A simple iterative linear solver for the 3D incompressible Navier-Stokes equations discretized by the finite element method

Report 95-64

Guus Segal
Kees Vuik

TU Delft
Technische Universiteit Delft
Delft University of Technology

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

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, ftp.twi.tudelft.nl. They are located in directory /pub/publications/tech-reports. They can also be accessed on the World Wide Web at:
http://www.twi.tudelft.nl/TWI/Publications/Overview.html

A simple iterative linear solver for the 3D incompressible Navier-Stokes equations discretized by the finite element method

Guus Segal and Kees Vuik
Faculty of Technical Mathematics and Informatics,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands,
e-mail: g.segal@math.tudelft.nl c.vuik@math.tudelft.nl

Abstract

In this paper we consider the solution of the systems of non-linear equations arising from the discretization of the incompressible Navier-Stokes equations. These equations are linearized by a combination of Picard and Newton iterations. The resulting systems of linear equations suffer from the occurrence of zero elements on the main diagonal. In this paper we describe some ordering techniques which allow us to renumber these equations a-priori such that direct solvers may be used without pivoting. Combination of these ordering techniques with standard ILU preconditioning makes it possible to solve the system of equations by standard iterative methods. In this way a considerable amount of computation time and memory is saved.

1 Introduction

The solution of the incompressible Navier-Stokes equations has been a challenging problem for the last decades. Important practical problems are for example the construction of turbulence models and the viscosity model to be used. In this paper, however, we shall limit ourselves to laminar flow with a Newtonian viscosity model.
One of the important problems related to the incompressible Navier-Stokes equations is that the pressure is absent in the continuity equation. Since the continuity equation is strongly related to the pressure this implies that the systems of linear equations to be solved have zeros at the main diagonals for the rows corresponding to the pressure unknowns. Such a property makes the solution of the system of equations in general a

difficult task.
We shall restrict ourselves to the solution of the coupled Navier-Stokes equations by the finite element method, although other discretization techniques like finite volumes have the same problems and presumably our way of solving may also be applied to those types of discretization. Furthermore we restrict ourselves to three-dimensions since for that class of problems the need for iterative solution methods is most urgent.

In Section 2 we consider the Navier-Stokes equations and the discretization by the finite element method. The appearance of the zero diagonal elements is shown.
Section 3 recalls some common techniques to overcome the problem of the zero pivots. Advantages and disadvantages of these methods will be summarized. In Section 4, we describe and investigate some orderings, where the zero main diagonal elements become non zero elements during the decomposition of the matrix. In our numerical experiments we observe no break down of the direct method without pivoting.
Thereafter, in Section 5, we specify our iterative method. The effect of scaling of the continuity equation on the rate of convergence of the iterative method is investigated. Two incomplete LU decompositions are given, which are used as preconditioners for Krylov subspace methods. Finally the influence of the ordering of the unknowns is investigated.

The described methods are compared numerically in Section 6. Even for small problem sizes the preconditioned Krylov subspace methods appear to be much better than the direct or penalty methods. For large problem sizes it is impossible to use direct or penalty methods due to excessive memory requirements. Furthermore for medium problem sizes the CPU time for a direct or penalty method is orders of magnitudes larger than that for the iterative method.

2 Discretization of the Navier-Stokes equations

In this section we consider the discretization of the incompressible Navier-Stokes equations by finite element methods. For the sake of simplicity we restrict ourselves to time-independent equations. However, we may expect that the techniques we derive in this paper will also be suitable for the time-dependent case, since then the matrices involved become more diagonally dominant.
The general form of the stationary Navier-Stokes equations may be written as

$$-div\ \boldsymbol{\sigma} + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{f}, \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress tensor:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla \mathbf{u}^T + \nabla \mathbf{u}). \tag{2}$$

μ denotes the viscosity, $\mathbf{f}$ is some external force and $\mathbf{u} \cdot \nabla \mathbf{u}$ represents the convective terms. The continuity equation is given by

$$div\ \mathbf{u} = 0. \tag{3}$$

In order to have a unique solution it is necessary to prescribe boundary conditions. Common types of boundary conditions are prescribed velocities (Dirichlet boundary conditions) and prescribed stresses or combinations of these two.

Equations (1) to (3) are solved by a standard finite element method based upon the Galerkin formulation as can be found in for example Cuvelier et al [7].
To that end the weak formulation is derived by multiplying the equations (1) and (3) by test functions for the velocity and pressure respectively. Application of the Gauss divergence theorem leads to

$$\int_{\Omega} \boldsymbol{\sigma} \cdot \nabla \delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} \cdot \delta \mathbf{u} d\Omega = \int_{\Omega} \mathbf{f} \cdot \delta \mathbf{u} d\Omega + \int_{\Gamma_{stress}} \sigma_n \delta u_n + \boldsymbol{\sigma}_t \cdot \delta \mathbf{u}_t d\Gamma, \tag{4}$$

and

$$\int_{\Omega} div\ \mathbf{u} \cdot \delta p d\Omega = 0. \tag{5}$$

Ω is the definition region and Γ_{stress} is that part of the boundary where stresses are prescribed. The vector $\mathbf{n}$ denotes the outward normal and $\mathbf{t}$ a tangential vector at the boundary. Vector $\delta \mathbf{u}$ represents the test functions for the velocity and δp the test functions for the pressure.
In this paper we restrict ourselves to the three-dimensional case. The elements used are the triquadratic isoparametric Crouzeix Raviart hexahedrons [7]. This means that the velocity is approximated by a quadratic polynomial in each direction in the reference element and the pressure is approximated by a linear polynomial, which is discontinuous over the element boundaries.
The Galerkin formulation is derived by substituting a linear combination of basis functions for velocity and pressure and the complete set of basis functions for the test functions.

Since the convective terms are non-linear it is necessary to use some linearization scheme. Commonly used schemes are the Picard iteration where the convective terms at the new level are approximated by:

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \approx \mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k+1}, \tag{6}$$

and the Newton scheme:

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \approx \mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k+1} + \mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k} - \mathbf{u}^{k} \cdot \nabla \mathbf{u}^{k}. \tag{7}$$

k denotes the old iteration level and $k+1$ the new one.
A common way of iteration is to use the solution of the Stokes equations (i.e. the Navier-Stokes equations without convective terms) as initial guess, proceed with a number of

so-called Picard iterations and finally apply Newton linearization.
In each step of the iteration process, it is necessary to solve a large system of linear equations. Formally this system of equations can be written as:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{8}$$

where A denotes the discretization of the stress tensor and the linearized convection terms, $B^T p$ the discretization of the pressure gradient and $Bu = 0$ the discretization of minus the continuity equation. The vector u represents the velocity unknowns and the vector p the unknown pressures.
Equations of the type (8) are easily recognized as corresponding to a saddle-point problem. These equations are characterized by the presence of a large zero matrix at the diagonal. The complete matrix is not symmetric positive definite even in the absence of convective terms. This type of equations is in general much more difficult to solve than the classical matrices arising from standard discretizations.
In the sequel we shall refer to the zero matrix at the main diagonal as the *zero-block*. This zero matrix is due to the fact that the pressure is not present in the continuity equation, whereas the equation itself is coupled to the pressure unknowns. This property is inherent to incompressible Navier-Stokes equations and is independent of the type of discretization.

3 Solution methods for the discretized Navier-Stokes equations

In the literature many attempts have been made to solve the equations arising from the discretization of the Navier-stokes equations. Of course one can try to solve equations of the type (8) immediately by some direct or iterative solver. However, a straight forward direct solver may have some problems with the zeros at the main diagonal, depending on the ordering of the equations. We shall return to this matter in Section 4. Also standard iterative solvers usually converge very slowly or even do not converge at all in case of saddle-point problems. For that reason many alternative formulations are used in practice to prevent the *zero-block*. Most of them are based on a segregation of pressure and velocity. We shall shortly recall some popular methods and mention advantages and disadvantages.

In the context of finite element methods, the penalty function method is very popular. In this approach the continuity equation is perturbed by a small parameter ε times the pressure:

$$\varepsilon p + div\ \mathbf{u} = 0. \tag{9}$$

From equation (9) the pressure may be eliminated and hence pressure and velocity computation are segregated. One can prove ([7]) that the perturbed solution approximates

the original solution provided the parameter ε is small enough. In the discrete version the penalty function formulation corresponding to equation (8) becomes:

$$p = -\frac{1}{\varepsilon} M^{-1} B u, \tag{10}$$

$$(A + \frac{1}{\varepsilon} B^T M^{-1} B) u = f, \tag{11}$$

where M is the pressure mass matrix. In many practical problems M is approximated by a diagonal matrix.
From equations (10) and (11) it is clear that first the velocity and then the pressure is solved. The difference between perturbed solution and original solution is usually of the order of ε. The size of the matrix to be solved by the penalty function method is much smaller than that of the original problem. Furthermore the *zero-block* has been disappeared. As a consequence also the solution time is reduced considerably.
A disadvantage of the penalty method is that the choice of the parameter ε depends on the magnitude of the pressure. In practice ε is chosen between 10^{-5} and 10^{-9}. Another more severe problem is that the matrix to be inverted is very ill-conditioned due to the factor $\frac{1}{\varepsilon}$. As a consequence iterative methods do not converge and only direct methods may be applied. For three-dimensional problems this means that large amounts of computing time and memory are necessary.

In finite volume and finite difference techniques the pressure correction technique ([5]) or variants of it are very popular. With some minor adaptations this method is also applied in finite element methods ([9]). The pressure-correction method is in fact developed for time-dependent problems. Roughly speaking the pressure-correction method can be formulated as a prediction step in which the velocity is estimated from the momentum equations using the pressure at the previous time-level but without taking into account the continuity equation. Next the velocity is projected onto the space of divergence-free vector fields, resulting in a Laplacian-like equation for the pressure ([25]) and the velocity is updated. The matrices to be inverted have nice properties and both direct and iterative solvers may be applied. A disadvantage of the pressure-correction method is that this method converges slowly if a steady state must be reached. For time-dependent problems, however, it is a very attractive method.

In the context of finite element methods the solenoidal approach using basis functions that are approximately divergence-free have been proposed by Griffiths ([15]) and Thomasset ([22]) for two-dimensional problems and Hecht ([16]) for three-dimensions. Conceptually this approach seems the most attractive, however, the construction of the basis functions is not straight-forward especially in three-dimensions and the method does not seem to be very popular.

Another approach is based on the Uzawa scheme [13] developed for optimisation problems

with constraints, just as the penalty function method. This method seems to become more popular for the iterative solution of the (Navier-)Stokes equations ([18], [12]). It is based on the dual problem; instead of eliminating the pressure one tries to eliminate the velocity and solve the pressure equations. In terms of equation (8) this means:

$$u = A^{-1}(f - B^T p), \tag{12}$$

$$BA^{-1}B^T p = BA^{-1}f. \tag{13}$$

From equations (12) and (13) it is possible to derive the following iterative method:

$$Au^n = f - B^T p^n, \tag{14}$$

$$p^{n+1} = p^n + \rho B u^n, \tag{15}$$

where n denotes the n^{th} iteration level. By a clever choice of ρ or a sequence of different ρ's in combination with a suitable preconditioning one may arrive at a convergent method. Unfortunately the Uzawa scheme converges slowly for high Reynolds numbers and is therefore not robust enough for practical applications.

In the next section we shall define an a priori ordering technique that makes the a posteriori pivoting of the system of equations unnecessary in case of a direct solution technique. This ordering has been constructed such that a nearly optimal profile is still present. The same ordering technique will be used in combination with an ILU preconditioner in order to get a fast converging iterative method.

4 Optimal ordering of the unknowns without the necessity of pivoting

The system of linear equations (8) may be solved by either a direct method (LU-decomposition) or an iterative method. In this section we restrict ourselves to direct methods.

The notation used in equation (8) suggest that the unknowns are ordered in the sequence: all velocity unknowns and then all pressure unknowns. In the sequel we shall denote this ordering by p-last ordering. Since pressure and velocity are locally coupled such an ordering produces a large profile and is therefore very uneconomical for a direct solver. A much smaller profile will be produced if the unknowns are ordered in the sequence of the nodal points, where first all unknowns of node 1 are used, then of node 2 etcetera, provided the nodal points are numbered such that an optimal profile is possible.
Unfortunately the *zero-block* complicates things considerably. Due to boundary conditions it may be possible that the first diagonal element in the matrix is zero and as a consequence straight-forward LU-decomposition is not possible. Hence it may be necessary to use a kind of pivoting strategy, which influences the structure of the matrix and makes

the a priori estimation of the memory required to store the matrix a difficult task. So in fact one would like to have some a priori numbering of nodal points and unknowns that prevents the presence of a zero diagonal element during the elimination process and moreover, produces a kind of optimal profile. This numbering must be such that the first unknowns are velocity degrees of freedom independent of the type of boundary conditions. Furthermore it must have a very small influence at the local band width (profile).

Let us, for the sake of the argument, consider the simple rectangular domain of Figure 1, subdivided into quadrilaterals. We assume that the velocity is approximated by a bi-linear polynomial based on the velocities in the vertices and a constant pressure per element based on the pressure in the centroid. Furthermore we assume that the velocity is prescribed at

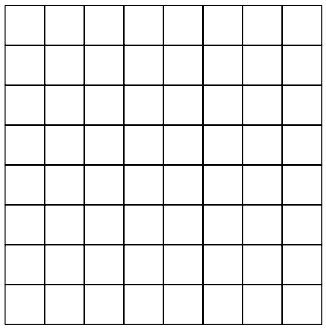

Figure 1: Rectangular domain, subdivided into quadrilaterals

the complete boundary. This means implicitly that the pressure is fixed upon an additive constant. The fact that this element does not satisfy the so-called B-B condition ([7]) is not important for our argument. If the nodes are numbered in a natural way, from left to right and line-wise from below to the upper boundary, it is clear that the first unknown is the pressure in the first element, since all velocity degrees of freedom corresponding to nodes with a lower sequence number are prescribed. The band width is determined by one stroke of elements only. One may expect that if we first number all velocities and then all pressures as suggested by equation (8), then during the elimination procedure the pressure diagonal elements become non-zero due to fill-in. What we want is an ordering of the unknowns such that pressure diagonal elements become non-zero during elimination, but without increasing the local band width (profile) considerably. Since the band width of this mesh is completely determined by the width of one stroke of elements, a natural ordering technique is to number the unknowns such that per stroke of elements first all velocities are numbered and then all pressures. Figure 2 shows the first three levels of

nodes.
In this way, due to fill in, the elimination may change the pressure diagonal elements to

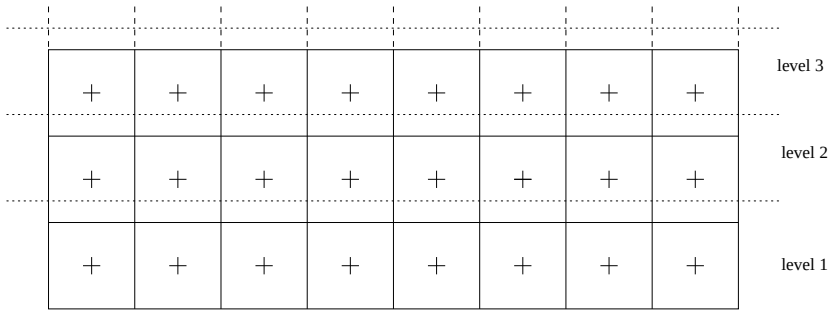

Figure 2: First three levels in rectangular region

non-zero elements and the actual band width is hardly changed. Such a renumbering will be called pressure last renumbering per level or p-last per level.

This renumbering is simple for a rectangular region. In a general irregular shaped region an automatic procedure is necessary to define the equivalent of the strokes of elements. To that end we assume that the nodal points have been renumbered before, in order to get a small profile for example by the standard reversed Cuthill-McKee renumbering algorithm [14] or the algorithm proposed by Sloan [21]. Next we define a level structure which is very similar to the Cuthill-McKee structure. We start with node 1 and find all neighbours of this node. Let node i_1 have the highest number of this set of neighbours. Then level 1 is defined as the set of nodes 1 to i_1. The next level is found by considering all new neighbours of level 1. Let node i_2 have the highest number of this new set. Then the next level is defined as the i_1+1 to i_2. This process is repeated until all neighbours are part of a level. With respect to the start it might be necessary to combine levels 1 and 2. Per level we number first the velocities and then the pressure degrees of freedom. In this way we get a nearly optimal numbering, which may be applied in combination with Gaussian elimination.
In practice it might be better to start with a row of points instead of one single point. If we start the Cuthill-McKee renumbering process with a boundary instead of a nodal point and use this boundary as start of the level structure we might get a numbering that is more similar to the natural numbering shown in Figure 2. In fact in this simple example it is exactly the type of level structure we get. Experiments with Navier-Stokes and convection-diffusion equations have shown that the outflow boundary forms in general a nice starting set.

We have applied this renumbering procedure to a number of test examples both two and three-dimensional problems. In all cases we had no problems with zero diagonal elements and moreover the profile was more or less optimal. However, compared to the penalty

function method, direct solution of the coupled system of equations by Gaussian elimination is still expensive. So only in the case that there are problems with the choice of ε this direct method is preferred.
The main reason to introduce this new renumbering technique is not to use it for direct methods, although they have inspired us to develop this renumbering but for the iterative methods treated in the next section.

5 Incomplete LU decompositions

In this section we consider the iterative solution of the linear system given in Section 2. We will solve this system by iterative methods of Krylov subspace type combined with preconditioners based on incomplete LU decompositions. For a recent overview of preconditioners and iterative methods we refer to [2]. After giving some known methods we will present our solution method for the coupled linear system. It will be shown that a scaling of the continuity equation by a multiplication factor may influence the rate of convergence of the iterative process without effecting the exact solution. Next two different ILU preconditioners will be defined, and some existence results given. We will conclude this section with an investigation of the influence of the ordering used on the convergence of the iterative method.

Discussion of known methods
In Section 3 an overview of the literature to circumvent the *zero-block* is presented. Here we discuss the literature with respect to iterative methods applied to the coupled momentum and continuity equations. In [18], [27], and [20] the discrete Stokes problem is solved by iterative methods using the p-last ordering. The preconditioners used are restricted to block diagonal forms, so the preconditioner of the velocity part is independent of the pressure part. In our approach we use a preconditioner based on the coupled velocity and pressure part.
In [1] an inner outer iteration scheme is given to solve the Stokes problem. This method is closely related to the classic Uzawa algorithm. Another class of methods is formed by the distributive iterative methods, see [28] for an overview. In these methods the difficulties of the *zero-block* are circumvented by multiplying the coefficient matrix from the right by a postconditioner such that the product matrix is an M-matrix.
Dahl and Wille present a preconditioner based on the coupled equations in [8]. The Navier-Stokes equations are discretized by Taylor-Hood elements. In their experiments, Dahl and Wille only use preconditioners for the Navier-Stokes equations, which are based on the Stokes coefficient matrix. No results are given for irregular shaped domains. This approach is closely related to our iterative methods.
In [6], [3] a finite volume discretization on a structured grid is used for the 2-dimensional Navier-Stokes equations. The convective terms are discretized by upwind techniques. In [6] the RCM and MUM orderings are applied. To prevent zero pivots Clift and Forsyth

consider p-last, u v p* last (only the pressure unknowns related to a Dirichlet boundary are numbered last), and a pre-elimination of the pressure unknowns. They conclude from their experiments that pre-elimination combined with RCM is the most effective method.

Our solution method
Here we specify our choices to solve the discretized coupled Navier-Stokes equations.

Solution method

1. After mesh generation the grid points are reordered by the Cuthill-McKee or Sloan renumbering method.
2. To prevent a zero pivot during the incomplete LU decomposition, the unknowns per grid point are reordered by the p-last, or the p-last per level reordering methods.
3. If the normal component of the velocity is prescribed on the complete boundary, the pressure is determined up to a constant, so the coeffient matrix is singular. Using a Gaussian elimination method this is repaired by specifying the pressure in one grid point. We apply the iterative method directly to the singular system, which leads to a better rate of convergence. No problems occur as long as the right-hand side is in the column space of the coeffient matrix.
4. In terms of the p-last ordering the system of equations we solve has the following form:
$$M \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}. \tag{16}$$
which is a discretized version of (4) and (5).
5. An incomplete LU decomposition of the matrix M is constructed and used as a preconditioner.
6. A preconditioned Krylov subspace method (GMRES, Bi-CGSTAB, etc.) is used to approximate the solution.

In the next paragraphs we motivate our choices in more detail.

Two representations of the linear system
From Section 2 it is clear that when we use the p-last ordering we have to solve the following coupled system of equations:
$$\tilde{M} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{17}$$
$A \in I\!R^{n_v \times n_v}$, and $B \in I\!R^{n_p \times n_v}$, where n_v is the number of velocity unknowns and n_p is the number of pressure unknowns. Another representation of this system is given in (16). An

advantage of (17) is that for the Stokes problem $\tilde{M}$ is a symmetric matrix. For Navier-Stokes A is non-symmetric, so $\tilde{M}$ is also non-symmetric. A disadvantage is however that $\tilde{M}$ is never positive definite and as a consequence a symmetric incomplete LL^T decomposition breaks down. Let us consider the positive definiteness properties of $\tilde{M}$ and M. Pre- and post multiplication of the matrix by an arbitrary vector gives:

$$\begin{pmatrix} x^T & y^T \end{pmatrix} \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = x^T A x + 2x^T B y, \tag{18}$$

and

$$\begin{pmatrix} x^T & y^T \end{pmatrix} \begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = x^T A x. \tag{19}$$

For a given vector x the vector y can always be chosen in such a way that the right-hand side of (18) is negative. On the other hand, for every choice of x and y the sign of the right-hand side of (19) only depends on the properties of A. So if A is positive definite, M is positive semi-definite.

Eigenvalue properties of M and $\tilde{M}$

It is well known that the rate of convergence of Krylov subspace iterative methods depends on the spectrum of the coefficient matrix (see [19] and [24]). For this reason we have computed approximations of the eigenvalues of M and $\tilde{M}$. For the Stokes problem the spectrum of M contains complex eigenvalues but the real part of all eigenvalues are positive. The eigenvalues of $\tilde{M}$ are all real but the spectrum contains positive and negative eigenvalues. In general, a preconditioner will be constructed such that the eigenvalues of the preconditioned matrix are clustered. If all the eigenvalues of the original matrix have a positive real part no problems are expected. However, in the case that the original matrix has positive and negative eigenvalues difficulties may occur, because it is possible that a negative eigenvalue becomes an eigenvalue of the preconditioned matrix very close to zero, which leads to bad convergence. So we expect that the form (16) is preferred if we want to solve the system of equations by a preconditioned iterative method.

Scaling of the continuity equation

One may expect that the properties of M and thus the convergence of the iterative methods depend on the scaling between the momentum and the continuity equation. Suppose that the continuity equation is multiplied by a constant τ. The eigenvalues of M satisfy the following equation:

$$\begin{pmatrix} A & B^T \\ -\tau B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix}. \tag{20}$$

For $\lambda \neq 0$, equation (20) can be rewritten as

$$(A - \frac{\tau}{\lambda} B^T B)x = \lambda x. \tag{21}$$

First the spectrum of M is considered for τ small. Take e_i the ith unit vector in $\mathbb{R}^{n_p}$ and $v_i = -A^{-1}B^T e_i$ for $i = 1, ..., n_p$. Substitution of this choice into (20) leads to

$$M \begin{pmatrix} v_i \\ e_i \end{pmatrix} = \begin{pmatrix} 0 \\ -\tau B A^{-1} B^T e_i \end{pmatrix}. \tag{22}$$

This equation shows that for τ small $\| M \begin{pmatrix} v_i \\ e_i \end{pmatrix} \|_2$ is small, which implies that the vectors $\begin{pmatrix} v_i \\ e_i \end{pmatrix}$ are close to the eigenvectors corresponding to a small eigenvalue. So there are n_p eigenvalues of M clustered around zero. Furthermore it follows from (21) that the other eigenvalues of M are near the eigenvalues of A.
For τ large the eigenvalues of $B^T B$ play an important role. Since $B \in \mathbb{R}^{n_p \times n_v}$ the matrix $B^T B$ is an element of $\mathbb{R}^{n_v \times n_v}$ and rank $B^T B = n_p$. This implies that $B^T B$ is a singular matrix ($n_p \ll n_v$). So also for large values of τ eigenvalues close to zero are expected.
Without preconditioning, GMRES [19] converges for τ small, but no convergence occurs for τ large ($\tau > 50$). For $0 < \tau \ll 1$ the residual of the continuity equation has a small influence on the coupled residual. This explains that the approximated velocity field may be not divergence-free and thus large errors occur. If GMRES is combined with one of the preconditioners presented below, then for $\tau < 50$ the resulting method appears to be nearly independent of the choice of τ. However, for $\tau > 50$, also the preconditioned methods are non convergent. For that reason we have limited ourselves in the numerical experiments to the natural scaling ($\tau = 1$).

Preconditioners
We now consider two incomplete LU decompositions: the classic ILU decomposition and ILUD [17]. Both preconditioners are combined with Krylov subspace methods to solve the system (16). For the explanation of the incomplete LU decompositions we define the set P, which consists of pairs (i, j) such that the i^{th} component of the vector of unknowns is connected to the j^{th} component. It is not necessary that this is a non zero connection. In the classic ILU decomposition the matrix M is approximated by a product of a lower and upper triangular matrix where the elements on positions not in P are set equal to zero. In formulas this means: consider L a lower triangular matrix, D a diagonal matrix, U an upper triangular matrix. The decomposition is such that $M \approx \hat{L}\hat{U}$ with $\hat{L} = LD^{-1/2}$ and $\hat{U} = D^{-1/2}U$. The following rules are used:

ILU
1. $diag(L) = diag(U) = D$,
2. $l_{i,j} = u_{j,i} = 0$ for $(i, j) \notin P$,
3. $(\hat{L}\hat{U})_{i,j} = m_{i,j}$ for $(i, j) \in P$.

In the ILUD preconditioner the off diagonal elements of L and U are taken equal to the

corresponding elements of M, only the matrix D has to be determined. In formulas:

ILUD
1. $diag(L) = diag(U) = D$,
2. $l_{i,j} = m_{i,j}$, and $u_{j,i} = m_{j,i}$ for $i > j$,
3. $(\hat{L}\hat{U})_{i,i} = m_{i,i}$.

Note that for both preconditioners $d_i > 0$ is necessary in order to form $\hat{L}$ and $\hat{U}$. In practice ILU is robust, but expensive. The ILUD preconditioner is cheap to build and easier to analyse. Furthermore the ILUD preconditioned Krylov method can be optimized by using the Eisenstat implementation [11].

Existence results for the ILUD decomposition
In the following propositions existence results for the ILUD decomposition are given.

Proposition 1
If we use the p-last ordering and assume that the ILUD decomposition of A exists and every column of B^T contains a non zero element then the ILUD decomposition exists.

Proposition 2
For an arbitrary ordering we suppose that the ILUD decomposition exists for all $j < i$, where the i^{th} row is related to the continuity equation. If the i^{th} (pressure) unknown is preceded by at least one velocity unknown with a non zero connection to this pressure unknown, then the ILUD decomposition exists.

Both propositions are valid for Stokes and Navier-Stokes problems and are proved in Appendix A. With respect to Proposition 1 we note that the assumption on B is satisfied in many practical applications, but the assumption on A is not always satisfied. The assumptions given in Proposition 2 have motivated us to construct the p-last per level ordering. Initially we expect problems due to the zero diagonal elements. To our surprise we have had no problems with the zero diagonal elements if a correct ordering (p-last, p-last per level) is used.

Some remarks on orderings
Without preconditioning GMRES is independent of the ordering of the unknowns. If an ILU preconditioner is used then the convergence of preconditioned GMRES strongly depends on the ordering used. For symmetric problems the influence of the ordering is investigated in [10]. Duff and Meurant conclude that preconditioned CG converges fast for 'local' orderings, which means that neighbouring nodes in the underlying mesh have numbers that are not too far apart. This is closely related to minimizing the bandwidth or profile of the matrix, and motivates us to order the grid points using a Cuthill-McKee

[14] or Sloan [21] reordering. Thereafter the unknowns in every grid point are reordered by the p-last or p-last per level ordering to prevent zero pivots in the ILU decomposition.

From our numerical experiments we conclude that if various orderings are used, then the ordering with the largest elements in D leads to the fastest convergence. Furthermore we observe that ordering a pressure unknown before velocity unknowns with a non zero connection leads to larger elements of D. This explains that preconditioned GMRES using p-last per level converges faster than using the p-last ordering.

The effect of ordering a pressure unknown before or after a velocity unknown depends on the discretization method used. For the Crouzeix Raviart element we observe that using the Cuthill-McKee renumbering, the p-last per level and p-last ordering lead to exactly the same convergence results. We explain this phenomenon for the p-last per element ordering. This ordering is defined as follows: suppose that the finite elements are already ordered. Then we start with the first element and number its velocity unknowns and then its pressure unknowns. Thereafter the non numbered velocity unknowns of the second element are numbered and then its pressure unknowns etc.

Consider the construction of the ILUD decomposition. Note that for the Crouzeix Raviart element the pressure unknowns in an element are only connected to velocity unknowns in the same element. In the velocity part corresponding to the first element the decomposition is not influenced by the pressure unknowns due to the ordering used. So this part is equal to the one obtained by using the p-last ordering. The pressure part in the first element is also equal to the corresponding part of the decomposition obtained from the p-last ordering, because all velocity unknowns with a connection to the pressure unknowns are already numbered. Furthermore these pressure unknowns have no connections with the following velocity unknowns, so they do not influence the decomposition of the velocity part in the remaining elements. This implies that the p-last per element ordering leads to an ILUD decomposition which is a permuted version of the p-last numbered ILUD decomposition. So the approximations computed with preconditioned GMRES are the same for both orderings.

In the same way it can be shown that if the grid points are ordered by a Cuthill-McKee ordering then the p-last per level ordering leads to the same results as the p-last ordering. The same holds for the ILU decomposition. If the Sloan ordering is used the p-last per level ordering leads to better results than the p-last ordering. For other type of finite elements (for instance Taylor Hood elements) we expect that p-last per level differs from p-last.

6 Numerical experiments

In this section we give some numerical experiments with the preconditioners given in Section 5. We start with the solution of the Stokes equations, where we compare the iterative method with a direct and a penalty method. Thereafter we solve the Navier-Stokes equations on a three-dimensional Backward Facing Step problem and investigate the influence of various choices for the preconditioner, the iterative method, the ordering of the unknowns and the Reynolds number. All our experiments have been carried out by triquadratic Crouzeix-Raviart hexahedrons, using 3×27 velocity unknowns and three pressure unknowns per element.

6.1 The Stokes equations

Consider the Stokes equations on a cube. In Table 1 the number of unknowns and the size of the matrices for the coupled system are given. Only non-zero elements are stored and the rows and columns corresponding to essential boundary conditions have been removed. The ratio between these two numbers gives the average number of non-zero elements per row. For the triquadratic hexahedron this ratio is relatively large (≈ 180). This has two

number of elements	number of unknowns	non-zero entries of the matrix	ratio
$3 \times 3 \times 3$	483	75 000	155
$6 \times 6 \times 6$	4 857	840 000	173
$12 \times 12 \times 12$	43 400	7 800 000	180

Table 1: The size of the problem with respect to the grid-size

important implications: the CPU time for a matrix vector multiplication is large compared to the CPU time for a vector update, and a large part of the fill-in is used in the classic ILU decomposition, so we expect a fast convergence of the preconditioned iterative method.

In Table 2 the results are summarised for three different methods: a penalty method, a direct method and an iterative method (GMRES with ILUD). The last two methods are applied to the coupled problem (16). We observe no break down of the direct and iterative method if the p-last per level ordering technique is used (see Section 4). Comparing the different solution methods, we see that the iterative method leads to a large decrease in CPU time and memory requirements. The CPU time is measured in seconds on an HP 735 workstation.

The work and memory requirements can be estimated a priori. For the penalty and direct

method	$3 \times 3 \times 3$		$6 \times 6 \times 6$	
	non-zero entries	CPU time	non-zero entries	CPU time
direct	160,000	1.3	5,000,000	237
penalty	100,000	0.47	3,400,000	132
iterative	75,000	0.08	840,000	2.74

Table 2: The CPU time and the memory requirements for the various methods

method the memory required is proportional to n_1^5, where n_1 denotes the number of grid points in the x_1-direction. The amount of work for these methods depends on n_1^7. For the iterative method the values are n_1^3 and n_1^4 respectively. In Table 3 these expressions are compared with the measurements. For the iterative method we were able to obtain also

method	memory	work
direct	31.7 (32)	182 (128)
penalty	34 (32)	247 (128)
iterative	11 (8)	34 (16)

Table 3: The ratio for the memory and work requirements for $n_1 = 6$ and $n_1 = 3$. Between brackets the theoretically expected values

the ratios for $n_1 = 12$ and $n_1 = 6$. They are given by 9.4 (memory) and 24 (work). We see a reasonable correspondence between theory and experiment. Furthermore the differences between the direct and penalty method and the iterative method increases enormously for increasing grid size.

In this example both preconditioners are used. Our experiments indicate that ILU is more robust than ILUD and it leads to less iterations of the preconditioned GMRES method. However the construction of the ILU decomposition takes a lot more work and doubles the memory required. For this reason if the ILUD decomposition does not breakdown, ILUD is preferred, because the extra memory is negligible and the total CPU time is, in general, less than that for ILU.

One of the reasons to use M instead of $\tilde{M}$ is that we expect that the eigenvalues of M have positive real parts, whereas the eigenvalues of $\tilde{M}$ are real but there are positive and negative eigenvalues. To illustrate this the eigenvalues of M and $\tilde{M}$ are approximated (see Figure 3 and 4). These figures show that the properties of the eigenvalues of M and $\tilde{M}$ are in agreement with the expected ones. Note that the spectrum of M contains complex

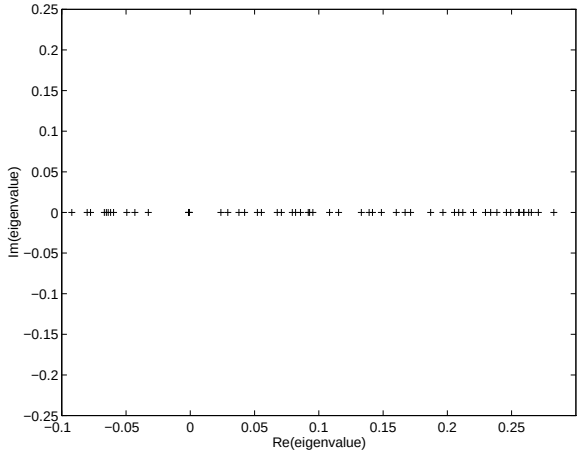

Figure 3: The spectrum of $\tilde{M}$

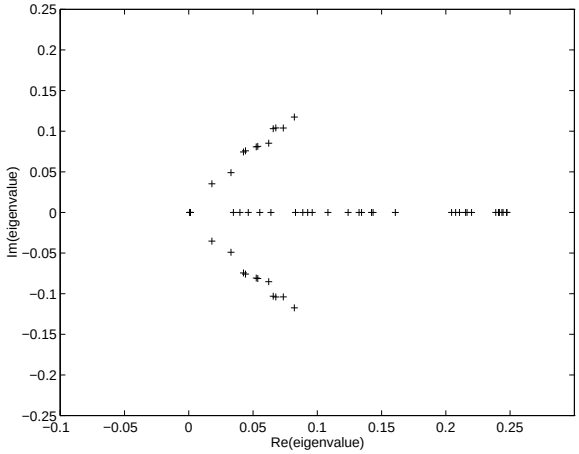

Figure 4: The spectrum of M

eigenvalues.

6.2 The Navier-Stokes equations

In this section we solve the Navier-Stokes equations on a three-dimensional Backward Facing Step problem. The geometry is given in Figure 5. At the left boundary surface we

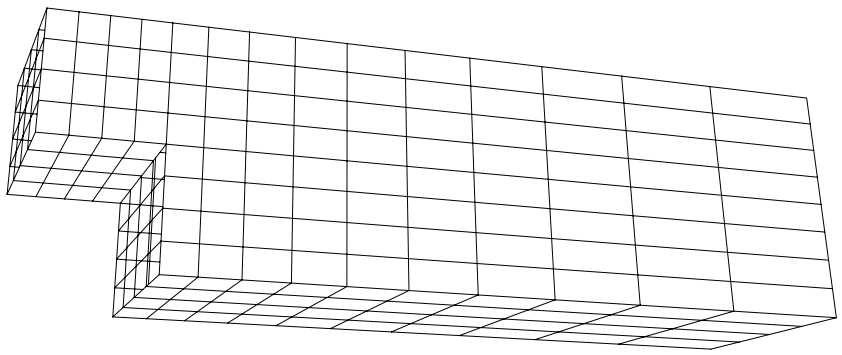

Figure 5: The geometry of the three-dimensional Backward Facing Step problem

use a Dirichlet inflow boundary condition and at the right boundary surface we use an outflow boundary condition: $\sigma_{nn} = 0$ and $\sigma_{\mathbf{nt}} = \mathbf{0}$. At all other boundaries we use a no slip condition. In this case we have to solve a non-linear problem. The strategy to

do this is given in Section 4. Initially we solve the corresponding Stokes equations, thereafter some Picard iterations are done and finally some Newton Raphson iterations are used.

Comparison of GMRES and Bi-CGSTAB
The ILU/ILUD decompositions are based on the current coefficient matrix, so a new decomposition is made in every outer iteration. As Krylov subspace methods we use GMRES [19] and Bi-CGSTAB [23]. When we restrict ourselves to Stokes and Picard outer iterations then for both methods 9 outer iterations are needed. The total number of inner iterations is 153 for GMRES and 122 for Bi-CGSTAB. The total CPU time (including building of the matrices and decompositions) is 351 s for GMRES and 385 s for Bi-CGSTAB. Note that Bi-CGSTAB uses less iterations, but one iteration of Bi-CGSTAB is approximately two times as expensive as an iteration of GMRES. This explains the bigger CPU time for Bi-CGSTAB. The optimal CPU time for GMRES is not unexpected since it is known that if the number of iterations is small and a matrix vector product is expensive (which means a large number of non zero elements per row) then GMRES is the best method (see [26]). Chin and Forsyth conclude in [4] that Bi-CGSTAB is faster than GMRES, which contrasts with our conclusion. An explanation could be that the matrices used in [4] are much sparser than the ones used in this paper.

Comparison of Picard and Newton Raphson
We have experimented with different strategies to solve the non-linear equations. Our experiments show that the number of inner iterations in a Newton Raphson step is slightly more than in a Picard step. However, in general less outer iterations are needed if Newton Raphson is used. For this reason we use the Stokes equations in the first iteration, Picard in the second iteration and Newton Raphson in the following iterations. In general 5 or 6 outer iterations are sufficient to reduce the initial error by a factor of 10^{-4}. We stop the inner iteration if $\|r_k\|_2/\|r_0\|_2 < eps$. If Picard iterations are used $eps = 10^{-1}$ is sufficient. If Newton Raphson steps are used it may be better to use $eps = 10^{-2}$, because then the outer iterations converge quadratically, whereas if $eps = 10^{-1}$ is used Newton Raphson has a linear convergence behaviour. Our results concerning the use of Picard or Newton Raphson are comparable to those given in [6].

Comparison of p-last and p-last per level
In Table 4 the number of GMRES iterations are given for the Sloan and Cuthill-McKee numbering. As expected the p-last and p-last per level leads to the same number of iterations for the Cuthill-McKee numbering. The results given in Table 4 show that Sloan numbering leads to a faster convergence of GMRES than Cuthill-McKee. Furthermore in these experiments the Sloan with p-last per level reordering leads to the best results.

Dependence on the Reynolds number
In general an increase of the Reynolds number leads to a lower rate of convergence for the

outer iteration	Sloan p-last per level	Sloan p-last	Cuthill-McKee p-last per level
1	10	13	20
2	25	32	40
3	15	27	34
4	15	20	38
5	21	23	38
6	17	30	39
7	19	19	38
8	18	17	39
9	13	23	37
total	153	204	323

Table 4: The number of preconditioned GMRES iterations for various orderings

non-linear iteration method and the iterative method for the coupled system. In Table 5 results are given for various choices of the Reynolds number. The Reynolds number is given by Re= $\frac{\rho h U}{\mu}$, where ρ is the density, h the stepsize and U the maximal value of the velocity at the inflow boundary.

Reynolds	outer iterations	average number of inner iterations	CPU-time
25	7	14	253
50	9	17	345
100	14	32	631

Table 5: The number of iterations for various values of the Reynolds number

Comparison of ILU and ILUD

Finally we observe that in this problem the ILUD decomposition breaks down, so we only use the ILU preconditioner. It appears that break down of the ILUD decomposition always happens in the velocity part, which is in agreement with the theory given in Section 5. The construction of the ILU decomposition is expensive. The CPU-time is comparable to the CPU time to build the coefficient matrix. With respect to the ordering techniques we observe no break down of the ILU decomposition using the p-last, or p-last per level ordering.

Results for a BFS problem on a fine grid

We end this section with some results for the BFS problem on an $8 \times 16 \times 28$ grid. The number of unknowns is equal to 8×10^4 and the number of non zero entries of the matrix is equal to 1.4×10^7. The CPU time to build the matrix is 2 min., whereas the CPU time to build the ILU preconditioner is 3 min. Using ILU, GMRES, p-last per level and $eps = 10^{-2}$, 5 outer iterations are needed. The total CPU time is 80 min. and the total number of inner iterations is 350.

7 Conclusions

In this paper we have considered the iterative solution of the incompressible Navier-Stokes equations. To that end the momentum equations coupled with the continuity equation has been solved. It has been shown that is necessary to use ordering techniques to prevent break down of the LU decomposition. Our numerical experiments demonstrate that direct methods can be used with the p-last per level ordering. The CPU time and memory requirements for the direct method are comparable to that of the penalty approach.

Exactly the same ordering techniques as for the direct method may be applied for the iterative approach. Combination of these orderings with preconditioned Krylov subspace methods lead to a good convergence and prevents break down of the ILU decomposition. This has been theoretically investigated and has been confirmed by our experiments. Furthermore the experiments show that the iterative solution methods presented in this paper are far superior above direct methods at least for three-dimensional problems. Not only is the required memory considerably less, also the CPU time has been largely reduced.

ACKNOWLEDGEMENT

The authors would like to thank C.G.M. Kassels for software support.

References

[1] R.E. Bank, B.D. Welfert, and H. Yserentant. A class of iterative methods for solving saddle point problems. *Numer. Math.*, 56:645–666, 1990.

[2] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems*. SIAM, Philadelphia, 1994.

[3] P. Chin, E.F. D'Azevedo, P.A. Forsyth, and W.L. Tang. Preconditioned conjugate gradient methods for the incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids*, 15:273–295, 1992.

[4] P. Chin and P.A. Forsyth. A comparison of GMRES and CGSTAB acceleration for incompressible Navier Stokes. *J. Comp. Appl. Math.*, 46:415–426, 1993.

[5] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[6] S.S. Clift and P.A. Forsyth. Linear and nonlinear iterative methods for incompressible Navier-Stokes equations. *Int. J. Num. Meth. in Fluids*, 18:229–256, 1994.

[7] C. Cuvelier, A. Segal, and A.A. van Steenhoven. *Finite element methods and Navier-Stokes equations*. Reidel Publishing Company, Dordrecht, Holland, 1986.

[8] O. Dahl and S.O. Wille. An ILU preconditioner with coupled node fill-in for iterative solution of the mixed finite element formulation of the 2D and 3D Navier-Stokes equations. *Int. J. Num. Meth. in Fluids*, 15:525–544, 1992.

[9] J. Donea, S. Giuliani, H. Laval, and L. Quartapelle. Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Comp. Meth. in Appl. Mech. and Engng.*, 30:53–73, 1982.

[10] I.S. Duff and G.A. Meurant. The effect of ordering on preconditioned conjugate gradient. *BIT*, 29:635–657, 1989.

[11] S.C. Eisenstat. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM J. Sci. Stat. Comput.*, 2:1–4, 1981.

[12] Howard C. Elman and Gene H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM J. Matrix Numer. Anal.*, 31:1645–1661, 1994.

[13] M. Fortin and R. Glowinski. *Augmented Lagrangian methods: Application to the numerical solution of boundary value problems*. Studies in Mathematics and its Applications. North Holland, Amsterdam, 1983.

[14] A. George and J.W.H. Liu. *Computer solution of large sparse positive definite systems.* Prentice-Hall, Engelwood Cliffs, 1981.

[15] D.F. Griffiths. An approximately divergence-free 9-node velocity element (with variations) for incompressible flows. *Int. J. for Num. Methods in Fluids*, 1:323–346, 1982.

[16] F. Hecht. Construction d'une base de fonctions p_1 non conforme à divergence nulle dans r^3. *RAIRO Anal. Num.*, 15:119–150, 1981.

[17] J.A. Meijerink and H.A. Van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.

[18] T. Rusten and R. Winther. A preconditioned iterative method for saddle-point problems. *SIAM J. Matrix Anal. Appl.*, 13:887–904, 1992.

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

[20] D.J. Silvester and A.J. Wathen. Fast iterative solution of stabilised Stokes systems Part 2: using general block preconditioners. *SIAM J. Num. Anal.*, 31:1352–1367, 1994.

[21] S.W. Sloan. An algorithm for profile and wavefront reduction of sparse matrices. *Int. J. Num. Meth. Engng.*, 23:239–251, 1986.

[22] F. Thomasset. *Implementation of finite element methods for Navier-Stokes equations.* Springer Verlag, Berlin, 1981.

[23] H.A. Van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13:631–644, 1992.

[24] H.A. van der Vorst and C. Vuik. The superlinear convergence behaviour of GMRES. *J. Comput. Appl. Math.*, 48:327–341, 1993.

[25] J.J.I.M. Van Kan. A second-order accurate pressure correction method for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.*, 7:870–891, 1986.

[26] C. Vuik. Further experiences with GMRESR. *Supercomputer*, 55:13–27, 1993.

[27] A.J. Wathen and D.J. Silvester. Fast iterative solution of stabilised Stokes systems Part 1: using simple diagonal preconditioners. *SIAM J. Num. Anal.*, 30:630–649, 1993.

[28] P. Wesseling. *An introduction to multigrid methods.* John Wiley & Sons, Chichester, 1992.

A Existence proofs

In this appendix Proposition 1 and 2 are proved. Before the proofs are given we consider the third rule from ILUD:
$(\hat{L}\hat{U})_{i,i} = (LD^{-1/2}D^{-1/2}U)_{i,i} = d_i + \sum_{j=1}^{i-1} \frac{l_{i,j} * u_{j,i}}{d_j} = m_{i,i}$.
Combination with rule 2 leads to:

$$d_i = m_{i,i} - \sum_{j=1}^{i-1} \frac{m_{i,j} * m_{j,i}}{d_j}. \tag{23}$$

Proposition 1
If we use the p-last ordering and assume that the ILUD decomposition of A exists and every column of B^T contains a non zero element then the ILUD decomposition exists and $d_i > 0$ for $i \in [1, n]$.

Proof: From the assumptions it follows that the ILUD decomposition of A exists and thus $d_j > 0$ for $j = 1, ..., n_v$. For $i \in (n_v, n]$ we have $m_{i,j} = -m_{j,i}$ and $m_{i,i} = 0$. This together with (23) implies that $d_i = \sum_{j=1}^{i-1} \frac{m_{i,j}^2}{d_j}$. Since the norm of a column of B^T is non-zero we have $\sum_{j=1}^{i-1} m_{i,j}^2 > 0$. Combined with $d_k > 0$ for $k < i$ it follows that

$$d_i \geq (min_{1 \leq k \leq i-1} \frac{1}{d_k}) \sum_{j=1}^{i-1} m_{i,j}^2 > 0.$$

□

Remark:
If the ILUD preconditioner is applied to $\tilde{M}$ then $d_j > 0$ for $j \leq n_v$, but $d_{n_v+1} < 0$ so it is impossible to form $\hat{L}$ and $\hat{U}$.

Suppose another ordering is used, for instance the p-last per level ordering. Then there exists a permutation matrix P such that M is given by

$$M = P^T \begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix} P.$$

The equations $m_{i,j} = -m_{j,i}$ and $m_{i,i} = 0$ again hold for a row which corresponds with the continuity equation.

Proposition 2
For an arbitrary ordering we suppose that the ILUD decomposition exists for all $j < i$ (so $d_j > 0$) , where the i^{th} row is related to the continuity equation. If the i^{th} (pressure)

unknown is preceded by a velocity unknown with a non zero connection (so there is one $k < i$ such that $m_{i,k} \neq 0$) then the ILUD decomposition exists and $d_i > 0$.

Proof: It follows again from (23) that

$$d_i = \sum_{j=1}^{i-1} \frac{m_{i,j}^2}{d_j}.$$

Since $d_j > 0$ for $j < i$ and $m_{i,k}^2 > 0$ for at least one $k < i$ we obtain $d_i > 0$ □