A Comparison of Three Solvers for the Incompressible Navier–Stokes Equations

V. John[1], G. Matthies[1], T.I. Mitkova[1], L. Tobiska[1] and P.S. Vassilevski[2]
[1]Otto–von–Guericke–Universität Magdeburg, Institut für Analysis und Numerik,
Postfach 4120, 39016 Magdeburg, Germany,
http://www.math.uni-magdeburg.de/institute/ian/
[2]Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences. Acad.
G. Bontchev str., Bl. 25 A, 1113 Sofia, Bulgaria

Summary

Three solvers for saddle point problems arising from the linearization and discretization of the steady state incompressible Navier–Stokes equations are numerically studied. The numerical tests are based on nonconforming finite element approximations of lowest order in two–dimensional domains using isotropic meshes. The investigated solvers are coupled multigrid methods with Vanka–type smoothers, flexible GMRES with a block–diagonal preconditioner proposed by Vassilevski and Lazarov (1996), and flexible GMRES with a coupled preconditioner which is based on an algebraic multigrid method.

AMS(MOS) subject classifications. 65N22, 65N55
Key words. Incompressible Navier–Stokes equations, coupled multigrid methods, preconditioned flexible GMRES, algebraic multigrid methods

1 Introduction

Many models of complex applications in physics and engineering describing phenomena connected with fluid motion involve as a part the incompressible Navier–Stokes equations. Examples are models in ferro hydrodynamics and crystal growth [5]. Hence, the fast and robust numerical solution of Navier–Stokes equations as a subproblem is the core of an efficient solution algorithm for more complex problems. Modern codes to attack these problems often contain several solvers for the Navier–Stokes equations. For a particular problem, the choice of an appropriate solver may be a difficult task because it depends on many aspects, e.g. if the equation is stationary or time dependent, on the size of the Reynold's number, the discretization, and the complexity of the domain. As a first step, this paper presents a comparison of three different solvers applied to examples with a simple geometric structure.

Here, we restrict ourselves to the steady state incompressible Navier–Stokes equations

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega \\ \nabla\cdot\mathbf{u} &= 0 && \text{in } \Omega \\ \mathbf{u} &= \mathbf{g} && \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain, ν is the viscosity of the fluid, $\mathbf{f}$ represents body forces driving the fluid, and $\mathbf{g}$ is the prescribed velocity field on the boundary.

The Navier–Stokes equations are solved with a fixed point iteration. Starting with an initial guess $(\mathbf{u}^0, p^0)$, an Oseen equation

$$\begin{aligned} -\nu\Delta\mathbf{u}^{n+1} + (\mathbf{u}^n\cdot\nabla)\mathbf{u}^{n+1} + \nabla p^{n+1} &= \mathbf{f} && \text{in } \Omega \\ \nabla\cdot\mathbf{u}^{n+1} &= 0 && \text{in } \Omega \\ \mathbf{u}^{n+1} &= \mathbf{g} && \text{on } \partial\Omega \end{aligned} \tag{1.2}$$

must be solved in each iteration step. In the case of small viscosity ν, one has also to take into consideration the transport dominated character of (1.2).

The linear equation (1.2) is discretized with nonconforming finite elements of lowest order which satisfy the inf–sup condition. In particular, the nonconforming P_1/P_0–finite element [1] have been used on triangular meshes and the nonconforming rotated bilinear Q_1^{rot}/Q_0–finite element [8] have been used on quadrilateral meshes. The dominating convective term has been stabilized by an upwind scheme [7, 11]. Finally, we have to solve a saddle point problem of the form

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ s \end{pmatrix}. \tag{1.3}$$

In this report, the efficiency of three different solvers for (1.3) is studied numerically.

2 The Solvers for the Saddle Point Problem

Solver 1 : A Coupled Multigrid Method with Vanka–Type Smoothers. A widely used class of solvers for (1.3) are coupled multigrid methods with smoothers of Vanka–type. Vanka–type smoothers, introduced in [17], can be considered as block Gauss–Seidel methods. A block consists of all degrees of freedom which are connected to one mesh cell, i.e. the block size is 7 for the nonconforming P_1/P_0–finite element method and 9 for the nonconforming Q_1^{rot}/Q_0–finite element method. Within a smoothing step of the Vanka–type smoother, a loop over all mesh cells of the triangulation is performed. In each mesh cell, the so-called *full Vanka smoother* solves a small linear saddle point problem of type (1.3) whereas the *diagonal Vanka smoother* solves a small system where the matrix A is replaced by its diagonal. The coarse grid equations are defined by discretizing the equation on the corresponding grid and by defect restriction from the next finer grid. An averaging process is used to prolongate nonconforming functions. The system on the coarsest grid is solved iteratively by the smoother. In Sect. 3, results obtained with the W(2,2)–cycle are presented.

This solver needs the storage of the complete grid hierarchy, the connections between the grids, and the coarse grid systems. However, the definition of appropriate coarse grids may be difficult for complicated domains and the Vanka–type smoothers become less efficient for meshes with anisotropic mesh cells.

Coupled multigrid methods with Vanka–type smoothers were studied already extensively within numerical tests [2, 3, 4, 13]. In these tests, they often showed a superior behaviour to other solvers of (1.3) which are not the subject of this paper.

Solver 2 : Flexible GMRES with Block Diagonal Preconditioner. The second solver is a flexible GMRES method (FGMRES), [9], with a block diagonal preconditioner proposed in [18]. To apply this solver, (1.3) is transformed into the equivalent system

$$\begin{pmatrix} A+\delta^{-1}BB^T & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{r}+\delta^{-1}Bs \\ s \end{pmatrix}. \tag{2.1}$$

The block diagonal preconditioner has the form

$$\begin{pmatrix} A+\delta^{-1}BB^T & 0 \\ 0 & \delta_1 I \end{pmatrix} \tag{2.2}$$

with appropriate chosen constants δ and δ_1. We used in the numerical tests the same parameters which are proposed in [18] for the Stokes equations, i.e. $\delta = 1$ and $\delta_1 = h^2$, where h is the maximal diameter of the mesh cells of the current grid.

An explicit storage of $A+\delta^{-1}BB^T$ increases the memory requirements considerably in comparison to (1.3). Since the matrices A and $A+\delta^{-1}BB^T$ have the structures

$$A = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{11} \end{pmatrix}, \quad A+\delta^{-1}BB^T = \begin{pmatrix} A_{11}+\delta^{-1}B_{11} & \delta^{-1}B_{12} \\ \delta^{-1}B_{12}^T & A_{11}+\delta^{-1}B_{22} \end{pmatrix},$$

in (1.3) only the block A_{11} must be stored whereas in (2.1) the storage of three blocks would be necessary. However, the system matrix of (2.1) is needed only in matrix–vector products during the FGMRES iteration. Therefore, in our implementation we did not used an explicit storage of $A+\delta^{-1}BB^T$. The application of the preconditioner requires the solution of a linear system with the system matrix $A+\delta^{-1}BB^T$. Since this matrix is known only implicitly, an iterative method for nonsymmetric systems which uses only matrix–vector products is a suitable solver. In the numerical tests presented below, we used BICGSTAB [14]. In our tests, at most 50 FGMRES iterations were applied for the solution of the Oseen problems and after each 20 iterations the FGMRES was restarted. The solution of the preconditioner system was stopped either after having reduced the Euclidean norm of the residual by the factor 10 or after 100 BICGSTAB iterations.

In contrast to Solver 1, only the system on the finest grid is needed. However, the vectors for the outer FGMRES iteration and the vectors for the preconditioner BICGSTAB iteration must be stored additionally.

In [18], this solver was studied for the Stokes equations. We present first results on its behaviour for Oseen equations.

Solver 3 : Flexible GMRES with Coupled Preconditioner. This method uses a preconditioner of the form

$$\begin{pmatrix} C & B \\ B^T & 0 \end{pmatrix} \tag{2.3}$$

within the FGMRES, where $C^{-1} \approx A^{-1}$ and C is easily invertible. In the numerical tests presented in this report, C^{-1} represents few cycles of an algebraic multigrid method applied to a system $Ax = b$. Within the FGMRES not more than 10 iterations were applied.

The application of the preconditioner requires the solution of a saddle point problem

$$\begin{pmatrix} C & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ s \end{pmatrix}, \tag{2.4}$$

which is solved by a pressure Schur complement method. In this method, the pressure Schur complement equation

$$B^T C^{-1} B p = B^T C^{-1} \mathbf{r} - s \tag{2.5}$$

is solved approximately with some GMRES iterations. In the numerical tests, 10 iterations were performed. After having computed an approximate solution for p, an approximate solution for u can be obtained by solving the first equation in (2.4).

The algebraic multigrid method used for the approximation of A^{-1} is described in [6]. It belongs to the class of aggregation methods, i.e. the unknowns of the coarse grid are defined by an appropriate clustering of the unknowns of the fine grid. The coarse grids and coarse grid matrices of the algebraic multigrid have to be constructed before the iterative solution of the linear system can start. The diagonal entry a_{ii}^H of the coarse grid matrix is the sum of all couplings of the fine grid nodes which are forming cluster i. The off diagonal entry a_{ij}^H, $i \neq j$, is the sum of all fine grid entries a_{kl}^h, $k \neq l$, where node k belongs to cluster i and node l to cluster j. We used a constant prolongation and the restriction is defined to be the adjoint operator. ILU iterations are applied as smoother in the algebraic multigrid. The system on the coarsest grid is solved approximately with the smoother. In the numerical tests, we used the W(1,1)–cycle.

In contrast to the other two solvers, no geometrical information is needed. The algebraic multigrid method has the advantage that coarse grids are constructed easily independent of the complexity of the domain. However, the memory requirements of this solver are relatively large. Similar to a geometric multigrid method, coarse grid systems and connections between the grids must be stored. In addition, memory is necessary for the ILU factorizations and the vectors and matrices for the outer FGMRES iteration, and the vectors and matrices for the inner pressure Schur complement GMRES iteration.

3 The Numerical Tests

In this section, we present the results of the numerical studies. We consider an example with polynomial solution and the benchmark problem "Flow around a cylinder" [10]. The nonlinear iteration started in all tests and on all levels with 0 for all degrees of freedom. The iteration was stopped if the Euclidean norm of the residual of the nonlinear system was less than 10^{-6}. Within the nonlinear iteration, the linear systems were approximately solved until the norm of the residual was reduced by the factor 10 or this norm was less than $9.8 \cdot 10^{-7}$ or the prescribed maximal number of iterations was reached. The computations were performed on a HP work station J 282 with 33 MFlops (LINPACK).

EXAMPLE 1 An Example with Polynomial Solution. The right hand side and the boundary conditions are chosen such that $\mathbf{u} = curl\Psi$, $\Psi = x^2y^2(1-x)^2(1-y)^2$, $p = x^3 + y^3 - 0.5$ is the solution of (1.1) in $\Omega = (0{,}1)^2$. The coarsest quadrilateral mesh consists of four quadratic cells of equal size. To obtain the triangular mesh, each quadratic element is subdivided into two triangles by the diagonal with slope one. Starting with these initial triangulations, the finer meshes are constructed by uniform refinement.

Table 1 Ex. 1, degrees of freedom.

level	quadrilateral grid	triangular grid
5	5248	8320
6	20736	33024
7	82432	131584
8	328702	525312
9	1312768	2099200

Table 2 Ex. 1, $\nu = 1$, quadrilateral grid, nonlinear iterations (total number of linear iterations) and computing times in sec.

level	Solver 1 diag Vanka		Solver 1 full Vanka		Solver 2		Solver 3	
5	3 (6)	1	4 (5)	2	7 (114)	22	7 (23)	3
6	3 (5)	2	4 (5)	7	7 (173)	315	6 (24)	20
7	3 (5)	11	4 (5)	30	6 (156)	2504	6 (30)	119
8	3 (5)	52	3 (4)	104	13 (390)	32326	6 (32)	690
9	3 (4)	184	3 (4)	434			5 (37)	4132

The number of nonlinear iterations, the total number of linear iterations within the nonlinear iterations, and the computing times for $\nu = 1$ and $\nu = 10^{-3}$ are presented in Tables 2 - 5. The coupled multigrid W(2,2)–cycle with the diagonal Vanka smoother

Table 3 Ex. 1, $\nu = 10^{-3}$, quadrilateral grid, nonlinear iterations (total number of linear iterations) and computing times in sec.

level	Solver 1 diag Vanka	Solver 1 full Vanka	Solver 2	Solver 3
5	14 (24) 2	4 (5) 7	16 (583) 62	11 (44) 6
6	6 (8) 4	6 (8) 11	41 (1388) 2108	10 (21) 20
7	4 (6) 13	6 (7) 43	145 (4970) 66358	10 (25) 117
8	3 (5) 52	5 (6) 158	not converged	7 (23) 575
9	3 (4) 184	4 (5) 544		6 (24) 3585

Table 4 Ex. 1, $\nu = 1$, triangular grid, nonlinear iterations (total number of linear iterations) and computing times in sec.

level	Solver 1 diag Vanka	Solver 1 full Vanka	Solver 2	Solver 3
5	4 (7) 1	3 (6) 2	7 (109) 28	6 (22) 5
6	3 (6) 4	3 (6) 11	7 (185) 645	6 (26) 32
7	3 (6) 21	3 (5) 40	7 (187) 5231	6 (31) 189
8	3 (5) 84	2 (4) 138	11 (364) 46531	5 (34) 1001
9	3 (5) 363	2 (4) 579		5 (43) 6152

Table 5 Ex. 1, $\nu = 10^{-3}$, triangular grid, nonlinear iterations (total number of linear iterations) and computing times in sec.

level	Solver 1 diag Vanka	Solver 1 full Vanka	Solver 2	Solver 3
5	46 (67) 10	25 (43) 19	54 (1805) 726	17 (78) 18
6	11 (19) 14	12 (19) 37	not converged	11 (32) 41
7	13 (15) 52	11 (12) 100		12 (28) 207
8	4 (6) 99	4 (6) 215		8 (25) 933
9	3 (5) 359	3 (5) 743		6 (27) 5020

was clearly the best solver in all tests. In this example, we observe that this solver behaved robust for $\nu \in [10^{-3},1]$. Using the full Vanka smoother, the coupled multigrid method was somewhat slower but still much faster than the other solvers. Solver 2 showed an unsatisfactory behaviour. In our approach, we had immense problems to solve the linear system of the preconditioner with some iterative method. In general, the BICGSTAB iteration converged very slowly. Since the system matrix $A+\delta^{-1}BB^T$ is known only implicitly, we could not construct an appropriate preconditioner for this iteration. For $\nu = 10^{-3}$, the problem could not be solved. The FGMRES method with the coupled preconditioner, Solver 3, behaved much better than the FGMRES method with the block–diagonal preconditioner. Often, the systems could be solved somewhat faster for $\nu = 10^{-3}$ than for $\nu = 1$. However, compared to Solver 1, considerable more iterations were needed to solve the linear saddle point problems. In our opinion, a weak point of this solver consists in the use of an aggregation based algebraic multigrid method with constant prolongation. It was observed in [12], p. 117, that this type of algebraic multigrid method is inefficient as stand alone solver and behave substantially better as preconditioner. Thus, there are at least two ways to improve Solver 3. First, the constant prolongation can be improved by smoothing processes as described in [15, 16]. The second way is to define the approximation C^{-1} of A^{-1} by some iterative scheme where the algebraic multigrid method is used as preconditioner. At present, another bottle neck of this method is the setup of the algebraic multigrid. In our tests, the setup time scaled about quadratically with the number of degrees of freedom. The reasons for this still have to be analyzed. Using a different matrix C in (2.3) which is not based on a multigrid strategy led to a considerable increase in the computing times.

Table 6 Ex. 1, triangular grid, memory requirements in MB.

level	Solver 1	Solver 2	Solver 3
6	7	17	14
7	25	66	56
8	96	263	224
9	381		891

Table 6 shows that the coupled multigrid method with Vanka–type smoothers was not only the most efficient solver with respect to the computing times but also with respect to the memory requirements.

EXAMPLE 2 A Benchmark Problem. We consider a benchmark problem defined in the DFG priority research program *Flow Simulation with High–Performance Computers*, [10]. This benchmark problem describes the flow in a channel around a cylinder, see Figure 1. The parameters of the Navier–Stokes equations (1.1) are $\nu = 10^{-3}$, $\mathbf{f} = 0$, the parabolic inflow and outflow profile

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

and no–slip conditions on the other boundaries. The computations were performed on a quadrilateral mesh.

Figure 1 Domain for Ex. 2.

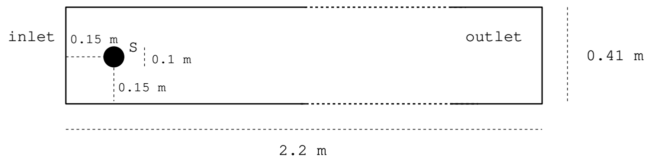

Table 7 Ex. 2, nonlinear iterations (total number of linear iterations) and computing times in sec.

degrees of freedom	Solver 1 diag Vanka	Solver 1 full Vanka	Solver 2	Solver 3
1984	7 (14) 1	7 (15) 3	12 (390) 99	11 (97) 4
7728	7 (16) 4	7 (17) 11	33 (1581) 2022	17 (157) 35
30496	7 (14) 14	7 (14) 34	157 (7801) 48294	25 (241) 347
121152	7 (13) 55	7 (12) 118	not converged	29 (278) 1943
482944	7 (12) 207	7 (10) 392		30 (285) 9854
1928448	7 (11) 770	6 (8) 1259		58 (567) 92931

The superiority of the coupled multigrid methods with Vanka–type smoothers is in this example even much more striking than in Example 1. Using Solver 3, great difficulties occurred in the solution of the Schur complement system (2.5). The number of nonlinear as well as linear iterations is much larger than for Solver 1.

4 Summary

In our studies, the coupled multigrid method with Vanka–type smoothers has been proven as clearly the best solver. Solver 3, which uses a multigrid strategy in the definition of the preconditioner (2.3), showed for small numbers of degrees of freedom an acceptable behaviour. There are several potential improvements to obtain a better performance for larger systems. First, smoothed prolongation in the algebraic multigrid method should be used instead of constant prolongation. Second, the approximation C^{-1} of A^{-1} could be defined as an iterative method with the algebraic multigrid method as preconditioner. And third, the bad scaling of the setup phase of the algebraic multigrid method has to be analyzed and improved. Solver 2, which does not include any multigrid component, was not competitive with the other solvers. Our numerical results with this new solver are in agreement with the conclusion of the DFG benchmark that appropriately applied multigrid methods are necessary to obtain efficient solvers for the incompressible Navier–Stokes equations, [10].

Acknowledgment

This work was partly supported by the Volkswagen Foundation under the grant I/70 480 and by the DFG under the grant EN 278/2 - 1.

Bibliography

[1] M. Crouzeix and P.-A. Raviart. Conforming and nonconforming finite element methods for solving the stationary Stokes equations I. *R.A.I.R.O. Analyse Numérique*, 7:33–76, 1973.

[2] V. John. On the parallel performance of coupled multigrid methods for the solution of incompressible Navier–Stokes equations. In M. Griebel, O.P. Iliev, S.D. Margenov, and P.S. Vassilevski, editors, *Large–Scale Scientific Computations of Engineering and Environmental Problems*, volume 62 of *Notes on Numerical Fluid Mechanics*, pages 269 – 280. Vieweg, 1998.

[3] V. John. A comparison of parallel solvers for the incompressible Navier–Stokes equations. *Comput. Visual. Sci.*, 1999. to appear.

[4] V. John and L. Tobiska. Numerical performance of smoothers in coupled multigrid methods for the parallel solution of the incompressible Navier–Stokes equations. Preprint 40/98, Otto-von-Guericke Universität Magdeburg, Fakultät für Mathematik, 1998.

[5] P. Knobloch. *Solvability and Finite Element Discretization of a Mathematical Model Related to Czochralski Crystal Growth*. PhD thesis, Otto–von–Guericke Universität Magdeburg, 1996.

[6] R. Lonsdale. An algebraic multigrid solver for the Navier–Stokes equations on unstructured meshes. *Int. J. Num. Meth. Heat Fluid Flow*, 3:3 – 14, 1993.

[7] K. Ohmori and T. Ushijima. A technique of upstream type applied to a linear nonconforming finite element approximation of convective diffusion equations. *R.A.I.R.O. Anal. Numer.*, 18(3):309–332, 1984.

[8] R. Rannacher and St. Turek. Simple nonconforming quadrilateral stokes element. *Numer. Meth. Part. Diff. Equ.*, 8:97 – 111, 1992.

[9] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[10] M. Schäfer and S. Turek. The benchmark problem "Flow around a cylinder". In E.H. Hirschel, editor, *Flow Simulation with High–Performance Computers II*, volume 52 of *Notes on Numerical Fluid Mechanics*, pages 547 – 566. Vieweg, 1996.

[11] F. Schieweck and L. Tobiska. An optimal order error estimate for an upwind discretization of the Navier–Stokes equations. *Num. Meth. Part. Diff. Equ.*, 12:407 – 421, 1996.

[12] K. Stüben. Algebraic multigrid (amg): An introduction with applications. Report 53, GMD, 1999. to appear in U. Trottenberg, C.W. Osterlee, A. Schüller, *Multigrid*, Academic Press.

[13] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An algorithmic approach in view of computational aspects*. Springer Verlag, 1999.

[14] H.A. van der Vorst. Bi–cgstab: A fast and smoothly converging variant of bi-cg for the solution of non–symmetric linear systems. *SIAM Sci. Stat. Comp.*, 12:631 – 644, 1992.

[15] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. UCD/CCM Report 34, University of Colorado at Denver, 1994.

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

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

[18] P.S. Vassilevski and R.D. Lazarov. Preconditioning mixed finite element saddle–point elliptic problems. *Num. Lin. Alg. with Appl.*, 3(1):1 – 20, 1996.