e.g. drag and lift coefficients at the obstacles, are of interest in applications.

A numerical study of the benchmark problem which defines a steady state two-dimensional (2D) flow around a cylinder can be found in Reference [2]. In this study, the most accurate results could be computed with isoparametric higher order finite element discretizations. However, the solution of the arising discrete saddle point problems caused some difficulties. The remedy was the use of a multigrid method which is based on different discretizations

* Correspondence to: V. John, Institut für Analysis und Numerik, Otto–von–Guericke–Universität Magdeburg, Postfach 4120, 39016 Magdeburg, Germany.
† E-mail: john@mathematik.uni-magdeburg.de

Received September 2001
Revised April 2002

on different grid levels—the multiple discretization multigrid method. A comparison of this method with other solvers was not presented in Reference [2].

The present paper can be considered as a continuation of paper [2]. We study a three-dimensional (3D) steady-state flow around a cylinder. The main aspects of our numerical study are the accuracy of discretizations and the efficiency and robustness of solvers.

Concerning the first aspect, boundary fitted higher order finite element discretizations turned out to compute the most accurate benchmark coefficients. This could be expected in view of the results of the 2D study. The more important outcome is a considerable improvement of the reference values for the benchmark parameters in comparison to those given in Reference [1]. The new reference values, presented in Section 5, may serve for a better evaluation of numerical techniques which are used in solving the 3D Navier–Stokes equations.

All solvers of the arising saddle point problems which are considered here are based on coupled multigrid methods. Besides using the standard multigrid approach and the multiple discretization multigrid method as solvers, we apply them also as preconditioners in the flexible GMRES (FGMRES) method by Saad [3]. For the saddle point problems arising from higher order finite element discretizations, the FGMRES method with multiple discretization multigrid method turns out to be the best solver.

2. THE BENCHMARK PROBLEM OF THE 3D FLOW AROUND A CYLINDER

We consider the flow around a cylinder which was defined by Schäfer and Turek [1] as a benchmark problem within the DFG high-priority research program *flow simulation with high-performance computers*. The flow is governed by the steady state incompressible Navier–Stokes equations

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

The domain Ω is the channel with a cylinder shown in Figure 1. The boundary of Ω is denoted by $\partial\Omega$ and the outer normal by $\mathbf{n}$. The height of the channel is $H = 0.41$ m and the diameter of the cylinder $D = 0.1$ m. The right-hand side of the momentum equation vanishes in the benchmark problem, $\mathbf{f} = \mathbf{0}$. The kinematic viscosity of the fluid is given by $\nu = 10^{-3}\,\mathrm{m}^2/\mathrm{s}$ and its density by $\rho = 1\,\mathrm{kg/m}^3$. The inflow is prescribed by

$$\mathbf{g} = \begin{pmatrix} 16Uyz(H-y)(H-z)/H^4 \\ 0 \\ 0 \end{pmatrix}$$

with $U = 0.45$ m/s. The flow has the Reynolds number 20 based on ν, D and the mean inflow $\bar{U} = 0.2$ m/s.

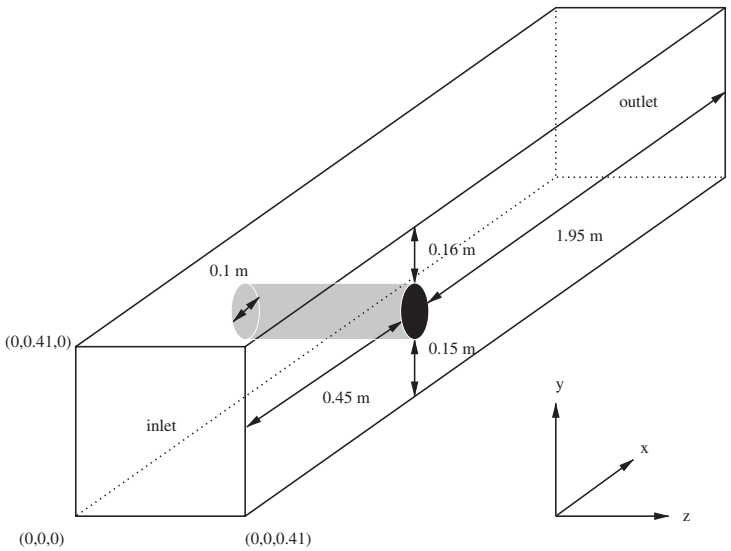

Figure 1. The channel with the cylinder.

The benchmark coefficients to compute are the drag and the lift coefficient at the cylinder and the pressure difference Δp between the point $(0.45, 0.2, 0.205)$ at the front of the cylinder and the point $(0.55, 0.2, 0.205)$ at the back of the cylinder. Let S be the surface of the cylinder, $\mathbf{n}_S$ be the inward pointing unit normal with respect to Ω and τ_1, τ_2 tangential vectors given by

$$\mathbf{n}_S = \begin{pmatrix} n_x \\ n_y \\ 0 \end{pmatrix}, \quad \tau_1 = \begin{pmatrix} n_y \\ -n_x \\ 0 \end{pmatrix}, \quad \tau_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{2}$$

The projection of the velocity into the direction of τ_1 is denoted by $\mathbf{u}_{\tau_1} = \mathbf{u} \cdot \tau_1$. With the drag force F_d and the lift force F_ℓ

$$F_\mathrm{d} = \int_S \left(\rho\nu \frac{\partial \mathbf{u}_{\tau_1}}{\partial \mathbf{n}_S} n_y - p n_x \right) \mathrm{d}s, \qquad F_\ell = -\int_S \left(\rho\nu \frac{\partial \mathbf{u}_{\tau_1}}{\partial \mathbf{n}_S} n_x + p n_y \right) \mathrm{d}s \tag{3}$$

the drag coefficient c_d and the lift coefficient c_ℓ are defined by

$$c_\mathrm{d} = \frac{2F_\mathrm{d}}{\rho \bar{U}^2 DH}, \qquad c_\ell = \frac{2F_\ell}{\rho \bar{U}^2 DH} \tag{4}$$

This gives in the present example

$$c_\mathrm{d} = \frac{500}{0.41} F_\mathrm{d}, \qquad c_\ell = \frac{500}{0.41} F_\ell$$

The computation of the surface integrals in (3) can be replaced by a computation of volume integrals which is in general preferable from the numerical point of view. We will derive

the volume integral formula for c_d. A straightforward computation gives for the vectors of form (2) and $\mathbf{u} = (u_1, u_2, u_3)$

$$\frac{\partial \mathbf{u}_{\tau_1}}{\partial \mathbf{n}_S} n_y = \left(\frac{\partial u_1}{\partial x} n_x n_y + \frac{\partial u_1}{\partial y} n_y^2 - \frac{\partial u_2}{\partial x} n_x^2 - \frac{\partial u_2}{\partial y} n_x n_y \right) n_y \tag{5}$$

From $\mathbf{u}|_S \equiv \mathbf{0}$ follows $\partial \mathbf{u}/\partial \tau_1 = \partial \mathbf{u}/\partial \tau_2 = \mathbf{0}$ on S, in particular

$$\frac{\partial u_1}{\partial x} n_y - \frac{\partial u_1}{\partial y} n_x = 0, \quad \frac{\partial u_2}{\partial x} n_y - \frac{\partial u_2}{\partial y} n_x = 0, \quad \frac{\partial u_3}{\partial z} = 0 \tag{6}$$

Substituting (6) into (5), noting $n_x^2 + n_y^2 = 1$ and using $\nabla \cdot \mathbf{u} = 0$ gives

$$\begin{aligned} \frac{\partial \mathbf{u}_{\tau_1}}{\partial \mathbf{n}_S} n_y &= \left(\frac{\partial u_1}{\partial y} - \frac{\partial u_2}{\partial x} \right) n_y = \frac{\partial u_1}{\partial y} n_y - \frac{\partial u_2}{\partial y} n_x = \frac{\partial u_1}{\partial x} n_x + \frac{\partial u_1}{\partial y} n_y + \frac{\partial u_3}{\partial z} n_x \\ &= \frac{\partial u_1}{\partial x} n_x + \frac{\partial u_1}{\partial y} n_y \end{aligned} \tag{7}$$

The volume integral is derived from the momentum equation of the Navier–Stokes equations. We consider a function $\mathbf{v}_d \in (H^1(\Omega))^3$, where $H^1(\Omega)$ is the standard Sobolev space, with

$$\mathbf{v}_d|_S = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{v}_d|_{\partial\Omega \setminus S} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

The momentum equation is multiplied with $\mathbf{v}_d$, integrated on Ω and integration by parts is applied. One obtains, using the special form of the normal $\mathbf{n}_S$ pointing into Ω

$$-[(\nu \nabla \mathbf{u}, \nabla \mathbf{v}_d) + ((\mathbf{u} \cdot \nabla)\mathbf{u}, \mathbf{v}_d) - (p, \nabla \cdot \mathbf{v}_d) - (\mathbf{f}, \mathbf{v}_d)] = \int_S \nu \left(\frac{\partial u_1}{\partial x} n_x + \frac{\partial u_1}{\partial y} n_y \right) - p n_x \, \mathrm{d}s \tag{9}$$

where $(\cdot, \cdot)$ denotes the inner product in $(L^2(\Omega))^d, d \geqslant 1$. Combining (3), (7) and (9) for $\rho = 1 \, \mathrm{kg/m^3}$, leads to the volume integral formula for computing the drag coefficient

$$c_d = -\frac{2}{\bar{U}^2 D H} [(\nu \nabla \mathbf{u}, \nabla \mathbf{v}_d) + ((\mathbf{u} \cdot \nabla)\mathbf{u}, \mathbf{v}_d) - (p, \nabla \cdot \mathbf{v}_d) - (\mathbf{f}, \mathbf{v}_d)] \tag{10}$$

Similar considerations with a function $\mathbf{v}_\ell \in (H^1(\Omega))^3$ with

$$\mathbf{v}_\ell|_S = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{v}_\ell|_{\partial\Omega \setminus S} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{11}$$

give a volume integral formula for the lift coefficient c_ℓ

$$c_\ell = -\frac{2}{\bar{U}^2 D H} [(\nu \nabla \mathbf{u}, \nabla \mathbf{v}_\ell) + ((\mathbf{u} \cdot \nabla)\mathbf{u}, \mathbf{v}_\ell) - (p, \nabla \cdot \mathbf{v}_\ell) - (\mathbf{f}, \mathbf{v}_\ell)] \tag{12}$$

Formulae (10) and (12) were used in all computations instead of the surface integral formulae.

Formulae (4), (10) for c_d and (4), (12) for c_ℓ are equivalent for the solution $(\mathbf{u}, p)$ of the continuous problem. The property $\nabla \cdot \mathbf{u} = 0$ was used in the derivation of (10) and (12). However, one obtains in general different results if the drag (lift) coefficient for a finite element solution $(\mathbf{u}^h, p^h)$ is computed with (4) or (10) (or (12)). A study of several possibilities of computing these values in a 2D Navier–Stokes problem [4] shows that the volume integral formulae give more accurate results than the surface integral formulae.

The functions $\mathbf{v}_d$ in (8) and $\mathbf{v}_\ell$ in (11) are prescribed only on the boundary of Ω. The form of $\mathbf{v}_d, \mathbf{v}_\ell$ in Ω is arbitrary as long as both functions belong to $(H^1(\Omega))^3$. To minimize the cost of computing the volume integrals in (10) and (12), both functions should have a small support around the obstacle. Let V^h be the conforming finite element space of the velocity. We extend V^h to a finite element space $\tilde{V}^h$ by all functions with the same polynomial degree which possess in addition degrees of freedom on the closure $\bar{S}$ of S. Then, we choose $\mathbf{v}_d, \mathbf{v}_\ell \in \tilde{V}^h$ such that for the nodes $\mathbf{x}$

$$\mathbf{v}_d = \begin{cases} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \text{if } \mathbf{x} \in \bar{S} \\ \mathbf{0} & \text{if } \mathbf{x} \notin \bar{S} \end{cases}, \quad \mathbf{v}_\ell = \begin{cases} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & \text{if } \mathbf{x} \in \bar{S} \\ \mathbf{0} & \text{if } \mathbf{x} \notin \bar{S} \end{cases} \tag{13}$$

Thus, the volume integrals need to be computed only in one layer of mesh cells around the obstacle and only finite element functions have to be integrated.

If V^h is a non-conforming finite element space, we define $\mathbf{v}_d$ and $\mathbf{v}_\ell$ also as given in (13). Then, both functions do not belong to $(H^1(\Omega))^3$ in general, but from the 2D benchmark problem, we have the experience that the computed benchmark parameters do not differ much compared to the benchmark coefficients obtained with conforming test functions, see Reference [4]. For discontinuous discrete pressure approximations, $p^h(0.45, 0.2, 0.205)$ and $p^h(0.55, 0.2, 0.205)$ are computed by averaging p^h in these points.

3. THE FINITE ELEMENT DISCRETIZATIONS

In this section, the finite element spaces and discretizations are described which were compared in the numerical tests.

The Navier–Stokes equations are linearized by a fixed point iteration. Let $(\mathbf{u}^n, p^n)$ be the current iterate, then the next iterate is computed by solving

$$\begin{aligned} -\nu \Delta \mathbf{u}^{n+1} + (\mathbf{u}^n \cdot \nabla)\mathbf{u}^{n+1} + \nabla p^{n+1} &= \mathbf{0} \text{ in } \Omega \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \text{ in } \Omega \end{aligned} \tag{14}$$

where $\mathbf{u}^{n+1}$ fulfils the same boundary conditions as $\mathbf{u}$. The linear saddle point problem (14) is discretized by a finite element method.

Let $\mathcal{T}_h$ be a decomposition of Ω into either hexahedra or tetrahedra. We will use so-called mapped finite elements, i.e. all finite elements are defined first on a reference cell $\hat{K}$ and the finite elements on an arbitrary mesh cell K are defined with the help of the reference map to the reference cell.

In the following, we will consider only such pairs of finite element spaces V_h, Q_h, which fulfil the inf–sup or Babuška–Brezzi condition, i.e. there exists a constant $\beta > 0$ independent of the triangulation $\mathscr{T}_h$ such that

$$\inf_{q_h \in Q_h} \sup_{v_h \in V_h} \frac{(\nabla_h \cdot v_h, q_h)}{|v_h|_1 \, \|q_h\|_0} \geqslant \beta \tag{15}$$

where $\|\cdot\|_0$ and $|\cdot|_1$ denote the norm in $L^2(\Omega)$ and the seminorm in $(H^1(\Omega))^3$, respectively. In the case of small data, (15) together with the coerciveness of the discrete bilinear form $(\nabla_h \cdot, \nabla_h \cdot)$ guarantee the unique solvability of the arising discrete problem.

The cube $(-1,1)^3$ is used as reference hexahedron $\hat{K}$. The reference transformation from the closure of $\hat{K}$ onto the closure of a mesh cell K is denoted by F_K. In general, the mapping F_K is trilinear. We denote by $Q_k(\hat{K})$ and $P_k(\hat{K})$ the following sets of polynomials on $\hat{K}$:

$$Q_k(\hat{K}) := \left\{ \hat{q} = \sum_{i,j,l=0}^{k} a_{ijl} \hat{x}^i \hat{y}^j \hat{z}^l \right\}, \quad P_k(\hat{K}) := \left\{ \hat{p} = \sum_{i,j,l=0}^{i+j+l \leqslant k} b_{ijl} \hat{x}^i \hat{y}^j \hat{z}^l \right\}$$

The spaces on an arbitrary mesh cell K are given by

$$Q_k(K) := \{ q = \hat{q} \circ F_K^{-1} \colon \ \hat{q} \in Q_k(\hat{K}) \}, \quad P_k(K) := \{ p = \hat{p} \circ F_K^{-1} \colon \ \hat{p} \in P_k(\hat{K}) \}$$

and the global finite element spaces by

$$\begin{aligned} Q_k &:= \{ v \in H^1(\Omega) \colon \ v|_K \in Q_k(K) \}, \quad k \geqslant 1 \\ Q_0 &:= \{ v \in L^2(\Omega) \colon \ v|_K \in Q_0(K) \} \\ P_k^{\text{disc}} &:= \{ v \in L^2(\Omega) \colon \ v|_k \in P_k(K) \}, \quad k \geqslant 1 \end{aligned}$$

In addition, let Q_1^{rot} be the space of non-conforming, point-value oriented, rotated bilinears defined and analysed by Rannacher and Turek [5] and Schieweck [6]. In our numerical tests, we use the following inf–sup stable pairs of finite element spaces on hexahedral grids:

- Q_1^{rot}/Q_0, Q_2/P_1^{disc}, Q_2/Q_1, Q_3/P_2^{disc}

As commonly done, the fact that the velocity space is a vector-valued function is not indicated in these notations. For defining tetrahedral finite elements, the reference tetrahedron with the vertices $(0,0,0)$, $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ is used. In the case of tetrahedra, the reference map is affine. Using the spaces on the mesh cells given above, we define

$$\begin{aligned} P_0 &:= \{ v \in L^2(\Omega) \colon \ v|_K \in P_0(K) \} \\ P_k &:= \{ v \in H^1(\Omega) \colon \ v|_K \in P_k(K) \}, \quad k \geqslant 1 \end{aligned}$$

Furthermore, let P_1^{nc} the space of non-conforming, piecewise linears which are continuous at the midpoint of each face, the so-called Crouzeix–Raviart finite element space [7]. We will use the following inf–sup stable pairs of finite element spaces on tetrahedral grids:

- P_1^{nc}/P_0, P_2/P_1, P_3/P_2

Since the domain Ω has a non-polyhedral boundary, the cylinder, the application of boundary fitted finite elements seems to be advantageous. The numerical study of the 2D channel flow around a cylinder presented in Reference [2] shows a great improvement in the accuracy using isoparametric finite elements compared to a polygonal approximation of the boundary. In order to obtain a better approximation of the boundary for the 3D flow around a cylinder, all degrees of freedom which belong to the curved face S were moved from the plane face of a mesh cell onto S. We will refer to this technique as boundary fitted finite elements. This technique was applied to the Q_2/P_1^{disc}, Q_2/Q_1, Q_3/P_2^{disc}, P_2/P_1 and P_3/P_2 pairs of finite element spaces.

The numerical study of the 2D flow around a cylinder in Reference [2] shows that the standard Galerkin discretization gives the most accurate results for higher order finite elements. A stabilization of the convective term was not necessary. The situation is somewhat different for non-conforming finite elements of lowest order. That is why, we will present also results for the Samarskij upwind discretization. Upwind stabilizations are studied analytically by Schieweck and Tobiska [8] for the Crouzeix–Raviart finite element and by Schieweck [6] for the Q_1^{rot}/Q_0 pair of finite element spaces.

4. THE SOLVER OF THE LINEAR SADDLE POINT PROBLEMS

Linearization (14) and discretization of the Navier–Stokes equations lead to a linear saddle point problem of the abstract form

$$\mathscr{A}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} A & B \\ C & 0 \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{16}$$

A system of this type has to be solved approximately in each step of the fixed point iteration.

We will study four different types of solvers for (16) which are based on multigrid approaches. We use so-called coupled multigrid methods, i.e. multigrid methods which solve (16) for both types of unknowns, the velocity u and the pressure p, together. In Section 4.1, we will present two types of multigrid approaches, the standard one and one approach which uses different discretizations on different levels of the multigrid hierarchy. In the numerical tests, these two multigrid methods are applied as solvers for (16) and as preconditioner within a flexible GMRES (FGMRES) method.

This section describes the individual components of the solvers. A multigrid method is defined by

- the grid hierarchy,
- the grid transfer operators (restriction and prolongation),
- the smoother on finer levels,
- the coarse grid solver.

We will also mention the parameters of the solvers, which are not varied in the numerical tests.

4.1. The standard and the multiple discretization multigrid approach

We have used two different multigrid approaches for the solution of (16), see Figure 2. In the standard multigrid approach, the number of geometric grid levels and the number of

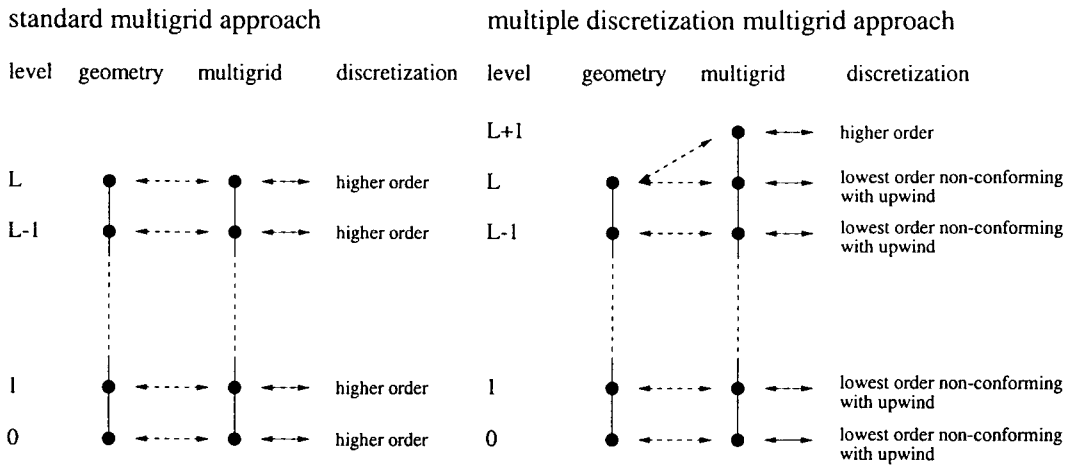

Figure 2. The standard and the multiple discretization multigrid approach for higher order discretizations.

levels in the multigrid hierarchy coincide. The same discretization is used on each multigrid level. In a numerical study of a benchmark problem for the Navier–Stokes equations in 2D [2] great difficulties are reported with the standard multigrid approach for solving the linear saddle point problems arising in some higher order finite element discretizations. However, the higher order finite element discretizations have given very accurate results for the benchmark reference values. In contrast, lowest order non-conforming discretization with upwind stabilization were rather inaccurate but the standard multigrid approach has been proven as a very efficient solver, see also Reference [9]. This situation led to the idea of constructing a multigrid method for higher order finite element discretizations which is based on a stable lowest order non-conforming finite element discretization. We will call this approach the multiple discretization multigrid method. There are of course many possibilities to construct such a multigrid method. We use in our tests the approach depicted in Figure 2. In this approach, the multigrid hierarchy possesses one level more than the geometric grid hierarchy. On the finest geometric grid, level L, two discretizations are applied. One of them, which forms the finest level of the multigrid hierarchy, is the discretization which we are interested in, e.g. a higher order discretization. The second discretization on the geometric level L is a lowest order non-conforming discretization with upwind. On all coarser geometric levels, also a stable lowest order non-conforming discretization is applied.

In the numerical study of the benchmark problem of the Navier–Stokes equations in 2D [2] it is mentioned that the multiple discretization multigrid approach works for higher order finite element discretizations in principal and the solutions could be obtained in reasonable computing times. A comparison of both multigrid approaches was not presented.

A convergence analysis of the multiple discretization multigrid approach applied to the Stokes equations can be found in Reference [10]. The analysis is done for smoothers of Braess–Sarazin-type [11].

4.2. The grid transfer operators

In the multiple discretization multigrid approach, the data have to be transferred between different finite element spaces on the multigrid levels L and $L+1$. We use a grid transfer operator by Schieweck [12]. This operator allows a transfer between almost arbitrary finite element spaces. Its application is given by an appropriate local averaging. For a detailed description of the construction and the analytical properties of this operator, we refer to References [12, 10]. A short description is contained also in Reference [2].

In the multigrid approach, the prolongation of the update from level l can be damped and then will be added to the current iterate on level $l+1$. In the numerical tests presented in this report, we use the damping factor 1.0 (no damping).

4.3. The Vanka-type smoothers

This section contains a detailed description of the smoothers which are used. In addition, we illustrate the increase of the numerical costs of these smoothers in three dimensions compared to two dimensions.

The coupled multigrid method is applied with local smoothers, the so-called Vanka-type smoothers, see Reference [13]. Vanka-type smoothers can be considered as block Gauss–Seidel methods. Let $\mathscr{V}^h$ and $\mathscr{Q}^h$ be the set of velocity and pressure degrees of freedom, respectively. These sets are decomposed into

$$\mathscr{V}^h = \bigcup_{j=1}^{J} \mathscr{V}_j^h, \quad \mathscr{Q}^h = \bigcup_{j=1}^{J} \mathscr{Q}_j^h \tag{17}$$

The subsets are not required to be disjoint.

Let $\mathscr{A}_j$ be the block of the matrix $\mathscr{A}$ which is connected with the degrees of freedom of $\mathscr{W}_j^h = \mathscr{V}_j^h \cup \mathscr{Q}_j^h$, i.e. the intersection of the rows and columns of $\mathscr{A}$ with the global indices belonging to $\mathscr{W}_j^h$,

$$\mathscr{A}_j = \begin{pmatrix} A_j & B_j \\ C_j & 0 \end{pmatrix} \in \mathbb{R}^{\dim(\mathscr{W}_j^h)\times\dim(\mathscr{W}_j^h)}$$

Similarly, we denote by $(\cdot)_j$ the restriction of a vector on the rows corresponding to the degrees of freedom in $\mathscr{W}_j^h$. Each smoothing step with a Vanka-type smoother consists in a loop over all sets $\mathscr{W}_j^h$, where for each $\mathscr{W}_j^h$ a local system of equations connected with the degrees of freedom in this set is solved. The local solutions are updated in a Gauss–Seidel manner. The (full) Vanka smoother computes new velocity and pressure values for the degrees of freedom in $\mathscr{W}_j^h$ by

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

The general strategy for choosing the sets $\mathscr{V}_j^h$ and $\mathscr{Q}_j^h$ is as follows. First, pick some pressure degrees of freedom which define $\mathscr{Q}_j^h$. Second, $\mathscr{V}_j^h$ is defined by all velocity degrees

Table I. Degrees of freedom for the local systems of the mesh cell oriented Vanka smoother (velocity: each component).

	2D			3D		
	Velocity	Pressure	Total	Velocity	Pressure	Total
Q_1^{nc}/Q_0	4	1	9	6	1	19
Q_2/P_1^{disc}	9	3	21	27	4	85
Q_3/P_2^{disc}	16	6	38	64	10	202
P_1^{nc}/P_0	3	1	7	4	1	13

of freedom which are connected with the pressure degrees of freedom from $\mathcal{Q}_j^h$ by non-zero entries in the matrix C.

We have applied two types of Vanka smoothers with respect to this strategy. The first one is called mesh cell oriented, because $\mathcal{Q}_j^h$ is defined by all pressure degrees of freedom which are connected to the mesh cell j. For this type of Vanka smoother, J coincides with the number of mesh cells. The mesh cell oriented Vanka smoother is applied only for discretizations with discontinuous pressure approximation, i.e. $Q^h \in \{P_0, Q_0, P_1^{\mathrm{disc}}, P_2^{\mathrm{disc}}\}$. For such discretizations, $\mathcal{V}_j^h$ consists of all velocity degrees of freedom which are connected to the mesh cell j. This property is not given for discretizations with continuous pressure approximation. For these discretizations, we use a decomposition in which $\mathcal{Q}_j^h$ is defined by a single pressure degree of freedom, $\dim \mathcal{Q}_j^h = 1$. This smoother is called pressure node oriented Vanka smoother. For this smoother, the number of subsets J in decomposition (17) is equal to the number of pressure degrees of freedom.

For a continuous pressure approximation, a pressure degree of freedom on a given mesh cell K is in general connected to velocity degrees of freedom on other mesh cells. In this case it is also possible to generate the local systems with the pressure degrees of freedom on K in order to perform a mesh cell oriented Vanka smoother. But in comparison to discontinuous pressure approximations, the overhead of searching all connections and allocating the local matrix and right-hand side is much higher.

Using discontinuous discrete pressure, the local matrix of the mesh cell oriented Vanka smoother can be generated on the current mesh cell. The size of the local systems is known *a priori* and it is given for different discretizations in Table I. It can be seen that the dimension of the local systems for the higher order discretizations in 3D is relatively large.

The size of the local systems for the pressure node oriented Vanka smoother applied in discretizations with continuous pressure approximation depends on the particular pressure degree of freedom and on the given grid. In addition, the size of the local systems cannot be bounded *a priori* since it depends on the maximal number of neighbour cells of K. A neighbour is a mesh cell K_1 with $\bar{K} \cap \bar{K}_1 \neq \emptyset$. To illustrate the size of the local systems, we give it for concrete degrees of freedom in typical situations (Figure 3) in Table II. It can be observed that these sizes are considerably larger than for the mesh cell oriented Vanka smoother for discretizations with discontinuous pressure (Table I). In addition, the number of local systems which must be solved in each smoothing step is in general a multiple compared to the mesh cell oriented Vanka smoother since the number of pressure degrees of freedom is in general larger than the number of mesh cells.

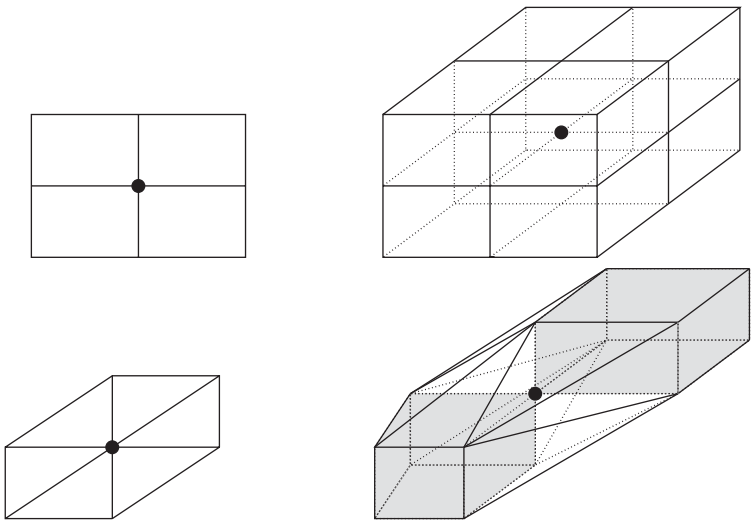

Figure 3. Degree of freedom for which the size of the local systems in the pressure node oriented Vanka smoother is given in Table II. (bottom right: 6 tetrahedra in two directions (coloured) and 2 tetrahedra in six directions, i.e. 24 tetrahedra are connected with this pressure degree of freedom).

Table II. Degrees of freedom for the local systems of the pressure node oriented Vanka smoother (velocity: each component).

	2D			3D		
	Velocity	Pressure	Total	Velocity	Pressure	Total
Q_2/Q_1	25	1	51	125	1	376
P_2/P_1	19	1	39	65	1	196
P_3/P_2	37	1	75	175	1	526

The size of the local systems using higher order finite element discretizations may be rather large, see Tables I and II. We have applied in the computations two approaches for their solution:

- Direct solution using the Gaussian elimination with column pivoting.
- Approximate solution using the GMRES method which stops after having reduced the Euclidean norm of the residual by a prescribed factor. In our numerical tests, the factor 10 is prescribed.

In both approaches, the solution of the large number of local systems is the most time-consuming part of the whole algorithm.

The mesh cell and the pressure node oriented Vanka smoother are equivalent for piecewise constant discrete pressure if the mesh cells and the pressure nodes are numbered in the same way.

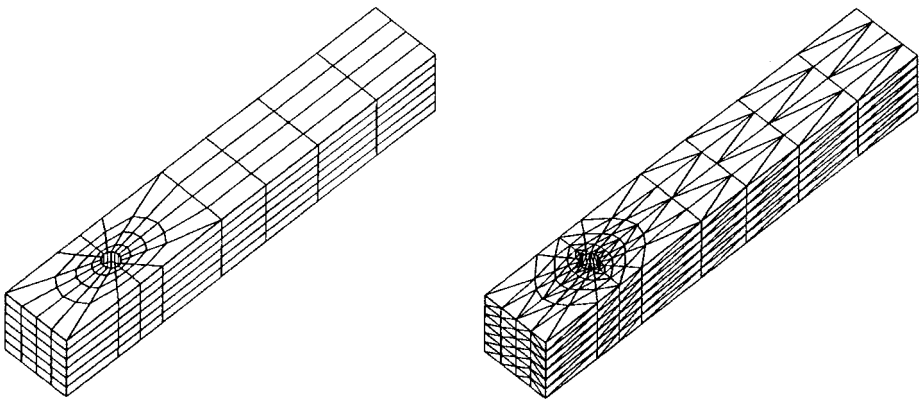

Figure 4. Coarse grids, level 0.

Since the Vanka-type smoothers are block Gauss–Seidel methods, their performance depends on the ordering of the degrees of freedom. We did not study this dependence in our tests. In the computations, we ordered the degrees of freedom, with possibly few exceptions, from the inflow to the outflow.

Sometimes it becomes necessary to damp the result of 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 computed by $(u_l, p_l) + \omega_l(\delta u_l, \delta p_l)$. The damping parameter can be chosen differently on all levels of the multigrid hierarchy.

4.4. The coarse grid solver

The systems on the coarsest grid, level 0, are solved approximately by the same iterative scheme which is used as smoother. The iteration is stopped either after having reduced the Euclidean norm of the residual by the factor 10 or after 100 iterations.

4.5. The flexible GMRES (FGMRES) method

Besides using the coupled multigrid methods described above as solver for systems of type (16), we applied them also as preconditioned in the right preconditioned flexible GMRES method by Saad [3, 14].

We will describe only the part of this approach which is connected with the application of the preconditioner. After the ith step of the FGMRES iteration, an $(i+1)$th dimensional Krylov space with basis $\{v_0, \ldots, v_i\}$, $\|v_j\| = 1$, $j = 0, \ldots, i$, is constructed. The application of the preconditioner M in the next FGMRES iteration looks as follows:

```
1. y_i := 0
2. (k := 0;k<prec_maxit; k++)
3.     solve Md_i = v_i - Ay_i
4.     update y_i := y_i + d_i
5. endfor
```


Table III. The benchmark coefficients computed with the boundary fitted Q_2/P_1^{disc} finite element discretization and the derived reference values.

Level l	d.o.f.	c_{d}	Order	c_ℓ	Order	Δp
1	117 360	6.1652287		1.0098526e − 2		1.7092581e − 1
2	899 040	6.1835048		9.4548796e − 3		1.7012569e − 1
3	7 035 840	6.1852335	3.40	9.4047889e − 3	3.68	1.7040289e − 1
4	55 666 560	6.1853267	4.21	9.4012217e − 3	3.81	1.7077855e − 1
4th order extrapolation		6.1853329		9.4009839e − 3		

In our case, M^{-1} represents one coupled multigrid cycle and A is the system matrix of (16). The parameter `prec_maxit` controls the number of multigrid cycles applied for preconditioning. In all computations, we use `prec_maxit=1`. The result of the application of the preconditioner is a vector y_i which is needed for the computation of the final result of the FGMRES iteration.

In lines 2–5, a system $Ay_i = v_i$ is solved iteratively with the help of the preconditioner. The result of this iteration depends of course on the initial iterate, which is set in line 1. We use in all computations the zero vector as initial iterate. We think that this choice is not optimal since the norm of the right-hand side v_i is one, and thus, the solution of $Ay_i = v_i$ cannot be expected to be close to the zero vector. But we are not aware of any studies of the topic how to choose the initial iterate. We use the zero vector just for simplicity and expect that a clever choice of the initial iterate will enhance the performance of the method.

5. THE ACCURACY OF THE COMPUTED BENCHMARK COEFFICIENTS

The benchmark coefficients which are computed with the different finite element spaces are compared in this section. In Reference [1], reference intervals are defined: $c_{\text{d,ref}} \in [6.05, 6.25]$, $c_{\text{l,ref}} \in [0.008, 0.01], \Delta p_{\text{ref}} \in [0.165, 0.175]$. We will first derive reference values from our numerical results to make the comparison more transparent.

The non-linear systems are solved in these tests up to an Euclidean norm of the residual vector less than 10^{-12}.

For the 2D flow around a cylinder, it was possible to obtain very accurate reference values from computational results on very fine grids, see Reference [2]. These reference values can be considered to be reliable for at least six leading digits. But in 3D it is not possible to use such extremely fine meshes like in 2D. Hence, the same accuracy of the reference values cannot be expected. However, it is possible to define reference values which allow a clear assessment of the results obtained with higher and lower order finite element spaces or with boundary fitted and non-boundary fitted discretizations. We obtain the reference values by extrapolating the results which we consider as most accurate. These are the results computed with the boundary fitted Q_2/P_1^{disc} finite element discretization, see Table III. The numerical order of convergence for the drag and the lift coefficient, which is given in the column 'Order'

in this table, is computed by

$$\log\left(\frac{|c_{\mathrm{d}}^{l-1} - c_{\mathrm{d}}^{l}|}{|c_{\mathrm{d}}^{l} - c_{\mathrm{d}}^{l+1}|}\right) \Big/ \log(2)$$

This order is approximately 4. A fourth order convergence of these coefficients was observed for this discretization also in the 2D flow around a cylinder [2]. Thus, we take $c_{\mathrm{d,ref}} = 6.1853267$ and $c_{\ell,\mathrm{ref}} = 9.4009839\mathrm{e}-3$. The values for Δp do not converge monotonically for the boundary fitted Q_2/P_1^{disc} finite element discretization such that we cannot extrapolate them. To define a reference value, we take the average of the value on level 4 given in Table III and the value computed with the boundary fitted Q_3/P_2^{disc} finite element discretization on level 3 (22.7 millions degrees of freedom) $\Delta p = 1.70875442\mathrm{e}-1$. This leads to $\Delta p_{\mathrm{ref}} = 1.70826996\mathrm{e}-1$. We think that at least the three leading digits of all defined reference values are correct.

The differences of the computed drag coefficients to $c_{\mathrm{d,ref}}$ are presented in Figure 5. The most accurate coefficients are computed with boundary fitted finite element discretizations of higher order. Using a polyhedral approximation of the boundary gives in general drag coefficients which are considerably worse. Only for the P_2/P_1 finite element discretization, the order of convergence seems to be the same for the boundary fitted and the polyhedral version. The most inaccurate results are obtained with non-conforming finite element discretizations of lowest order, e.g. the Q_1^{rot}/Q_0 Samarskij upwind discretization with 20 million degrees of freedom gives a more inaccurate result than the boundary fitted Q_3/P_2^{disc} finite element discretization with 50 000 degrees of freedom. It was not possible to solve the systems for the lowest order non-conforming Galerkin finite element discretizations on coarse levels.

The computed lift coefficients are compared in Figure 6. Similar to the drag coefficient, the most accurate lift coefficients are computed in general with boundary fitted finite element discretizations of higher order. Only for the P_2/P_1 finite element, the non-boundary fitted version gives a bit more accurate lift coefficients than the boundary fitted counterpart. In comparison to the higher order discretizations, the results obtained with the lowest order non-conforming finite element discretizations are extremely inaccurate.

Figure 7 presents the errors of the computed pressure differences to Δp_{ref}. In general, there is almost no difference in the computed values between boundary fitted and polyhedral finite element methods of higher order. Once more, the lowest order non-conforming discretizations are by far the most inaccurate ones.

6. THE COMPARISON OF THE SOLVERS

This section investigates the efficiency of the solvers introduced in Section 4. They were applied for solving the linear saddle point problems obtained by higher order finite element discretizations and non-stabilized lowest order non-conforming discretizations.

The finite element spaces which were considered in this numerical study, the discretizations which were applied, the geometric grid levels L on which the solutions are computed and the corresponding number of degrees of freedom on the finest level of the multigrid hierarchy are given in Table IV. We used the same initial grids as for computing the benchmark parameters, see Figure 4.

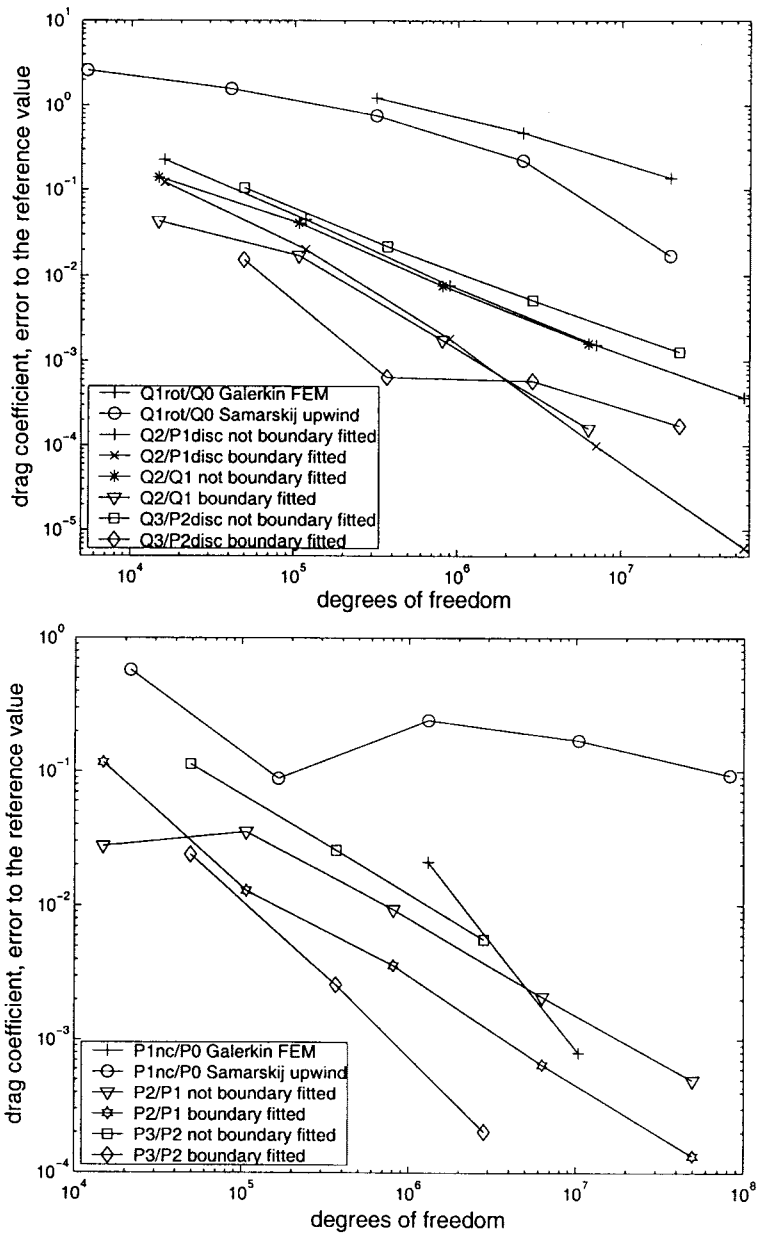

Figure 5. Errors of the computed drag coefficients to $c_{\mathrm{d,ref}}$.

The initial iterate on the geometric grid level L was chosen to be the interpolation of the solution of level $L-1$ in each test. We applied two stopping criteria for the iterative solution of the linear system in each step of the fixed point iteration for solving the Navier–Stokes equations. Either, the iteration is stopped after the reduction of the Euclidean norm

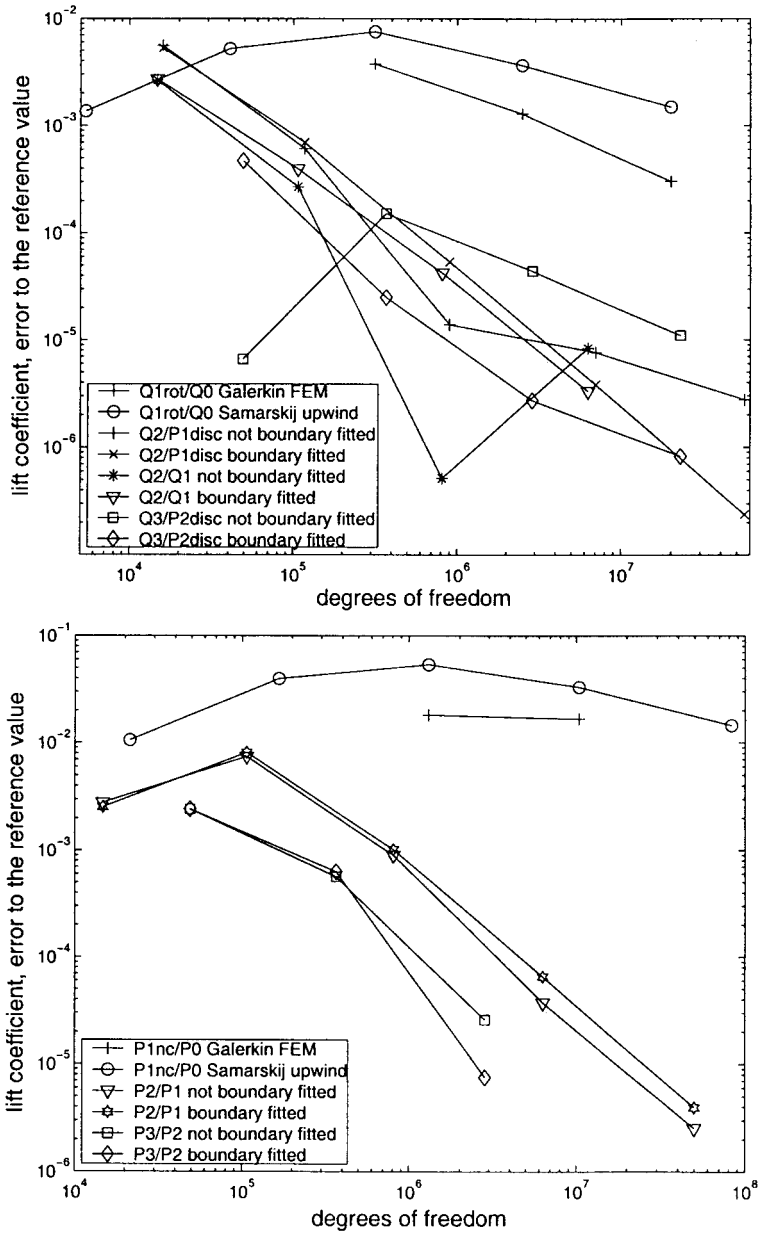

Figure 6. Errors of the computed lift coefficients to $c_{\ell,\mathrm{ref}}$.

of the residual vector by the factor 10 or maximal 10 iterations (multigrid or FGMRES) were performed. The fixed point iteration itself was stopped if the Euclidean norm of the residual vector was less than 10^{-10}. If not mentioned otherwise, the local systems are solved by Gaussian elimination.

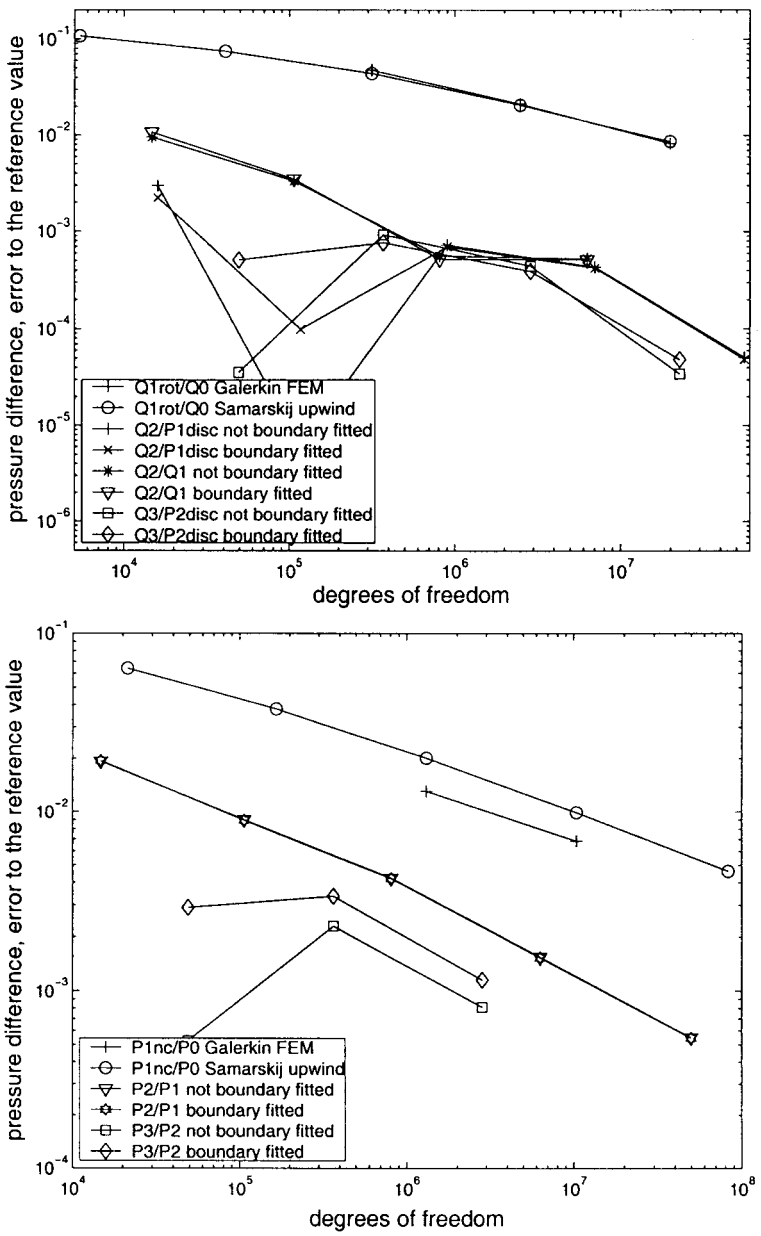

Figure 7. Errors of the computed pressure differences to Δp_{ref}.

The efficiency of the solvers will be evaluated by computing times. Of course, the computing time is influenced by many factors which are computer and compiler dependent, but it is the most important measure in applications. All computations were performed on a computer with HP PA-RISC 8500 processors (440 MHz, 1760 Mflops/s peak). To simplify the evaluation of

Table IV. Finite element spaces, discretizations, geometric grid level L and number of degrees of freedom for the comparison of the solvers.

Finite element space	Discretization	L	d.o.f.
Q_1^{rot}/Q_0	Galerkin	3	2 496 000
P_1^{nc}/P_0	Galerkin	3	10 398 720
Q_2/P_1^{disc}	boundary fitted Galerkin	3	7 035 840
Q_2/Q_1	boundary fitted Galerkin	2	810 160
P_2/P_1	boundary fitted Galerkin	2	810 160
Q_3/P_2^{disc}	boundary fitted Galerkin	2	2 882 640
P_3/P_2	boundary fitted Galerkin	1	367 000

Table V. Q_1^{rot}/Q_0 Galerkin finite element discretization, multigrid methods with mesh cell oriented Vanka smoother.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	—	Div.	22/209	13 054	12/108	3822	10/70	4472
$V(2,2)$	11/96	6146	11/55	6503	10/82	5311	10/49	5803
$V(3,3)$	11/90	8365	10/47	8099	11/70	6515	10/39	6722
$F(1,1)$	—	Div.	10/29	*3044*	10/69	*2988*	10/28	*2954*
$F(2,2)$	10/49	3750	10/19	3668	10/44	3373	10/20	3865
$F(3,3)$	10/43	4716	10/19	5303	10/37	4038	10/18	5044
$W(1,1)$	—	Div.	10/28	3286	10/62	*2805*	10/28	3273
$W(2,2)$	10/44	3474	10/18	3788	10/39	3095	10/18	3780
$W(3,3)$	10/38	4282	10/18	5447	10/35	3961	10/18	5443

the results given in Tables V–XII, we emphasize the fastest computing times in each test and all other computing times which are within 10% of the fastest time.

We did not study the behaviour of the solvers with respect to the damping parameter in the smoother iteration, see Section 4.3. In most tests, we did not apply damping at all, i.e. we used $\omega_l = 1$ on all grid levels l. Only in the computations using the Q_2/Q_1 and P_3/P_2 finite element spaces, we used $\omega_l = 0.8$, $l = 0, \ldots, L$ in the standard multigrid method and $\omega_l = 1$, $l = 0, \ldots, L$, $\omega_{L+1} = 0.8$ in the multiple discretization multigrid method.

In the column 'mg cyc.' of Tables V–XII, we give the number of fixed point iterations to solve the non-linear system and the total number of iterations of the linear solver within these fixed point iterations. The abbreviation 'Div.' indicates that the iteration diverges and 'No conv.' means that the stopping criterion of the fixed point iteration is not fulfilled after 50 steps. The computing times are given in seconds.

The evaluation of the numerical results is split into three parts.

In the first part, we consider non-stabilized lowest order non-conforming finite element discretizations, see Tables V and VI. Although the computed benchmark coefficients are rather inaccurate, see Section 5, the solution of the discrete problems causes some difficulties such

Table VI. P_1^{nc}/P_0 Galerkin finite element discretization, multigrid methods with mesh cell oriented Vanka smoother.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	Div.	—	Div.	—	19/184	20 960	10/81	16 750
$V(2,2)$	Div.	—	Div.	—	11/92	18 449	10/51	18 511
$V(3,3)$	Div.	—	No conv.	—	10/68	19 268	9/39	20 255
$F(1,1)$	Div.	—	Div.	—	No conv.	—	No conv.	—
$F(2,2)$	Div.	—	9/27	15 937	No conv.	—	9/24	*14 176*
$F(3,3)$	Div.	—	9/21	17 649	9/42	*13 913*	9/19	15 965
$W(1,1)$	Div.	—	Div.	—	Div.	—	No conv.	—
$W(2,2)$	Div.	—	9/33	20 800	No conv.	—	9/24	15 418
$W(3,3)$	Div.	—	9/20	18 232	9/42	*14 330*	9/18	16 340

Table VII. Q_2/P_1^{disc} boundary fitted Galerkin finite element discretization, multigrid methods with mesh cell oriented Vanka smoother.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	5/23	26 709	No conv.	—	5/17	19 439	6/40	38 994
$V(2,2)$	5/12	26 787	6/36	69 234	5/13	28 546	6/25	48 130
$V(3,3)$	5/11	36 387	5/20	57 486	5/10	32 550	5/19	54 650
$F(1,1)$	5/18	24 836	5/23	23 388	5/14	19 232	5/16	*16 252*
$F(2,2)$	5/11	29 043	5/18	36 067	5/9	23 335	3/13	26 009
$F(3,3)$	5/8	31 002	5/13	38 892	5/8	30 483	5/11	32 820
$W(1,1)$	5/17	24 309	5/23	23 518	5/14	20 094	5/16	*16 417*
$W(2,2)$	5/11	30 398	5/17	34 357	5/9	24 369	5/13	26 213
$W(3,3)$	5/8	32 006	5/12	36 381	5/8	31 430	5/11	33 057

that these discretizations may serve as good examples for testing solvers. For the Q_1^{rot}/Q_0 Galerkin finite element discretization, only the standard multigrid method with one pre- and post-smoothing step fails. The fastest computing times of the multiple discretization multigrid method and the FGMRES method with both types of multigrid preconditioners are similar. Considering also the computing times using more than one pre- and post-smoothing step, the FGMRES method with standard multigrid method preconditioner might be preferred somewhat. The situation is considerably different for the P_1^{nc}/P_0 Galerkin finite element discretization, Table VI. For this discretization, the standard multigrid method failed completely. The multiple discretization multigrid method converged if the F- or W-cycle with at least two pre- and post-smoothing steps were used. Using the multigrid methods as preconditioner in FGMRES increases the robustness of the solver considerably. The FGMRES

Table VIII. Q_3/P_2^{disc} boundary fitted Galerkin finite element discretization, multigrid methods with mesh cell oriented Vanka smoother, local systems solved with Gaussian elimination.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	5/25	45 017	No conv.	—	5/18	32 065	6/43	50 536
$V(2,2)$	5/13	42 302	6/33	78 390	5/13	41 440	6/25	58 387
$V(3,3)$	5/11	50 430	6/25	88 935	5/11	49 681	6/22	76 969
$F(1,1)$	5/29	64 974	5/36	42 435	5/17	38 841	5/23	*27 126*
$F(2,2)$	5/12	47 718	5/24	56 156	5/11	45 238	5/19	44 585
$F(3,3)$	5/11	59 641	5/18	63 031	5/11	59 748	5/12	42 192
$W(1,1)$	5/29	64 596	5/36	43 373	5/17	38 816	5/23	*27 193*
$W(2,2)$	5/12	47 505	5/24	56 471	5/11	45 219	5/18	42 207
$W(3,3)$	5/11	59 109	5/18	64 620	5/11	59 631	5/14	49 123

Table IX. Q_3/P_2^{disc} boundary fitted Galerkin finite element discretization, multigrid methods with mesh cell oriented Vanka smoother, local systems solved approximately with GMRES.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	Div.	—	Div.	—	5/28	28 846	6/45	30 469
$V(2,2)$	6/29	51 444	Div.	—	5/15	*27 775*	6/34	46 686
$V(3,3)$	5/16	41 569	Div.	—	5/10	*27 121*	6/29	58 694
$F(1,1)$	Div.	—	Div.	—	No conv.	—	6/36	*26 229*
$F(2,2)$	5/36	71 531	Div.	—	5/16	38 606	5/24	34 411
$F(3,3)$	5/17	51 889	Div.	—	5/10	32 797	5/19	40 726
$W(1,1)$	Div.	—	Div.	—	No conv.	—	6/35	*25 590*
$W(2,2)$	5/36	71 492	Div.	—	5/16	38 334	5/24	34 689
$W(3,3)$	5/17	51 874	Div.	—	5/10	32 448	5/19	40 771

with the multiple discretization multigrid method preconditioner converges already with two pre- and post-smoothing steps in the F- and W-cycle whereas the standard multigrid method preconditioner within the FGMRES needs three pre- and post-smoothing steps.

Altogether, for non-stabilized lowest order non-conforming finite element discretizations, the FGMRES with either type of multigrid preconditioner was the most efficient of the considered solvers.

The second part of the evaluation concerning the behaviour of the solvers considers their application to higher order finite element discretizations with discontinuous pressure approximation (Tables VII–IX). We apply the mesh cell oriented Vanka smoother for these discretizations. For the Q_2/P_1^{disc} boundary fitted Galerkin finite element discretization, the fastest

Table X. P_2/P_1 boundary fitted Galerkin finite element discretization, multigrid methods with pressure node oriented Vanka smoother.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	9/78	22 231	10/85	18 287	9/62	18 145	10/69	15 598
$V(2,2)$	9/44	24 265	9/46	19 240	9/32	18 328	9/41	18 186
$V(3,3)$	9/34	26 765	9/36	22 050	9/25	20 426	9/33	21 731
$F(1,1)$	9/81	26 611	10/54	*12 657*	9/61	20 704	10/48	*11 963*
$F(2,2)$	9/50	31 793	9/30	13 527	9/34	22 216	9/29	13 979
$F(3,3)$	9/34	32 438	9/25	16 725	9/26	25 838	9/24	16 865
$W(1,1)$	9/81	26 746	10/51	*12 200*	9/61	20 682	10/48	*12 058*
$W(2,2)$	9/50	31 788	9/28	*12 959*	9/34	22 243	9/27	*13 049*
$W(3,3)$	9/34	32 516	9/25	16 903	9/26	25 840	9/24	17 191

Table XI. Q_2/Q_1 boundary fitted Galerkin finite element discretization, multigrid methods with pressure node oriented Vanka smoother.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	12/120	137 114	No conv.	—	10/95	113 719	11/105	101 441
$V(2,2)$	11/110	241 408	11/107	201 564	9/85	190 924	9/84	164 816
$V(3,3)$	12/120	390 973	9/86	242 904	9/83	274 209	9/75	221 758
$F(1,1)$	11/110	139 530	12/120	113 453	10/94	134 995	10/87	*86 610*
$F(2,2)$	11/110	279 385	9/86	163 748	9/82	219 214	9/78	153 761
$F(3,3)$	12/120	502 987	9/85	243 214	9/78	308 347	9/68	194 954
$W(1,1)$	11/110	138 957	11/110	104 770	10/94	133 899	10/87	*86 486*
$W(2,2)$	11/110	275 076	9/86	163 610	9/82	217 081	9/78	151 624
$W(3,3)$	12/120	490 209	9/85	249 410	9/78	307 020	9/67	193 803

computing times are obtained with FGMRES and the multiple discretization multigrid method $F(1,1)$- and $W(1,1)$-cycle as preconditioner. Using both multigrid methods as preconditioner in FGMRES was in general faster than using them as solver. Most of the solvers work satisfactorily, only the solvers with the multiple discretization multigrid method V-cycles behaved worse than the other ones. For the Q_3/P_2^{disc} boundary fitted Galerkin finite element discretization, we study also the behaviour of the solvers with respect to the two approaches of solving the local systems described in Section 4.3. Table VIII contains the results for solving the local systems by Gaussian elimination. The solvers behave very similar to the Q_2/P_1^{disc} pair of finite element spaces. Clearly the fastest solver is the FGMRES method with the multiple discretization multigrid method $F(1,1)$- and $W(1,1)$-cycle as preconditioner. And, the

Table XII. P_3/P_2 boundary fitted Galerkin finite element discretization, multigrid methods with pressure node oriented Vanka smoother.

Cycle					FGMRES +			
	Standard mg		Mult. disc. mg		Standard mg		Mult. disc. mg	
	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time	mg cyc.	Time
$V(1,1)$	Div.	—	9/84	42 586	Div.	—	10/78	40 679
$V(2,2)$	Div.	—	10/63	64 401	Div.	—	10/56	58 474
$V(3,3)$	Div.	—	9/52	79 431	Div.	—	9/41	65 661
$F(1,1)$	Div.	—	10/79	40 317	Div.	—	10/62	*33 171*
$F(2,2)$	Div.	—	10/55	55 802	Div.	—	9/43	45 883
$F(3,3)$	Div.	—	9/38	57 340	Div.	—	9/33	51 495
$W(1,1)$	Div.	—	9/79	39 872	Div.	—	10/62	*33 359*
$W(2,2)$	Div.	—	10/55	56 866	Div.	—	9/43	45 840
$W(3,3)$	Div.	—	9/38	59 118	Div.	—	9/33	53 110

solvers with multiple discretization multigrid V-cycles worked considerably worse than all other solvers. Solving the local systems only approximately makes the solvers less robust, see Table IX. The multiple discretization multigrid method failed completely as solver and also the standard multigrid method worked much less efficient. The FGMRES method was necessary to construct an efficient and robust solver using this approach for solving the local systems. Only FGMRES with the standard multigrid method $F(1,1)$- and $W(1,1)$-cycle preconditioner did not converge. For all other multigrid cycles, the computing times are better than using Gaussian elimination for solving the local systems. However, comparing the fastest computing times, the gain is small.

For discretizations with discontinuous pressure approximation, the FGMRES with the multiple discretization multigrid method $F(1,1)$- and $W(1,1)$-cycle as preconditioner was the most efficient solver. The application of the multigrid methods as preconditioner in FGMRES is much more efficient than using them as solver. The global saddle point problems can be solved with nearly all solvers using Gaussian elimination for solving the local systems. The approximate solution of the local systems improves the efficiency in most cases in the FGMRES solver and it led to a completely unsatisfactory behaviour of the multigrid methods as solvers. The stopping criterion for the GMRES iteration in the solution of the local systems is a sensible parameter which balances efficiency and robustness. We prefer, for the reason of the better robustness, the Gaussian elimination for solving the local problems.

Last, we consider discretizations with continuous pressure approximation (Tables X–XII). For solving the saddle point problems arising from such discretizations, we used the pressure node oriented Vanka smoother and the local problems were solved approximately by GMRES, see Section 4.3. This was much faster than using Gaussian elimination, e.g. considering the P_2/P_1 finite element discretization and FGMRES with the multiple discretization multigrid method $F(1,1)$-cycle preconditioner, the computation takes with Gaussian elimination more than 78 000 s in contrast to less than 12 000 s as given in Table X. For the boundary fitted P_2/P_1 finite element discretization, the multiple discretization multigrid method is clearly more efficient than the standard multigrid method. There is almost no difference between using the

former as solver or as preconditioner in FGMRES. Again, the multiple discretization multigrid method V-cycle is considerably worse than the F- and W-cycle. The computing times using the Q_2/Q_1 finite element discretization, Table XI, are extremely long compared to the P_2/P_1 finite element discretization with the same number of degrees of freedom, Table X, and the Q_2/P_1^{disc} discretization on a finer grid, see Table VII. The standard multigrid approach was also for this discretization clearly less efficient than the multiple discretization multigrid approach. Best behaves once more the FGMRES method with multiple discretization multigrid $F(1,1)$- and $W(1,1)$-cycle preconditioner. The standard multigrid method fails completely for the P_3/P_2 finite element discretization, Table XII. The reason is the divergence of the smoother iteration on the coarsest grid. In contrast, the saddle point problems can be solved with all solvers which use the multiple discretization multigrid method. Also here, the fastest computing times were obtained by the FGMRES method with the multiple discretization multigrid $F(1,1)$- and $W(1,1)$-cycle as preconditioner.

Only the FGMRES method with multiple discretization multigrid preconditioner has been proven to be an efficient solver for the saddle point problems arising from the considered discretizations with continuous pressure approximation. For such discretizations, the multiple discretization multigrid approach was much superior to the standard multigrid approach. It was more efficient to use the multigrid methods as preconditioner in FGMRES than to use them as solvers.

7. SUMMARY

We like to summarize the most important conclusions of our numerical study at the three-dimensional flow around a cylinder.

- New reference values for the benchmark coefficients could be defined.
- The most accurate benchmark coefficients, especially drag and lift coefficients, could be obtained with boundary fitted higher order finite element discretizations.
- The use of lowest order non-conforming discretizations led to comparatively very inaccurate results.
- The arising discrete systems could be solved fastest with the FGMRES method and the multiple discretization multigrid method F- and W-cycle preconditioner.
- Using the multigrid methods as preconditioner in FGMRES was much more efficient than using them as solver.

REFERENCES

1. Schäfer M, Turek S. The benchmark problem "Flow around a cylinder". In Hirschel EH (ed.), *Flow Simulation with High-Performance Computers II*, vol. 52. Notes on Numerical Fluid Mechanics, Vieweg. 1996; 547–566.
2. 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.
3. Saad Y. A flexible inner–outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* 1993; **14**(2):461–469.
4. John V. *Parallele Lösung der inkompressiblen Navier–Stokes Gleichungen auf adaptiv verfeinerten Gittern*. Ph.D. Thesis, Otto-von-Guericke-Universität Magdeburg, Fakultät für Mathematik, 1997.
5. Rannacher R, Turek S. Simple nonconforming quadrilateral Stokes element. *Numerical Methods for Partial Differential Equations* 1992; **8**:97–111.
6. Schieweck F. *Parallele Lösung der stationären inkompressiblen Navier–Stokes Gleichungen*. Otto-von-Guericke-Universität Magdeburg, Fakultät für Mathematik, 1997. Habillitation.

7. Crouzeix M, Raviart P-A. Conforming and nonconforming finite element methods for solving the stationary Stokes equations I. *RAIRO Analyse Numérique* 1973; **7**:33–76.
8. Schieweck F, Tobiska L. An optimal order error estimate for an upwind discretization of the Navier–Stokes equations. *Numerical Methods for Partial Differential Equations* 1996; **12**:407–421.
9. John V, Tobiska L. Smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2000; **33**:453–473.
10. 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.
11. Braess D, Sarazin R. An efficient smoother for the Stokes problem. *Applied Numerical Mathematics* 1997; **23**(1):3–19.
12. Schieweck F. *A general transfer operator for arbitrary finite element spaces*. Preprint 00-25, Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg, 2000.
13. Vanka S. Block-implicit multigrid calculation of two-dimensional recirculating flows. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**(1):29–48.
14. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company: Boston, 1996.

