V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

PRECONDITIONED KRYLOV METHODS FOR THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

C. Vuik*, M. ur Rehman* and A. Segal*

*Delft University of Technology
Mekelweg 4, 2628 CD Delft, Netherlands
e-mail: {c.vuik,m.urrehman,a.segal}@tudelft.nl

Key words: Navier-Stokes equations; finite element method; block preconditioners; SIMPLE-type schemes; iterative methods; incompressible fluids

Abstract. *In this paper, we solve the discretized steady incompressible Navier-Stokes equations with preconditioned Krylov methods. Discretization is carried out with a finite element method. The linearized Navier-Stokes equations give rise to a saddle point problem due to the absence of pressure in the continuity equation. We discuss two types of preconditioners: algebraic, based on ILU factorization, and block preconditioners exploiting the block structure of the coefficient matrix. We show that both preconditioners give good convergence.*

1 Introduction

We consider the basic equations of fluid dynamics and its discretization. We start with the steady state incompressible Navier-Stokes equations governing the flow of a Newtonian, incompressible viscous fluid. The equations are given by

$$-\nu\Delta\mathbf{u} + \mathbf{u}.\nabla\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in} \quad \Omega \tag{1}$$

$$\nabla.\mathbf{u} = 0 \quad \text{in} \quad \Omega. \tag{2}$$

$\Omega \subset \mathbf{R}^d (d = 2 \text{ } or \text{ } 3)$ is the flow domain with piecewise smooth boundary $\partial\Omega$, $\mathbf{u}$ is the fluid velocity, p is the pressure field, $\nu > 0$ is the kinematic viscosity coefficient (inversely proportional to the Reynolds number, Re), Δ is the Laplace operator, ∇ denotes the gradient and $\nabla.$ is the divergence operator.

Equation (1) represents conservation of momentum, while Equation (2) represents the incompressibility condition, or mass conservation. The boundary value problem that is considered is the system (1), (2), together with boundary conditions on $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ given by

$$\mathbf{u} = \mathbf{w} \ \text{ on } \ \partial\Omega_D, \quad \nu\frac{\partial\mathbf{u}}{\partial n} - \mathbf{n}p = \mathbf{s} \ \text{ on } \ \partial\Omega_N.$$

The presence of the convective term $\mathbf{u}.\nabla\mathbf{u}$ in the momentum equation makes the Navier-Stokes system nonlinear. It can be linearized with, for example, Picard or Newton's method. We use a finite element discretization with elements that satisfy the Ladyshenskaya-Babuska-Brezzi (LBB) condition. The linear equations in matrix form can be written as:

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \tag{3}$$

where $F \in \mathbb{R}^{n \times n}$ is the discretized convection-diffusion operator, $B \in \mathbb{R}^{m \times n}$ is the discretized divergence operator and $m \leq n$. The number of velocity unknowns is n and the number of pressure unknowns is m. The system matrix is sparse, symmetric indefinite in the case of the Stokes problem and unsymmetric indefinite in the Navier-Stokes problem.

To solve linear system (3), Krylov subspace methods with a suitable preconditioning technique are used. For most applications, convergence of Krylov subspace methods depends on the spectrum of the coefficient matrix. A preconditioner P transforms the linear system $Ax = b$ to a preconditioned system $P^{-1}Ax = P^{-1}b$, such that $P^{-1}y$ is cheap to compute and $P^{-1}A$ has a favorable spectrum for convergence. In general, preconditioning techniques based on algebraic and physics-based approaches are widely used. Algebraic type preconditioners are based on an ILU factorization or an approximate inverse of the coefficient matrix, where some pivoting or a priori reordering strategies are used to make the preconditioner stable and effective [4–6, 8, 10, 11, 17, 22, 25]. These preconditioners are known for their cheap and simple implementation and they require no extra knowledge of the underlying system.

Algebraic preconditioners applied to the complete system (3) may breakdown due to zero pivots. This problem can be solved by pivoting, which is in general very expensive. An alternative is to apply a suitable a-priori renumbering. In Section 2, we will shortly discuss the saddle point ILU (SILU) preconditioner, which is based on this strategy.

An alternative approach is the use of block preconditioners based on block factorization of the coefficient matrix. Separate subsystems for velocity and pressure are solved during each iteration. An important aspect of this approach is a good approximation of the Schur complement. Examples of such preconditioners can be found in [2, 3, 12, 13, 16, 24]. The final goal is to develop preconditioners that give convergence independent of the Reynolds number. Dependence of the number of nodal points should only be visible in the inner solves. In Section 3, we treat some block triangular preconditioners based on an approximation of the Schur complement by a pressure convection-diffusion operator (PCD).

2 The Saddle point ILU preconditioner (SILU)

The Saddle point ILU preconditioner is based on an incomplete factorization of the complete coefficient matrix with an a priori renumbering that makes the preconditioner applicable to saddle point problems [22]. Two kinds of reordering are introduced for this preconditioner:

1. Renumbering of grid points, that can be accomplished by any renumbering method that gives an optimal profile. Examples are the techniques described by Sloan [20] and Cuthill McKee [7].

2. Since we are dealing with saddle point problems, zero pivots may arise during ILU decomposition. An obvious way to avoid this problem is to renumber the unknowns in the sequence: first all the velocity unknowns and then the pressure unknowns. We call this *p-last* ordering. A more sophisticated reordering of unknowns is the so-called *p-last per level* reordering. The grid is subdivided into levels, where each level is a connected set of nodes. Thereafter unknowns are reordered per level, first the velocity unknowns and then the pressures.

After renumbering, standard ILU decomposition is applied to the reordered coefficient matrix A. The sparseness structure is defined as follows:

$$(LD^{-1}U)_{i,j} \neq 0 \text{ for } (i,j) \in \mathcal{S}, \tag{4}$$

where $\mathcal{S}$ consists of those entries of A that are filled by the standard finite element assembly procedure. So some elements in the pressure zero block are still part of $\mathcal{S}$ although their corresponding matrix coefficients are zero.

Both p-last and p-last per level avoid breakdown of the ILU preconditioner. The profile with the p-last ordering is relatively large compared to that of the p-last per level. Due to the local block reordering, zero pivots become non-zero, during factorization, and no a posteriori pivoting is required.

In the p-last per level reordering, one has to be careful at the start of this process. If, for example, the velocities in the first node, are prescribed, we start with a pressure unknown that gives rise to a zero pivot. Therefore, we always combine the first few levels, into a new level. If the number of free velocity unknowns in this new level, is less than the number of pressure unknowns, we also add the next level to level 1, and if necessary this process is repeated. In practice a combination of 2 or 3 levels is sufficient. Note that the starting level has always a small contribution to the global profile [22]. In our experiments, p-last per level in combination with a suitable renumbering of grid points is used. We have

observed that p-last per level improves the convergence of the preconditioned iterative method and avoids the breakdown of ILU. The convergence of the SILU preconditioner, however, depends on the size of the grid and depends mildly on the Reynolds number.
In some cases the SILU preconditioner has problems with convergence, especially in the case of stretched grids. In that case the preconditioner can be improved by assuming extra fill-in (SILUF). The standard fill-in is defined by considering all unknowns which are directly coupled by the mesh to a degree of freedom in a row. In our case the extra fill-in is defined by extending this non-zero pattern by all degrees of freedom that are also coupled to these neighbors, provided they are part of the profile of the matrix. Although SILUF requires extra memory, it helps to reduce computation time, even in the case that SILU works fine.

2.1 Breakdown of LU or ILU factorization

In this section we analyze under which condition the p-last per level strategy has no breakdown.

In the first level we need at least the same number of unprescribed velocity degrees of freedom as there are pressure degrees of freedom. Furthermore, the velocity unknowns should have a nonzero connection to the pressure unknowns. Experimentally, we have seen that this also holds for the ILU preconditioner. Consider the nonsymmetric case where we multiply the discretized continuity equation by a minus sign, hence $-Bu = -g$. We will prove that the preconditioner exists theoretically for an ILUD (see Definition 2.1) decomposition of this matrix,

$$\mathcal{A}_{ns} = \begin{bmatrix} F & B^T \\ -B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ -g \end{bmatrix}, \tag{5}$$

in which the off-diagonal elements of $\hat{L}$ and $\hat{U}$ are taken equal to the corresponding elements in $\mathcal{A}_{ns}$. Only the matrix $\mathcal{D}$ has to be determined.

Definition 2.1. ***ILUD:***

1. *$diag(\hat{L}) = diag(\hat{U}) = \mathcal{D}$,*
2. *$l_{i,j} = a_{i,j}$ for $i > j$ and $u_{i,j} = a_{i,j}$ for $j > i$,*
3. *$(\hat{L}\mathcal{D}^{-1}\hat{U}L)_{i,j} = a_{i,j}$.*

Proposition 2.1. *If we use the p-last ordering and assume that the ILUD decomposition of F exists with positive matrix D then the complete ILUD decomposition exists because every column of B^T contains a nonzero element. Note that if B^T has a zero column then $\mathcal{A}_{ns}$ is singular.*

Proof: We apply the ILUD decomposition to (5). We consider the computation of $\mathcal{D}$:

$$(\hat{L}\mathcal{D}^{-1}\hat{U})_{i,i} = d_i + \sum_{j=1}^{i-1} \frac{l_{i,j} \cdot u_{j,i}}{d_j} = a_{i,i}, \tag{6}$$

this leads to

$$d_i = a_{i,i} - \sum_{j=1}^{i-1} \frac{a_{i,j} \cdot a_{j,i}}{d_j}. \tag{7}$$

From the assumption it follows that the ILUD decomposition of F exists and thus $d_j > 0$ for $j = 1, ..., n$. For $i \in (n+1, n+m)$ we have $a_{i,j} = -a_{j,i}$ and $a_{i,i} = 0$. This together with (7) implies that $d_i = \sum_{j=1}^{i-1} \frac{a_{i,j}^2}{d_j}$. Since the norm of a column B^T is nonzero we have $\sum_{j=1}^{i-1} \frac{a_{i,j}^2}{d_j} > 0$. Combined with $d_k > 0$ for $k < i$ it follows that

$$d_i \geq (\min_{1 \leq k \leq i-1} \frac{1}{d_k}) \sum_{j=1}^{i-1} a_{i,j}^2 > 0. \tag{8}$$

Proposition 2.2. *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 nonzero connection to this pressure unknown (so there is one $k < i$ such that $a_{i,k} \neq 0$), then the ILUD decomposition exists and $d_i > 0$.*

Proof: It follows again from (7) that

$$d_i > \sum_{j=1}^{i-1} \frac{l_{i,j} \cdot u_{j,i}}{d_j}. \tag{9}$$

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

From the above propositions it is clear that the ILUD decomposition gives positive diagonal elements. If we use the original matrix (symmetric case) the diagonal elements corresponding to the pressure part appear to be negative. This is also the case for the ILU decomposition, but we have no theoretical proof.

One could ask oneself if it is possible to apply a standard profile renumbering scheme based on the matrix itself, like for example Approximate Minimum Degree (AMD) [1]. There are several reasons why SILU is better suited than AMD when applied to the incompressible Navier-Stokes equations.

The first is that SILU is cheaper to compute the reordered matrix. The reason is that the reordering is applied on grid nodes, and after that the levels are defined. AMD, on the other hand, is applied to the complete matrix. In the Navier-Stokes problem, the complete matrix is much larger than the adjacency matrix because the number of degrees of freedom per point is larger than one.

The second reason is, if in AMD, the first unknown with minimum degree corresponds to a pressure unknown, ILU will break-down, since the first diagonal element is equal to zero [9].

The main problem is that ordering schemes like AMD are built for SPD matrices or matrices having nonzero diagonals. In our case, the breakdown criterion is simple. If at least one velocity unknown is connected to a pressure unknown in each level, it will not breakdown. So applying AMD type methods for saddle point problems is somewhat dangerous.

3 Block preconditioners

Block preconditioners for the incompressible Navier-Stokes problem are based on block LDU decomposition of the coefficient matrix. The factorization can be written as:

$$\mathcal{A} = LDU = \begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ BF^{-1} & I \end{bmatrix} \begin{bmatrix} F & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & F^{-1}B^T \\ 0 & I \end{bmatrix}, \tag{10}$$

where $S = -BF^{-1}B^T$ is the Schur complement matrix. Most preconditioners are based on a combination of these blocks and a suitable approximation of the Schur complement matrix. A popular class of preconditioners is the set of block triangular preconditioners based on the DU decomposition [14], [2]:

$$P_t = \begin{bmatrix} F & B^T \\ 0 & S \end{bmatrix}. \tag{11}$$

In general, the use of F^{-1} and S^{-1} is not practical since the matrices are very expensive to compute and to store. F^{-1} is formally approximated by a matrix $\hat{F}^{-1}$. Usually, such an approximation consists of a small number of iterations with an iterative method. S is first approximated and then the system $Sz = r$ is solved approximately.

In this paper we discuss the pressure convection-diffusion preconditioner (PCD) that is of block triangular type [16], [19], [14]. The preconditioner is based on the assumption that a commutator of the convection diffusion operator on the velocity space ($\mathcal{L}$), multiplied by the gradient operator, on the velocity space, and the gradient operator, acting on the convection diffusion operator in the pressure space ($\mathcal{L}_p$), is small.

$$\varepsilon = \mathcal{L}\nabla - \nabla\mathcal{L}_p. \tag{12}$$

In matrix form, basically we are minimizing the norm:

$$\left|\left|FB^T - B^T F_p\right|\right|, \tag{13}$$

where F_p is the convection-diffusion operator on pressure space. This gives rise to a Schur complement approximation of the form:

$$BF^{-1}B^T \approx BQ_u^{-1}B^T F_p^{-1} Q_p, \tag{14}$$

where Q_u denotes the velocity mass matrix and Q_p the pressure mass matrix. The expensive part $BQ_u^{-1}B^T$ (dense) in (14) is replaced by the pressure Laplacian matrix, A_p, which is spectrally equivalent to $BQ_u^{-1}B^T$.

$$S = -BF^{-1}B^T \approx -A_p F_p^{-1} Q_p. \tag{15}$$

The preconditioner has some nice convergence properties. It gives mesh independent convergence if the equations are discretized with Q2-Q1 finite elements. The main issue regarding the effectiveness of this preconditioner are the boundary conditions of operator F_p. It may happen that the commutator in (13) is small everywhere except on the boundaries due to the choice of boundary conditions. The classical ad-hoc choice for boundary conditions for F_p is, Dirichlet boundary conditions at inflow and homogeneous Neumann boundary conditions at outflow. The same choice is made for the operator A_p. This choice of boundary conditions is based on numerical experiments.

Recently it has been discovered, that better boundary conditions for F_p are possible, resulting in a much better performance of PCD. Instead of (13), one considers the commutator based on gradient and divergence:

$$||BF - F_p B|| \approx 0 \tag{16}$$

The new formulation based on (16) becomes:

$$S = -BF^{-1}B^T \approx -Q_p F_p^{-1} A_p. \tag{17}$$

It has also been observed that in (16), boundary conditions for F_p are more important than for A_p, therefore A_p is replaced by $B\hat{Q}_u^{-1}B^T$, where $\hat{Q}_u$ is the diagonal of the velocity mass matrix. This choice avoids the need of boundary conditions for A_p. The new approximation uses a Robin boundary condition at inflow given by:

$$-\nu \frac{\partial p}{\partial n} + \mathbf{w}.\mathbf{n}p = 0, \tag{18}$$

with $\mathbf{w}$ the inflow velocity.

If ν is small, then (18) effectively reduces to a Dirichlet boundary condition. In the following discussion we call this version *new PCD*.

We will compare PCD and *new PCD* with the MSIMPLER (Modified SIMPLER) preconditioner that does not require boundary conditions at all [23], [21].

MSIMPLER is based on a variant of the classical SIMPLE (Semi-Implicit Pressure Linked Equations) method of Patankar used as preconditioner. Instead of replacing F by $diag(F) = D$ (as in SIMPLE and SIMPLER), F is replaced by $diag(Q_u) = \hat{Q_u}$. The preconditioner is given by:

MSIMPLER preconditioner:

1. Solve $\hat{S}p^* = r_p - B\hat{Q_u}^{-1}r_u$.
2. Solve $Fu^* = r_u - B^Tp^*$.
3. Solve $\hat{S}\delta p = r_p - Bu^*$.
4. update $u = u^* - \hat{Q_u}^{-1}B^T\delta p$.
5. update $p = p^* + \delta p$.

where $\hat{S} = B\hat{Q_u}^{-1}B^T$. The preconditioner consists of one velocity solve and two Poisson solves and does not require any extra operators to construct. It shows a mild dependence of grid size and Reynolds number. In PCD, two new operators (A_p and F_p) are constructed. Per iteration PCD is cheaper as it requires one velocity solve and one Poisson solve along with a cheap computation of Q_p^{-1}.

In the next section some numerical experiments are given to get a clear idea how these preconditioners perform.

4 Numerical Experiments

The preconditioners discussed are applied on a driven cavity flow (enclosed flow) in a square cavity with enclosed boundary conditions and a lid moving from left to right given as:

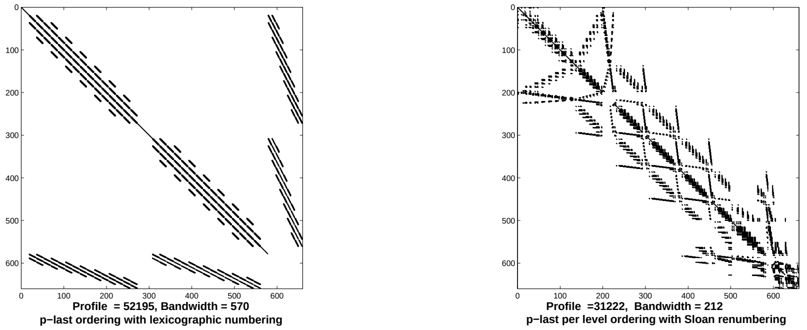
$$u_x = 1 - x^4 \text{ at } y = 1; \ -1 \leq x \leq 1,$$

which is known as the regularized cavity problem, and also to a flow over an obstruction (inflow-outflow). Parabolic inflow conditions are imposed on the rectangular region that has a rectangular obstacle inside the domain. Velocities are zero along the top and bottom of the channel and along the obstruction.

First we perform some numerical experiments to test the SILU preconditioner. Next we compare block preconditioners for both types of flows. The iteration is stopped if the linear systems satisfy $\frac{\|r^k\|_2}{\|b\|_2} \leq tol$, where r^k is the residual at the kth step of the Krylov subspace method, b is the right-hand side, and tol is the desired tolerance. Finally we compare SILU and SILUF with MSIMPLER. All comparisons are done in MATLAB, except the last one, where we use the Finite Element Package SEPRAN [18], written in FORTRAN.

In Figure 1, it can be clearly seen that our renumbering p-last per level gives an optimal profile in combination with a suitable node numbering strategy. The main advantage of this ordering is that no pivoting is necessary, since during factorization, the zeros on the main diagonal in the zero pressure block disappear. This reduces memory and CPU time due to decrease of fill-in.

Table 1, compares p-last and p-last per level ordering, showing that the last one reduces the time approximately by a factor 2.



Profile = 52195, Bandwidth = 570
p−last ordering with lexicographic numbering

Profile =31222, Bandwidth = 212
p−last per level ordering with Sloan renumbering

Figure 1: Effect of Sloan renumbering of grid points and p-last per level reordering of unknowns on the profile and bandwidth of the matrix

Table 1: Solution of the Stokes problem with the Q2-Q1 discretization in the square domain with an *accuracy* of 10^{-6} (Time = total time).

Grid size	p-last	p-last per level
	Number of iterations (Time in seconds)	
16×16	36 (0.11)	25(0.1)
32×32	90 (0.92)	59 (0.66)
64×64	255(11.98)	135(6.7)

Next we compare the PCD preconditioners with MSIMPLER. Table 2 gives a comparison for Re=100. "iter." stands for outer iterations and "Ts" for time in seconds.We solve the subsystems with a direct solver or one V-cycle of AMG for both the velocity and pressure subsystem. It can be seen that the new variant of PCD shows much better convergence than the classical one. This suggests that proper boundary conditions lead to much better results. The MSIMPLER preconditioner still performs better than both PCD variants. The convergence shows a mild dependence on the increase in number of grid elements.

For the PCD variants we see a difference in outer iterations if AMG is used instead of a direct solver, but in case of MSIMPLER it remains constant. Figure 2 shows that also for a higher Reynolds number MSIMPLER converges better than PCD.

Table 2: Flow over an obstruction, Q2-Q1 grid (GMRES, *accuracy* $= 10^{-6}$).

Grid	PCD		New PCD		MSIMPLER	
	Exact	AMG	Exact	AMG]	Exact	AMG
	iter.	iter. (Ts)	iter.	iter. (Ts)	iter.	iter. (Ts)
16×64	39	40(1.13)	20	25(0.87)	17	18(0.52)
32×128	46	53(6.69)	22	26(3.94)	16	16(2.0)
64×256	59	67 (30)	25	30(16)	22	22(9.3)

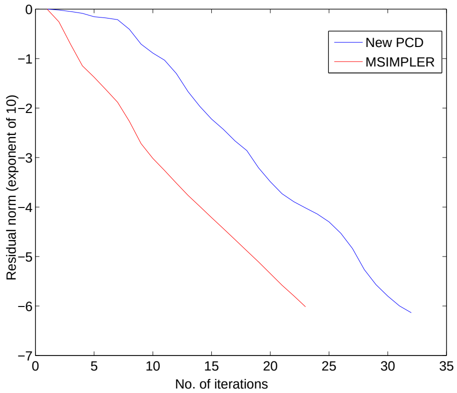


Figure 2: Flow over an obstruction (Re=300).

To compare the block preconditioners for high Reynolds numbers and increasing grid size, we did some experiments with the driven cavity problem (see Table 3). In [15] it is shown that the flow in the 2D driven cavity remains steady even up to Re=20000 if sufficient grid points are used. We test these preconditioners up to Re=8000. For a coarse mesh it can been seen that the number of iterations doubles with doubling the Reynolds number. As the mesh becomes finer, the increase in number of iterations becomes small. From Table 3 it is clear that usually MSIMPLER takes less iterations than PCD. However, even if the number of iterations is almost the same, like for RE=1000 on a 256 × 256 grid, still the

Table 3: Driven cavity flow problem, AMG used for the subsystems, Q2-Q1 grid (GMRES, *accuracy* $= 10^{-6}$).

Re	64×64		128×128		256×256	
	New PCD	MSIMPLER	New PCD	MSIMPLER	New PCD	MSIMPLER
	iter. (Ts)	iter. (Ts)	iter. (Ts)	iter. (Ts)	iter. (Ts)	iter. (Ts)
500	44(6.8)	34(4.6)	45(29)	42(19)	47(155)	56(122)
1000	54(9.7)	45(8.3)	56(45)	46(28)	58(202)	56(136)
2000	89(15)	69(10)	71(63)	57(39)	74(314)	61(188)
4000	171(30)	118(17)	113(100)	86(58)	94(450)	72(249)
8000	488(109)	304(52)	315(300)	184(127)	198(1035)	125(477)

Table 4: CPU time in seconds for solving driven cavity Stokes flow, P2-P1 grid, (*accuracy* $= 10^{-6}$).

Number of nodes	MSIMPLER	SILU	SILUF
17×17	0.0080	0.0000	0.0000
33×33	0.0800	0.0160	0.0080
65×65	0.9161	0.1320	0.0960
129×129	13.3408	1.6281	0.8561
257×257	250.7437	19.9972	7.7845
513×513	4200.4704	275.7932	86.1254

CPU time of MSIMPLER is much smaller. PCD takes more time in solving the Schur complement approximate system.

Finally in Table 4 we compare SILU, SILUF and MSIMPLER for a Stokes flow in a square cavity for a series of grids. The inner iterations in MSIMPLER are carried out with an ILU preconditioned CG method, using a relative accuracy of 10^{-2}. It is clear that SILUF is the best performer with respect to CPU time.

5 Conclusions

We have solved the incompressible Navier-Stokes equations with preconditioned Krylov methods. A reordering has been developed which reduces fill-in in LU decomposition and is effective if used in an ILU preconditioner. The p-last per level ordering is a good choice for reordering unknowns with any type of grid renumbering scheme that reduces the profile. A further reduction in time can be achieved by allowing extra fill-in.

We compared the MSIMPLER preconditioner with a new PCD version. It is observed that MSIMPLER is a competitive choice, even if proper boundary conditions are used in the PCD preconditioner.

REFERENCES

[1] P.R. Amestoy, T.A. Davis, and I.S. Duff. An Approximate Minimum Degree Ordering Algorithm. *SIAM. J. Matrix Anal. & Appl.*, 17(4):886–905, October 1996.

[2] M. Benzi, G.H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[3] M. Benzi and M.A. Olshanskii. An Augmented Lagrangian-Based Approach to the Oseen Problem. *SIAM J. Sci. Comput.*, 28(6):2095–2113, 2006.

[4] M. Benzi, D.B. Szyld, and A. van Duin. Orderings for Incomplete Factorization Preconditioning of Nonsymmetric Problems. *SIAM J. Sci. Comput.*, 20(5):1652–1670, 1999.

[5] M. Benzi and M. Tuma. A Sparse Approximate Inverse Preconditioner for Nonsymmetric Linear Systems. *SIAM Journal on Scientific Computing*, 19(3):968–994, 1998.

[6] M. Bollhöfer and Y. Saad. Multilevel Preconditioners Constructed From Inverse-Based ILUs. *SIAM J. Sci. Comput.*, 27(5):1627–1650, 2006.

[7] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pages 157–172. ACM Press, 1969.

[8] O. Dahl and S. Ø. 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. Numer. Meth. Fluids*, 15(5):525–544, 1992.

[9] A.C. de Niet and F.W. Wubs. Numerically stable LDLT-factorization of - type saddle point matrices. *IMA Journal of Numerical Analysis*, 29(1):208–234, 2009.

[10] I.S. Duff and G.A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT*, 29(4):635–657, 1989.

[11] L.C. Dutto. The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations. *Int. J. Numer. Meth. Engng.*, 36(3):457–497, 1993.

[12] H. C. Elman. Preconditioning for the Steady-State Navier–Stokes Equations with Low Viscosity. *SIAM Journal on Scientific Computing*, 20(4):1299–1316, 1999.

[13] H.C. Elman, V.E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block Preconditioners Based on Approximate Commutators. *SIAM J. Sci. Comput.*, 27(5):1651–1668, 2006.

[14] H.C. Elman, D. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers with applications in incompressible fluids dynamics.* Oxford University Press, Oxford, 2005.

[15] E. Erturk. Discussions on driven cavity flow. *International Journal for Numerical Methods in Fluids*, 60(3):275–294, 2009.

[16] D. Kay, D. Loghin, and A. Wathen. A Preconditioner for the Steady-State Navier-Stokes Equations. *SIAM J. Sci. Comput.*, 24(1):237–256, 2002.

[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. of Comp.*, 31(137):148–162, 1977.

[18] A. Segal. *SEPRAN manuals.* Den Haag, The Netherlands, 2010.

[19] D. Silvester, H.C. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *J. Comput. Appl. Math.*, 128(1-2):261–279, 2001.

[20] S.W. Sloan. An algorithm for profile and wavefront reduction of sparse matrices. *Int. J. Numer. Meth. Engng.*, 23(2):239–251, 1986.

[21] M. ur Rehman. *Fast Iterative Methods for The Incompressible Navier-Stokes Equations.* PHD Thesis, Delft University of Technology, The Netherlands, 2010.

[22] M. ur Rehman, C. Vuik, and G. Segal. A comparison of preconditioners for incompressible Navier-Stokes solvers. *International Journal for Numerical Methods in Fluids*, 57:1731–1751, 2008.

[23] M. ur Rehman, C. Vuik, and G. Segal. SIMPLE-type preconditioners for the Oseen problem. *International Journal for Numerical Methods in Fluids*, 61(4):432–452, 2009.

[24] C. Vuik, A. Saghir, and G.P. Boerstoel. The Krylov accelerated SIMPLE(R) method for flow problems in industrial furnaces. *Int. J. Numer. Meth. Fluids*, 33(7):1027–1040, 2000.

[25] S. Ø. Wille and A.F.D. Loula. A priori pivoting in solving the Navier-Stokes equations. *Commun. Numer. Meth. Engng.*, 18(10):691–698, 2002.