

Solution of the discretized incompressible Navier-Stokes equations with the GMRES method

Report 91-24

C. Vuik



Technische Universiteit Delft
Delft University of Technology

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

ISSN 0922-5641

Copyright © 1991 by the Faculty of Technical Mathematics and Informatics, Delft, The Netherlands.

No part of this Journal may be reproduced in any form, by print, photoprint, microfilm, or any other means without permission from the Faculty of Technical Mathematics and Informatics, Delft University of Technology, The Netherlands.

Copies of these reports may be obtained from the bureau of the Faculty of Technical Mathematics and Informatics, Julianalaan 132, 2628 BL Delft, phone +31 15784568.

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](ftp://ftp.twi.tudelft.nl)

Solution of the discretized incompressible Navier-Stokes equations with the GMRES method

C. Vuik

Abstract

We describe some experiences using iterative solution methods of GMRES type to solve the discretized Navier-Stokes equations. The pressure equation is solved with full GMRES combined with a suitable preconditioner. The diagonally scaled momentum equations are solved by GMRES(m), a restarted version of GMRES.

1 Introduction

In [Van Kan et al., 1991] a numerical discretization of the incompressible Navier-Stokes equations in general curvilinear co-ordinates is treated. In this paper we present some experiences with an iterative solution method, GMRES, applied to these equations. We start with a description of our test problem. In Section 3 we present time measurements of relevant parts of the solution method. The solution of the pressure equation is discussed in Section 4. In the given test problem it appears that GMRES should be combined with a preconditioner to obtain a fast iterative method. Finally, in Section 5, we use GMRES to solve the momentum equations. We show that a correct scaling of the equations is important. Furthermore, we describe a termination criterion, which is combined with the GMRES method. We end this paper with conclusions in Section 6.

2 Statement of the problem

In this section we specify the incompressible Navier-Stokes equations and outline the discretization of these equations in time and space. Subsequently, we describe the geometry and initial and boundary conditions for a test problem, which is used in the remainder of this paper.

We consider the flow of an incompressible fluid in a two dimensional configuration. In [Van Kan et al., 1991] the Navier-Stokes equations, which can be used to describe this flow, are formulated in general co-ordinates. For the sake of simplicity we describe these equations in Cartesian co-ordinates:

$$\frac{\partial u_1}{\partial t} - \left(\frac{\partial \tau_{11}}{\partial x_1} + \frac{\partial \tau_{22}}{\partial x_2} \right) + \frac{\partial u_1^2}{\partial x_1} + \frac{\partial u_1 u_2}{\partial x_2} + \frac{\partial p}{\partial x_1} = 0, \quad (1)$$

$$\frac{\partial u_2}{\partial t} - \left(\frac{\partial \tau_{21}}{\partial x_1} + \frac{\partial \tau_{22}}{\partial x_2} \right) + \frac{\partial u_1 u_2}{\partial x_1} + \frac{\partial u_2^2}{\partial x_2} + \frac{\partial p}{\partial x_2} = 0,$$

where

$$\tau_{11} = Re^{-1} \left(\frac{4}{3} \frac{\partial u_1}{\partial x_1} - \frac{2}{3} \frac{\partial u_2}{\partial x_2} \right), \quad \tau_{12} = Re^{-1} \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \right),$$

$$\tau_{21} = Re^{-1} \left(\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2} \right), \quad \tau_{22} = Re^{-1} \left(\frac{4}{3} \frac{\partial u_2}{\partial x_2} - \frac{2}{3} \frac{\partial u_1}{\partial x_1} \right),$$

together with the incompressibility condition

$$\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} = 0, \quad (2)$$

and appropriate initial and boundary conditions. In these equations u_i is the component of the velocity of the fluid in x_i -direction, p is the pressure and Re is a parameter called the Reynolds number.

In the time discretization, finite differences are used. In this test problem we use an equidistant time discretization. We note that the discretization described in [Van Kan et al., 1991] is not necessarily equidistant. For a given positive integer N we define $k = T/N$. In the following v^n denotes the numerical approximation of $v(nk)$. Using the pressure correction method [Van Kan, 1986] we obtain the following equations:

$$\frac{\hat{u}_1^{n+1} - u_1^n}{k} - \left(\frac{\partial \hat{\tau}_{11}^{n+1}}{\partial x_1} + \frac{\partial \hat{\tau}_{12}^{n+1}}{\partial x_2} \right) + \frac{\partial (\hat{u}_1^{n+1})^2}{\partial x_1} + \frac{\partial \hat{u}_1^{n+1} \hat{u}_2^{n+1}}{\partial x_2} + \frac{\partial p^n}{\partial x_1} = 0, \quad (3)$$

$$\frac{\hat{u}_2^{n+1} - u_2^n}{k} - \left(\frac{\partial \hat{\tau}_{21}^{n+1}}{\partial x_1} + \frac{\partial \hat{\tau}_{22}^{n+1}}{\partial x_2} \right) + \frac{\partial \hat{u}_1^{n+1} \hat{u}_2^{n+1}}{\partial x_1} + \frac{\partial (\hat{u}_2^{n+1})^2}{\partial x_2} + \frac{\partial p^n}{\partial x_2} = 0,$$

$$\frac{\partial^2 p^{n+1}}{\partial x_1^2} + \frac{\partial^2 p^{n+1}}{\partial x_2^2} = \frac{\partial^2 p^n}{\partial x_1^2} + \frac{\partial^2 p^n}{\partial x_2^2} + \frac{1}{k} \left(\frac{\partial \hat{u}_1^{n+1}}{\partial x_1} + \frac{\partial \hat{u}_2^{n+1}}{\partial x_2} \right), \quad (4)$$

and, finally,

$$u_i^{n+1} = \hat{u}_i^{n+1} + k \frac{\partial (p^n - p^{n+1})}{\partial x_i}, \quad i = 1, 2. \quad (5)$$

The nonlinear terms $\hat{u}_i^{n+1} \hat{u}_j^{n+1}$ appearing in (3) are linearized (Newton linearization) as follows:

$$\hat{u}_i^{n+1} \hat{u}_j^{n+1} \cong \hat{u}_i^{n+1} u_j^n + u_i^n \hat{u}_j^{n+1} - u_i^n u_j^n \quad \text{with } i, j \in \{1, 2\}.$$

For the discretization in space the physical domain is mapped onto a rectangle (computational domain). Combining this co-ordinate transformation with finite volumes on a staggered grid in the computational domain we obtain a space discretization of the equations (3), (4) and (5). The operator div.grad in the pressure equation (4) is discretized by $\text{div}_h.\text{grad}_h$, where the same operators div_h and grad_h are used in the discretization of the momentum equations (3). For more details we refer to [Van Kan et al., 1991].

The resulting equations can be divided into two linear systems. The first system, the momentum equations, is a discretized version of (3), whereas the second system, the pressure equation, is a discretized version of (4). The space discretization is such that it is possible to implement algorithms, for building the matrices and solving the systems of equations, which are suitable for vector and parallel computers.

In the remainder of this paper we consider one test problem, which describes the flow through a curved channel. Our experiments with this problem give valuable insight into the behavior of GMRES solving the pressure and the momentum equations. The physical domain of the problem is displayed in Figure 1. Initially, the velocities are equal to zero. The boundary conditions are: a parabolic velocity profile, with the maximal velocity equal to one, on the inflow boundary (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 chose $Re = 500, T = 0.3$ and $N = 2$. For the space discretization the two squares (see Figure 1) are divided into 16×16 finite volumes, whereas the curve is divided into 16×32 finite volumes. The total number of finite volumes is $16 \times 64 = 1024$.

3 CPU time measurements

In this section some timing results are given. First of all we consider two simple loops and measure the megafloprate on three different computers. Thereafter, we determine the megafloprate of some typical routines used in the solution of the discretized Navier-Stokes equations.

Most loops used in the GMRES method (to be described in Section 4) are vector updates and inner products. Such loops are suitable for vector and parallel computers. We measure the megafloprate of such a loop, in order to compare the performance of the computers, which are used in our department.

vector update: $a(i) = b(i) + c * a(i), i = 1, \dots, 1000$.

In the following sections we use GMRES combined with an ILU preconditioner. This preconditioner makes use of recursions. It is known that recursions are bad for the performance of vector and parallel computers. To check this we also present time measurements for the loop:

recursion: $a(i) = b(i) + c * a(i - 1), i = 1, \dots, 1000$.

We use the following computers: HP 9000-845, Alliant FX/4, and Convex C240. The first has

only one scalar processor, whereas the second and the third consist of four parallel vector processors. The results are given in Table 1, where the operations $+$, $-$, and $*$ are each counted as one floating point operation (flop). These results suggest that the vector update runs in

	HP	Alliant 1 CPU	Alliant 4 CPU	Convex 1 CPU
vector update	2.2	2.4	5.5	10
recursion	2.2	0.83	0.83	2.6

Table 1: Megafloprates

vector speed and the megafloprate of this loop shows an increase on parallel processors, but as expected the recursion runs in scalar mode on all computers. Note that to get a fast solver of the Navier-Stokes equations on vector machines it is important to use vectorizable loops as much as possible.

operation	CPU time (s)	Megafloprate
construction of the system	0.12	
matrix vector multiplication	0.78×10^{-2}	8
vector update	0.49×10^{-3}	10
inner product	0.33×10^{-3}	16
diagonal preconditioning	0.44×10^{-3}	

Table 2: Computer time and megafloprates for parts of the algorithm to solve the momentum equations

Table 2 shows some timing results on the Convex for typical parts of the solver, which is used in the solution of the momentum equations. We emphasize that these results are only used for comparison. On different runs, deviations of 10% are observed. In the following sections we compare two different solution algorithms. The second differs from the first in the following sense: the amount of vector updates decreases, whereas the amount of matrix vector multiplications increases. For this comparison we note that the CPU time of a matrix vector multiplication is comparable to the CPU time of 16 vector updates.

In Table 3 we give the results for the pressure equation.

The construction of the linear system is not included in Table 3, because in contrast to the momentum equations, the pressure equation remains the same in every timestep (except for the right-hand side). Note that the CPU time for the vector update and inner product in Table 2 is two times the CPU time in Table 3. This agrees with the fact that the dimension of the momentum equations is two times the dimension of the pressure equation. Finally, it follows from Table 3 that the ILU preconditioner costs three times as much as one matrix vector multiplication. This corresponds with the fact that ILU uses recursions. In the near future a vectorizable variant of the ILU preconditioning will be implemented (see [Van der

operation	CPU time (s)	Megafloprate
matrix vector multiplication	0.27×10^{-2}	8
vector update	0.24×10^{-3}	11
inner product	0.15×10^{-3}	16
diagonal preconditioning	0.24×10^{-3}	
ILU preconditioning	0.76×10^{-2}	

Table 3: Computer time and megafloprates for parts of the algorithm to solve the pressure equation

Vorst, 1989)).

4 The pressure equation

In this section we start with a specification of the discretized pressure equation. After that we describe the GMRES method, which is used to solve this equation. We obtain valuable insights from the convergence properties of full GMRES. However, full GMRES is expensive with respect to computing time and memory requirements. A fast iterative method is obtained by a combination of GMRES with a polynomial or an ILU preconditioner. We end the section with a remark on the memory requirements of this iterative solution method.

The physical domain of our test problem is the curved channel of Figure 1. The computational domain, which is used in the discretization of the pressure equation (4) is a rectangle, of which the edges are parallel to the co-ordinate axes. The orientation of the computational domain is such that $n_1 \leq n_2$, where n_i is the number of grid points in the x_i -direction.

We use a lexicographical ordering of the grid points, so the difference between the indices of neighboring grid points equals 1 in the x_1 -direction and n_1 in the x_2 -direction. The structure of the matrix $P \in \mathbb{R}^{(n_1 n_2) \times (n_1 n_2)}$ used in the discretized pressure equations is given in Figure 2. In this version of the discretization the matrix P is not symmetric [Van Kan et al., 1991; p. 44]. For a discretization of (4) such that the resulting matrix P is symmetric we refer to [Van Kan, 1991].

In this paper we solve the pressure equation $Px = b$ using the GMRES(m) method (see [Saad & Schultz, 1986]). In the GMRES method the vector z_k is chosen such that

$$z_k = \arg \min_{z \in K^k(P; r_0)} \| b - P(x_0 + z) \|_2, \quad (6)$$

where $r_0 = b - Px_0$ and the Krylov subspace $K^k(P; r_0)$ is defined by $K^k(P; r_0) = \text{span} \{r_0, Pr_0, \dots, P^{k-1}r_0\}$. In the following experiments we always take $x_0 = (0, \dots, 0)^T$. After m iteration steps the method restarts using $x_m = x_0 + z_m$ as start vector. For an implementation of this method we refer to [Saad & Schultz, 1986] and [Van der Vorst, 1989].

GMRES(50)

Using the insights given in [Huang & Van der Vorst, 1989] we choose the integer $m \in [5, 50]$ as large as possible with respect to memory requirements. In this example there are $16 \times 64 = 1024$ grid points, and it turns out to be possible to take m equal to 50. After 1220 iterations and 17 seconds CPU time on the Convex we obtain $\| r_{1220} \|_2 / \| r_o \|_2 \leq 10^{-6}$. This is a bad result, which motivates us to analyse the convergence behavior of the GMRES method in this application. To facilitate comparison we specify the amount of work and memory in Table 4.

Full GMRES

In the following experiment we solve the pressure equation with full GMRES, so the GMRES method is not restarted. It appears that $\| r_{184} \|_2 / \| b \|_2 \leq 10^{-6}$. This is a better result with respect to computing time but now the amount of required memory is (too) large (see Table 4).

Although full GMRES uses too much memory in this application, the results of this experiment give valuable information. From the local convergence behavior (see Figures 3 and 4) it appears that in the first hundred iterations the residual remains nearly the same. Taking into account the analysis of the convergence behavior of GMRES given in [Huang & Van der Vorst, 1989] and [Van der Vorst & Vuik, 1991] we conclude that it takes more than 100 iterations before the Ritz values approximate the eigenvalues, which results in a super linear convergence behavior of GMRES in a later phase. This explains that restarting GMRES is a bad idea in this application, because if we choose m less than 100, GMRES is only linearly convergent, and the reduction factor is nearly 1. This agrees with the results obtained with GMRES(50).

Polynomial Preconditioning

From these experiments it follows that a preconditioner is necessary to get an acceptable computing time and a reasonable amount of memory required. Since a polynomial preconditioner has good vectorization properties we start with such a preconditioner. In [Nachtigal, Reichel & Trefethen, 1990] and [Saylor & Smolarski, 1990] a combination of GMRES with a polynomial preconditioner is presented.

Using GMRES combined with a polynomial preconditioner we can not expect to need fewer matrix vector products than using full GMRES, because GMRES has the minimal residual property (6). So the main purpose of a polynomial preconditioner is to lower the number of vector updates and inner products, and the number of memory required.

In this paragraph we describe a preconditioner, where the polynomial is adapted in every iteration. We get the idea for this preconditioner from the EN method as given in [Eirola & Nevanlinna, 1989] and analysed in [Vuik & Van der Vorst, 1990]. In the EN method one approximates P^{-1} in every iterate by the polynomial $(I - P)$. We propose to approximate P^{-1} by a polynomial of higher degree, which is obtained by a call of full GMRES. The resulting

method, which we denote by GMRES² is given as follows (compare [Vuik & Van der Vorst, 1991]):

1. $u_0 = H_0 r_0 / \| P H_0 r_0 \|_2$, $c_0 = P u_0$, $k = 0$,
 $x_1 = x_0 + u_0 c_0^T r_0$ and $r_1 = r_0 - c_0 c_0^T r_0$;
 2. while $\| r_{k+1} \|_2 > \text{eps}$ do $k := k + 1$,
 $c_k^{(0)} = P H_k r_k$, $u_k^{(0)} = H_k r_k$
 $\beta_i = c_i^T c_k^{(i+1)}$, $c_k^{(i+1)} = c_k^{(i)} - \beta_i c_i$, $u_k^{(i+1)} = u_k^{(i)} - \beta_i u_i$, $i = 0, \dots, k - 1$,
 $c_k = c_k^{(k)} / \| c_k^{(k)} \|_2$, $u_k = u_k^{(k)} / \| c_k^{(k)} \|_2$,
 $x_{k+1} = x_k + u_k c_k^T r_k$ and $r_{k+1} = r_k - c_k c_k^T r_k$,
- end.

In this algorithm $H_k r_k$ denotes the solution of the system $P y_k = r_k$ with mpost iterations of full GMRES.

Application of GMRES² with mpost = 10 gives $\| r_{18} \|_2 / \| b \|_2 \leq 10^{-6}$. Since in every iteration of GMRES² we use 11 matrix vector products we obtain the solution with 198 matrix vector products. Comparing this with GMRES(50) and full GMRES we conclude that GMRES² has a favorable convergence behavior in this application.

It follows from Table 4 that the number of vector updates and inner products shows a considerable reduction, with respect to full GMRES. These observations explain the large reduction of the computing time.

With respect to memory requirements we note that using m iterations of GMRES² we need $2m + \text{mpost}$ vectors in memory. In this experiment $m = 18$ and mpost=10 so we need 46 vectors in memory, which is comparable with GMRES(50).

Note that both aims of polynomial preconditioning are achieved: reduction of the vector updates and inner products and a reduction of the memory required. The extra cost of this approach, 14 extra matrix vector multiplications, is negligible.

Application of GMRES² with mpost=15 or 20 gives comparable results. So the efficiency of the algorithm is not very sensitive to variation of mpost.

ILU preconditioning

Another successful preconditioning technique is to construct an Incomplete $LD^{-1}U$ Decomposition of P and to solve the system $LD^{-1}UPx = LD^{-1}Ub$ instead of $Px = b$ (see [Meyerink and Van der Vorst, 1977] and [Van der Vorst, 1981]. To implement this preconditioning we use the following rules to obtain L , D and U [Van der Vorst, 1981]:

- (a) $\text{diag}(L) = \text{diag}(U) = D$;
- (b) the off-diagonal parts of L and U are equal to the corresponding parts of P ;

(c) $\text{diag}(LD^{-1}U) = \text{diag}(P)$.

method	matvec	vector update	inner product	memory vectors	CPU time (s)	
					Convex	precondition matvec
GMRES(50)	1220	35000	35000	50	17	0
Full GMRES	184	17000	17000	184	7.2	0
GMRES ²	198	1386	1224	46	1.2	0
ILU	53	1405	1405	53	1.2	53
MILU($\alpha = 0.95$)	31	481	481	31	0.6	31

Table 4: The amount of work, memory, and CPU time

Note that this preconditioning needs only one extra vector in memory to store the diagonal matrix D .

Combination of the ILU preconditioner with full GMRES gives $\|r_{53}\|_2 / \|r_0\|_2 \leq 10^{-6}$. The resulting computing time and memory requirements are comparable with GMRES² (see Table 4). Note that in contrast to GMRES² we have not yet implemented a vectorizable version of the ILU preconditioning. Using such a version the computing time should be less than the computing time using GMRES².

MILU preconditioning

Finally we use the MILU preconditioning specified in [Gustafsson, 1978]. In this preconditioning the construction of D is such that the rowsum of $LD^{-1}U$ equals the rowsum of P . In the following experiments we use an average of the ILU and the MILU preconditioner [Axelsson & Lindskog, 1986], [Van der Vorst, 1989]. The ILU preconditioner corresponds with $\alpha = 0$, whereas the MILU preconditioner corresponds with $\alpha = 1$.

The number of iterations of MILU combined with full GMRES such that $\|r_i\|_2 / \|r_0\|_2 \leq 10^{-6}$ is given in Table 5 for different choices of α . From Table 4 it appears that this method, with $\alpha = 0.95$, is optimal with respect to the amount of work, required memory and computing time. In this application combination of MILU and polynomial preconditioning gives only a small reduction of the computing time and memory requirements.

α	0	0.9	0.925	0.95	0.98	1
iterations	53	33	32	31	40	48

Table 5: Number of iterations for different α

Memory requirements

From these experiments, it appears that GMRES combined with MILU preconditioning needs 31 vectors in memory. This is a large amount of memory with respect to the memory requirements of for instance CGS [Sonneveld, 1989]. However, the momentum equations are built every timestep so the memory required to store and solve the momentum equations can be used to solve the pressure equations. In Section 5 we shall show that the momentum matrix consists of 13 vectors, which are twice as long as the vectors used in the pressure equations. So, we do not need much extra memory to solve the pressure equations using full GMRES combined with a MILU preconditioner.

5 The momentum equations

In this section we specify the momentum equations. An application of full GMRES shows that the momentum matrix is "nearly" singular. It appears that this is a consequence of a wrong scaling of the equations. We show that a diagonally scaled version of the momentum matrix has much better properties. Thereafter we discuss some termination criteria. Finally, the section is concluded by some experiments with the GMRES(m) method.

The discretized momentum equations (see equation (3)) are denoted by $Mx = b$, where $M \in \mathbb{R}^{(2n_1 n_2) \times (2n_1 n_2)}$ and $x, b \in \mathbb{R}^{2n_1 n_2}$. The structure of the matrix M is given in Figure 5. From equation (3) it follows that the matrix M depends on the space- and the time-discretization, the Reynolds number and the velocity of the fluid in the preceding timestep. This implies that, in general, M is different in every timestep, therefore in the following the momentum matrix in the n^{th} timestep is denoted by M_n . From our experiments it appears that $M_1 x = b$ can be solved with a small number of GMRES iterations. In order to give an explanation, we note that the initial condition and the choice of Re imply that M_1 is approximately equal to a scalar times the identity matrix. It is known that GMRES converges fast for such a linear system. Since it appears that the convergence behavior of GMRES applied to $M_n x = b$ is more or less the same for every $n \geq 2$, we consider the system $M_2 x = b$ in the remainder of this section.

Full GMRES

As in Section 4 we apply full GMRES to the momentum equations to obtain insights into its convergence properties. To solve $M_2 x = b$ we start GMRES with $x_0 = 0$ and stop when $\|r_i\|_2 \leq 10^{-8}$. In this experiment GMRES satisfies the stopping criterion after 135 iterations. The converge history is given in Figure 6. Note that the convergence stagnates from iteration 50 until iteration 70. To obtain more insight into this stagnation phase we calculate the Ritz values, which are plotted in the Figures 7, 8 and 9. From these figures and the analysis given in [Van der Vorst & Vuik, 1991] we conclude the following: It appears from Figure 9 that there is a "small" eigenvalue, which means that its modulus is small with respect to the moduli of the other eigenvalues. Initially, the component in the corresponding eigenvector is small so there is no "small" Ritz value in Figure 7 and the convergence is fast (see Figure 6). However,

in the stagnation phase the process discovers that there is a "small" eigenvalue (Figure 8). After iteration 70 the "small" Ritz value is converged (Figure 9) and GMRES converges as if the "small" eigenvalue is absent, which corresponds with the results given in [Van der Vorst & Vuik, 1991].

Where does this "small" eigenvalue come from? In the discretization the Dirichlet boundary conditions are included as equations in the system $M_2x = b$. However, these extra equations are not scaled with respect to the other ones. As a result of this the main diagonal elements in these equations, which are equal to one, differ a factor 10^3 to 10^4 with the other nonzero main diagonal elements. To get rid of these "small" eigenvalues we propose the following approaches:

1. do not include the Dirichlet boundary conditions as extra equations,
2. take the start vector x_0 such that it already satisfies the extra equations,
3. scale the momentum equations with the inverse of the nonzero main diagonal elements.

In the following paragraphs we analyse these approaches in more detail.

Approach 1

The convergence should be better without the extra equations. However, since it is not easy to implement this in our discretization we have not experimented with this approach.

Approach 2

This approach is the same as approach 1 in the sense that the extra equations, and thus the "small" eigenvalues, no longer influence the convergence behavior of GMRES. From an experiment we obtain $\|r_{112}\|_2 \leq 10^{-8}$. So this approach saves 25 iterations. The local convergence behavior is given in Figure 10. Note that there is no stagnation phase (compare Figure 6). Calculation of the Ritz values, which are plotted in Figure 11, shows that "small" Ritz values are absent.

Approach 3

It is known ([Saad & Schultz, 1986] and [Huang & Van der Vorst, 1989]) that the convergence of GMRES depends on the convex hull of the eigenvalues. After the scaling of the equations we expect that all eigenvalues are clustered around one, which implies that the extra equations do not influence the convergence of GMRES. Since $\|r_0\|_2$ depends on the scaling of the equations, GMRES is stopped after the ratio of the norm of the i^{th} residual and the initial residual is the same as for the original process. Using this stopping criterion and starting with $x_0 = 0$, GMRES stops after 107 iterations. The local convergence behavior is more or less the same as that given in Figure 10. In Figure 12 we plot the Ritz values of GMRES combined with diagonal scaling. Note that the eigenvalue equal to one, which comes from the Dirichlet boundary condition, is inside the convex hull of the eigenvalues, as expected, so it does not influence the convergence of GMRES.

A termination criterion

A good termination criterion is important for iterative solution methods. In this paragraph we discuss the following criteria:

- $\| r_k \|_2 \leq \text{eps}$
The main disadvantage of this criterion is that it is not scaling invariant.
- $\| r_k \|_2 / \| r_0 \|_2 \leq \text{eps}$
This criterion is scaling invariant, however, the number of iterates is independent of the initial estimate x_0 . This is a drawback because we expect that after some time the solution of the foregoing timestep is a good starting solution.
- $\| r_k \|_2 / \| b \|_2 \leq \text{eps}$
This is a good stopping criterion.
- $K_2(A) \| r_k \|_2 / \| b \|_2 \leq \text{eps}$, where $K_2(A)$ is the condition number of A .
This is the best termination criterion. From the inequality

$$\frac{\| x - x_k \|_2}{\| x \|_2} \leq K_2(A) \frac{\| r_k \|_2}{\| b \|_2}$$

it follows that $\frac{\| x - x_k \|_2}{\| x \|_2} \leq \text{eps}$. However, in general, $K_2(A)$ is not known. To obtain an estimate for $K_2(A)$ we propose to calculate the singular values of $R_k \in \mathbb{R}^{(k+1) \times k}$. R_k is defined in [Saad & Schultz, 1986; p. 861]. An underestimate of $K_2(A)$ is given by the ratio $\sigma_1^{(k)} / \sigma_k^{(k)}$ where $\sigma_1^{(k)}$ is the largest and $\sigma_k^{(k)}$ is the smallest singular value of R_k .

In Figure 13 we plot the estimate from full GMRES applied to the original momentum equations. Note that initially the estimate is small but after 50 iterations the estimate increases to 10^4 . This corresponds with the results given in Figure 6.

In Figure 14 we give the estimate from full GMRES combined with a diagonal scaling of the momentum equations. This estimate remains small. Furthermore, it appears that after a small number of iterations the difference between the estimate and its final value is rather small. This motivates us to use the following termination strategy in the diagonal scaled momentum equations:

```

if      k = 10      calculate the estimate  $\hat{K}_2(\bar{A}) = \sigma_1^{(10)} / \sigma_{10}^{(10)}$ ,
if      k < 10      then
    stop           when       $\| r_k \|_2 / \| b \|_2 \leq \text{eps}$ 
else
    stop           when       $\hat{K}_2(A) \| r_k \|_2 / \| b \|_2 \leq \text{eps}$ 
endif

```

The time to estimate $\hat{K}_2(A)$ is less than 0.006 s CPU time on the CONVEX. Comparing this with Table 2 we conclude that this overhead is negligible.

GMRES(m) with diagonal scaling

From the local convergence behavior of full GMRES (Figure 10) we expect that GMRES(m) is a good iterative method. In Table 6 we give some experiments with $\text{eps} = 10^{-5}$ and different choices of m . In these experiments $\|r_0\|_2 = 0.92$ and $\|b\|_2 = 2$.

m	iterations	matvec	vector update	inner product	memory vectors	$\ r\ _2$	CPU time (s) CONVEX
53	53	53	1431	1431	53	0.44×10^{-5}	1.24
20	55	57	520	520	20	0.37×10^{-5}	0.72
10	55	60	263	263	10	0.44×10^{-5}	0.67
5	57	68	144	144	5	0.51×10^{-5}	0.62
3	62	82	100	100	3	0.63×10^{-5}	0.69

Table 6: The amount of work, memory, and CPU time

If $k = \min(10, m)$ we calculate the estimate $\hat{K}_2(A)$. This implies that $\hat{K}_2(A)$ is less for $m = 5$ ($\hat{K}_2(A) = 3.4$) and $m = 3$ ($\hat{K}_2(A) = 2.9$) than for $m \geq 10$ ($\hat{K}_2(A) = 4.2$). Since this estimate is used in the stopping criterion we expect larger norms of the final residual if m is less than 10 (see Table 6). It follows from Table 6 that if m decreases the amount of vector updates and inner products decreases, whereas the amount of matvec's increases. We observe from $m = 10$ and $m = 3$ that the decrease in computing time with respect to the vector updates and inner products is equal to the increase in computing time with respect to the matrix vector multiplications (compare Table 2, which implies that the costs of one matvec is equal to the cost of ten vector updates and inner products). Finally, we note that for the optimal choice ($m = 5$) the required number of vectors in memory is rather small. For m less than 6 the required memory is comparable with the required memory of CGS [Sonneveld, 1989].

6 Conclusions

In this paper we have described properties of GMRES-type iterative methods to solve a discretization of the Navier-Stokes equations. Our experiments consist of two parts: in the first part we solve the pressure equation and in the second part we solve the momentum equations.

Solving the pressure equation we note that only full GMRES gives a good iterative solution method. Restarting destroys the superlinear convergence behavior of GMRES [Van der Vorst & Vuik, 1991]. From these results it follows that a preconditioning is necessary to obtain reasonable computing time and memory requirements. It appears that polynomial preconditioning and (M)ILU preconditioning give good results. The required memory used in full GMRES combined with a MILU preconditioner is available because the memory required to store the momentum matrix can be used. Note that the momentum matrix is built anew every timestep.

Full GMRES applied to the momentum equations reveals that the scaling of the equations as originally given is unfavorable. Diagonal scaling of the matrix gives much better results. A termination criterion is proposed such that the norm of the relative error in the solution vector is less than a prescribed accuracy. Using diagonal scaling and this termination criterion we present results of some experiments with GMRES(m). It appears that a small m is sufficient, which implies that only a small amount of extra memory is required. This amount is comparable with the amount of memory used in the CGS method.

References

- [1] Axelsson, O. and Lindskog, G.
On the eigenvalue distribution of a class of preconditioning methods.
Numer. Math. **48**, 479-498, (1986).
- [2] Eirola, T. and Nevanlinna, O.
Accelerating with rank-one updates.
L.A.A. **121**, 511-520, (1989).
- [3] Gustafsson, I.A.
A class of first order factorization methods.
BIT **18**, 142-156, (1978).
- [4] Huang, Y. and Vorst, H.A. van der
Some observations on the convergence behavior of GMRES.
Report 89-09, Faculty of Technical Mathematics and Informatics, Delft University of Technology, (1989).
- [5] Kan, J.J.I.M. van
A second-order accurate pressure-correction scheme for viscous incompressible flow
SIAM J. Sci. Stat. Comput. **7**, 870-891, (1986).
- [6] Kan, J.J.I.M. van
A symmetric pressure matrix in general coordinates.
Report 91- , Faculty of Technical Mathematics and Informatics, Delft University of Technology, (1991).
- [7] Kan, J.J.I.M. van, Oosterlee, C.W., Segal, A. and Wesseling, P.
Discretization of the incompressible Navier-Stokes equations in general coordinates using contravariant velocity components.
Report 91-09, Faculty of Technical Mathematics and Informatics, Delft University of Technology, (1991).
- [8] Meijerink, J.A. and Vorst, H.A. van der
An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix.
Math. of Comp. **31**, 148-162, (1977).

- [9] Nachtigal, N.M., Reichel, L. and Trefethen, L.N.
A hybrid GMRES algorithm for non symmetric linear systems.
In: Proceedings of the Copper Mountain Conference on Iterative Methods, April 1-5, 1990, Book 3.
- [10] Saad, Y. and Schultz, M.H.
GMRES: a generalized minimal residual algorithm for solving non symmetric linear systems.
SIAM J. Sci. Stat. Comput. **7**, 856-869, (1986).
- [11] Saylor, P.E. and Smolarski, D.C.
Implementation of an adapted algorithm for Richardson's method.
Report 139, Departement Informatik Institut für Wissenschaftliches Rechnen, Eidgenössische Technische Hochschule Zürich, (1990).
- [12] Sonneveld, P.
CGS: a fast Lanczos-type solver for nonsymmetric linear systems.
SIAM J. Sci. Stat. Comp. **10**, 36-52, (1989).
- [13] Vorst, H.A. van der
Iterative solution method for certain sparse linear systems with a non-symmetric matrix arising from PDE-problems.
J. Comp. Phys. **44**, 1-19, (1981).
- [14] Vorst, H.A. van der
High performance preconditioning
SIAM J. Sci. Stat. Comput. **10**, 1174-1185, (1989).
- [15] Vorst, H.A. van der
The convergence behavior of some iterative solution methods.
In: Proceedings of the fifth International Symposium on Numerical Methods in Engineering **1**, 61-72.
Eds.: Gruber, P., Periaux, J., Shaw, R.P.
Springer Verlag, Berlin, (1989).
- [16] Vorst, H.A. van der and Vuik, C.
The rate of convergence of the GMRES method.
Submitted to SIAM J. Matrix An. Appl., (1991).
- [17] Vuik, C. and Vorst, H.A. van der
A comparison of some GMRES-like methods.
To appear in L.A.A., (1991).

Figure 1: The physical domain of the test problem

Figure 2: The pressure matrix P

Figure 3: Full GMRES

Figure 4: Full GMRES

Figure 5: The momentum matrix M , where $nu = n_1.n_2$

Figure 6: Full GMRES

Figure 7: Ritz values after 30 iterations

Figure 8: Ritz values after 50 iterations

Figure 9: Ritz values after 70 iterations

Figure 10: Full GMRES without the influence of the Dirichlet boundary condition

Figure 11: Ritz values after 112 iterations without the influence of the Dirichlet boundary conditions

Figure 12: Ritz values after 107 iterations of GMRES combined with a diagonal scaling

Figure 13: The estimate of the condition of M obtained from full GMRES

Figure 14: The estimate of the condition of M obtained from full GMRES combined with a diagonal scaling