The solution of the discretized incompressible Navier-Stokes equations with iterative methods

Report 93-54

C. Vuik

TU Delft

Faculteit der Technische Wiskunde en Informatica
Faculty of Technical Mathematics and Informatics

Technische Universiteit Delft
Delft University of Technology

ISSN 0922-5641

Copies of these reports may be obtained from the bureau of the Faculty of Technical Mathematics and Informatics, Julianalaan 132, 2628 BL Delft, phone +3115784568.
A selection of these reports is available in PostScript form at the Faculty's anonymous ftp-site. They are located in the directory /pub/publications/tech-reports at ftp.twi.tudelft.nl

The solution of the discretized incompressible Navier-Stokes equations with iterative methods

1 Invariant discretization of the incompressible Navier-Stokes equations

In this paper we describe a numerical solution method for the incompressible Navier-Stokes equations in complex geometries. We use a finite volume technique in combination with boundary-fitted co-ordinates. The physical domain is mapped onto a computational domain consisting of a number of rectangular blocks. In this paper we restrict ourselves to the one block case. We use a regular mapping $T : x \rightarrow \xi$ from an arbitrary domain Ω to a rectangle G. In G a uniform computational grid is chosen.

The motivation for these choices is that we want to solve large two and three dimensional problems. In these problems it is important to obtain fast iterative methods to solve the discretized equations. This is easier using a finite volume technique instead of a finite element technique. Finally the structured grid enables us to develop a fast implementation of the methods on vector computers.

The incompressible Navier-Stokes equations in general co-ordinates are given by [5]:
the continuity equation

$$U^{\alpha}_{,\alpha} = 0 \; , \tag{1}$$

and the momentum equations

$$\frac{\partial}{\partial t}(\rho U^{\alpha}) + (\rho U^{\alpha} U^{\beta})_{,\beta} + (g^{\alpha\beta} p)_{,\beta} - \tau^{\alpha\beta}_{,\beta} = \rho f^{\alpha} \; , \tag{2}$$

where $\tau^{\alpha\beta}$ represents the deviatoric stress tensor

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma} U^{\beta}_{,\gamma} + g^{\gamma\beta} U^{\alpha}_{,\gamma}) \; ,$$

with $g^{\alpha\beta}$ the contravariant metric tensor, μ the viscosity, p the pressure, U^{α} the contravariant velocity component and ρ the density of the fluid. The transport equation for a scalar T is given by

$$c\frac{\partial T}{\partial t} + (U^{\alpha} T)_{,\alpha} - (K^{\alpha\beta} T_{,\beta})_{,\alpha} + dT = e \; , \tag{3}$$

where c, $K^{\alpha\beta}$, d and e are given functions.

In order to avoid possible pressure oscilations we use a staggered grid. The pressure is

Finally the spatial discretization described above is combined with finite differences for the time derivative and a pressure correction method. After Newton linearization we obtain two systems of equations [5], [9]: the momentum system and the pressure system. A discretization of equation (3) is called a transport system.

2 Iterative solution methods

In this section we describe iterative solution methods applied to the equations given in Section 1. Fast iterative methods to solve a system of linear equations are: multigrid methods and Krylov subspace methods. For a short survey of iterative methods applied to the incompressible Navier-Stokes equations we refer to [9]. In this paper we restrict ourselves to Krylov subspace methods. Since the matrices are non symmetric we are not able to use the conjugate gradient or conjugate residual method.

The following Krylov subspace methods can be used to solve non symmetric linear systems: LSQR [3], CGS [6], Bi-CGSTAB [7] and GMRES [4]. In our work we use GMRES-type methods, because these are robust methods with an optimal rate of convergence. To save memory and CPU time we have formulated the GMRESR method [8], [10]. This is a GMRES-type method with an inner and outerloop. In the innerloop a good search direction for the outerloop is calculated.

It is well known that combinations of a preconditioner with an iterative method lead to efficient solution methods. We use incomplete LU decompositions as preconditioners. In such a preconditioner one constructs a lower triangular matrix L and an upper triangular matrix U, where L and U have a prescribed nonzero pattern, and LU is a good approximation of A. The iterative method is then applied to

$$AU^{-1}L^{-1}y = b \ ,$$

where the solution x is given by $x = U^{-1}L^{-1}y$.

ILUD

For this preconditioner we construct $LD^{-1}U$ as an approximation of A. For further details we refer to [9]. In this preconditioning we use two extra vectors in memory.

ILU

The matrices L and U satisfy the following rules:

- $diag\ (L) = I$;
- the nonzero structure of $L + U$ is comparable with the nonzero structure of A ;
- if $a_{ij} \neq 0$ then $(LU)_{ij} = a_{ij}$.

The last rule can for $i = j$ be replaced by

- rowsum (LU) = rowsum (A),

the MILU preconditioner if the memory space is available.

Due to recurrencies the multiplication by L^{-1} or U^{-1} for these preconditioners run in scalar speed on a vector computer. We rewrite the loops in these multiplications in such a way that they run in vectorspeed. This is achived by using a diagonal ordering for a nine point stencil (see [1]).

Suppose n_i is the number of grid points in the x_i-direction. The pressure and transport systems have $n_1.n_2$ rows and columns, whereas the momentum system has $2n_1.n_2$ rows and columns. For the non zero structure of the pressure and momentum systems we refer to [9]. In both systems the nonzero structure is symmetric. The non zero structure of a transport system is equal to the pressure system. The momentum system has 13 non zero elements per row, whereas the other systems have 9 non zero elements per row. This implies that the memory space and the work to multiply a matrix with a vector for the momentum system is 3 times as large as for the other equations.

For iterative solution methods it is important to specify good stopping criteria and an accurate starting vector. For details we refer to [11].

During the solution of the pressure or a transport system the memory space of the momentum matrix is available. This motivates us to use an MILU preconditioner to solve the pressure and transport systems. The momentum system is solved using the MILUD preconditioner, because this preconditioner needs only a small amount of extra memory.

The curvilinear grid combined with a co-ordinate transformation results in a non symmetric pressure matrix. This is a disadvantage of the finite volume method, since using a finite element technique one obtains a symmetric pressure matrix. For the pressure system it appear that restarting GMRES destroys the superlinear convergence behaviour. So we always use full or truncated GMRESR [8]. The rate of convergence for the pressure system only depends on the grid. Solving the momentum and transport systems the rate of convergence also depends on ρ, μ, Δt, t etc. For these systems restarted GMRES gives good results.

3 Results for two dimensional problems

We describe two testproblems: the flow through a curved channel and a Boussinesq problem.

The physical domain of the curved channel is displayed in Figure 1. As initial condition we take the velocities equal to zero. The boundary conditions are: a parabolic velocity profile at the inflow (Boundary 1), a no slip condition on Boundary 2 and 4 and the normal stress and tangential velocity given on the outflow boundary (Boundary 3). We take $Re = 100$.

In the Boussinesq problem the Navier-Stokes equations are coupled with a transport equation. We use a standard benchmark problem, published by [2]. The physical domain and the 20×10 grid is displayed in Figure 2. For the velocities we take no slip boundary conditions. The temperature is described by a transport equation. As boundary conditions we take $T = 1$ at the left-hand wall and $T = 0$ at the right-hand wall. The lower and upper walls are isolated. We calculate the solution with $Re = 1$, $Pr = 0.71$ and $Ra = 10^6$.

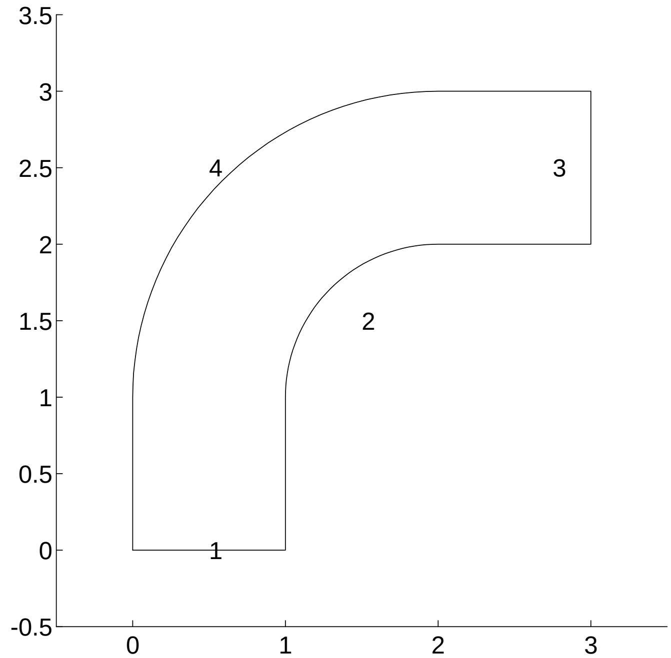

Figure 1: The physical domain of the curved channel problem

$\|r_k\|_2/\|r_0\|_2 \leq 10^{-6}$ is used. For large problems GMRESR(m) + MILU is the best method.

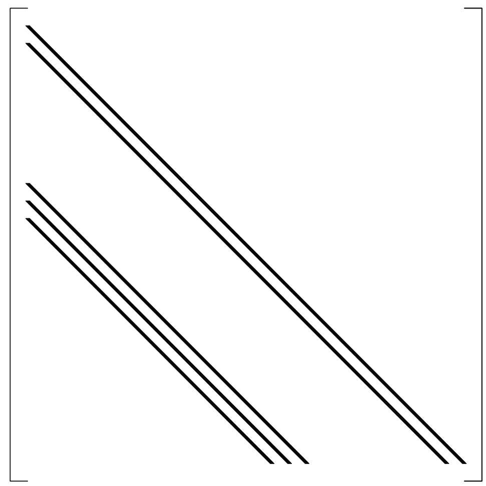

Figure 2: The 20×10 grid used in the Boussinesq problem

In the following table we give the vectorization properties of the preconditioner. In our

	GMRES(200)	GMRESR(10)	GMRESR(5) MILUD	GMRESR(3) MILU
iterations	178	19	8	9
CPU time	2.3	0.39	0.13	0.12
memory vectors	178	48	23	30

Table 1: The results for the pressure system.

problems the megaflop rate for a vector update is 35. Without vectorization the preconditioner has a megaflop rate equal to 9. From Table 2 it appears that the megaflop rate for the vectorized version becomes better for increasing grid sizes. For large grid sizes it is approximately equal to the megaflop rate of a vector update.

grid size	16×64	32×128	64×256
megaflop/s	14	30	35

Table 2: Megaflop rate of the vectorized preconditioner.

Finally we give a timing of all the different parts of the curved channel problem. The results are obtained for one timestep. In these problems 20-40 timesteps are in general sufficient to obtain a good approximation of the solution.

gridsize	timestep	building of the systems	momentum CPU	pressure	
				CPU	memory
16×64	0.15	0.06	0.08	0.14	28
32×128	0.075	0.19	0.22	0.5	31
64×256	0.0375	0.59	0.77	1.96	32
128×512	0.01875	2.10	3.2	9.26	37

Table 3: Results for one timestep using different gridsizes

gridsize	building of the systems	momentum CPU	pressure		transport CPU
			CPU	memory	
20×40	0.05	0.04	0.08	22	0.02
40×80	0.15	0.17	0.42	30	0.12
80×160	0.53	0.86	3.0	41	0.68
160×320	2.0	4.52	23.3	49	3.4

Table 4: Results for the Boussinesq problem.

4 Results for the three dimensional problems

Recently we have implemented a discretization of the three dimensional incompressible Navier-Stokes equations. We use our insights from 2D to obtain fast iterative solvers for 3D. It appears that the pressure matrix has $n_1 n_2 n_3$ rows and columns whereas the momentum matrix has $3n_1 n_2 n_3$ rows and columns. The non zero structure of the momentum matrix is not symmetric. One row of the pressure matrix has 19 non zero elements, and one row of the momentum matrix has 51 non zero elements. This means that the momentum matrix is $51 \cdot 3/19 = 8$ times as expensive as the pressure matrix with respect to memory and work.

We use GMRES combined with an ILUD preconditioner to solve the momentum system. The pressure system is solved by GMRESR(3) combined with the MILUD ($\alpha = 0.95$) preconditioner. The (M)ILUD preconditioner is vectorized and runs at 30 Mflop/s for a $30 \times 30 \times 30$ grid. The work and CPU time for one timestep are given in Table 5, solving a driven cavity with $Re = 1$ and $\Delta t = 0.025$. Note that as expected the momentum system costs more time

gridsize	building of the systems	momentum CPU	pressure	
			CPU	memory
$15 \times 15 \times 15$	2.9	0.9	0.35	11
$30 \times 30 \times 30$	22	9	3.3	27
$60 \times 60 \times 60$	150	121	37.0	41

Table 5: Results for one timestep using different gridsize

5 Conclusions

In this paper we have described properties of GMRES-type iterative methods to solve the discretized incompressible Navier-Stokes equations. We obtain good results with these methods, and observe that they run in vector speed on vector computers. In two dimensional problems the solution of the pressure equation is the most time consuming part, whereas in three dimensional problems the building of the systems is the most expensive part. Both points are subject of further study.

References

[1] C.C. Ashcraft and R.G. Grimes.
On vectorizing incomplete factorization and SSOR preconditioners.
SIAM J. Sci. Stat. Comput., **9**, pp. 122–151, (1988).

[2] G. De Vahl Davis and I.P. Jones.
Natural convection in a square cavity: a comparison exercise.
Int. J. Numer. Meth. Fluids, **3**, pp. 227–248, (1983).

[3] C.C. Paige and M.A. Saunders.
LSQR: an algorithm for sparse linear equations and sparse least squares.
ACM Trans. Math. Soft., **8**, pp. 43–71, (1982).

[4] Y. Saad and M.H. Schultz.
GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems.
SIAM J. Sci. Stat. Comput., **7**, pp. 856–869, (1986).

[5] A. Segal, P. Wesseling, J. van Kan, C.W. Oosterlee and K. Kassels.
Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted co-ordinates.
Int. J. Num. Meth. Fluids., **15**, pp. 411-426, (1992).

[6] P. Sonneveld.
CGS: a fast Lanczos-type solver for nonsymmetric linear systems.
SIAM J. Sci. Stat. Comput., **10**, pp. 36–52, (1989).

[7] 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 J. Sci. Stat. Comput., **13**, pp. 631–644, (1992).

[8] H.A. van der Vorst and C. Vuik.
GMRESR: a family of nested GMRES methods.
J. Num. Lin. Alg. Appl., to appear.
Report 91-80, Faculty of Technical Mathematics and Informatics,
Delft University of Technology, (1991).

[10] C. Vuik.
Further experiences with GMRESR.
Report 92-12, Faculty of Technical Mathematics and Informatics,
Delft University of Technology, (1992).

[11] C. Vuik.
Termination criteria for GMRES-like methods to solve the discretized incompressible Navier-Stokes equations.
Report 92-50, Faculty of Technical Mathematics and Informatics,
Delft University of Technology, (1992).