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

A ROBUST PARALLEL ILU SOLVER WITH GRID-INDEPENDENT CONVERGENCE FOR THE COUPLED STEADY INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

Fred W. Wubs and Jonas Thies

Johann Bernoulli Institute of Mathematics and Computing Science, University of Groningen,
P.O.Box 407, 9700 AK Groningen, The Netherlands
e-mail: f.w.wubs,j.thies@rug.nl

Key words: Saddle point problem, incomplete factorization, grid-independent convergence, incompressible Navier-Stokes equations, constraint preconditioning

Abstract. *We present a new hybrid direct/iterative approach to the solution of the steady incompressible Navier-Stokes equations on an Arakawa C-grid. The two-level method described here has the following properties: (i) it is very robust, even close to the point where the solution becomes unstable; (ii) a single parameter controls fill and convergence, making the method straightforward to use; (iii) the convergence rate is independent of the number of unknowns; (iv) the matrix on the second level has the same structure and numerical properties as the original problem, so the method can be applied recursively.*

1 INTRODUCTION

Presently, a typical computational fluid dynamics (CFD) problem may involve millions of unknowns. They represent velocities and pressures on a grid and are determined by solving a large sparse linear system of equations. Robust numerical methods are needed to achieve high fidelity. The hybrid direct/iterative approach presented here[1] seeks to combine the robustness of direct solvers with the memory and computational efficiency of iterative methods. It is based on the direct method[2] recently developed for the Stokes $\mathcal{F}$-matrix, which has the property that the fill does not increase in the "gradient" and "divergence" part of the matrix. To extend this to an incomplete factorization preconditioner one only has to drop velocity-velocity couplings to limit the amount of fill. We perform a non-overlapping domain decomposition of the grid, and eliminate the interior velocities using a direct method. For the remaining variables a Schur-complement problem has to be solved, which we do by a Krylov subspace method preconditioned by a novel incomplete factorization preconditioner.

The idea of combining direct and iterative methods has been used by Henon & Saad[3] and Gaidamour[4] to solve general sparse linear systems arising from the discretization of scalar PDEs. As in this paper, they reduce the problem to a Schur-complement system on the separators of a domain decomposition. However, the structural and numerical properties are not explicitly preserved which will make it hard to get a robust method and to ascertain grid-independent convergence.

For saddle point problems a lot of approaches can be followed, see for instance the survey by Benzi et al.[5]. Many of the iterative methods are segregated approaches, i.e. the velocities are solved independently from the pressures. This results in inner and outer iterations, the former for the two or more independent systems, and the latter to bring the solutions of these systems into balance with each other. In general this comes with quite a few parameters to be tuned. The method in this paper is merely a perturbation of an optimal direct approach which has only one parameter to be chosen.

In this paper we start out by reviewing the direct method[2]. In section 3 we will describe the iterative procedure based on this direct method. In section 4 we present numerical results for the incompressible Navier-Stokes equations. We conclude in section 5 by summarizing the method and results.

2 $\mathcal{F}$-MATRICES AND THE DIRECT SOLUTION METHOD

In this paper we study the solution of the equation

$$Kx = b, \tag{1}$$

where $K \in R^{(n+m)\times(n+m)}$ $(n \geq m)$ is a saddle point matrix that has the form

$$K = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}, \tag{2}$$

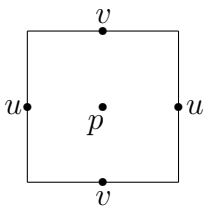

Figure 1: Positioning of velocity (u, v) and pressure (p) variables in the C-grid.

with $A \in R^{n\times n}$, $B \in R^{n\times m}$. Special attention is given to a class of saddle point matrices known as $\mathcal{F}$-matrices[6]. We start out by defining the gradient matrix in which the $\mathcal{F}$-matrix is expressed.

Definition 1 *A gradient-matrix has at most two nonzero entries per row and its row sum is zero.*

We have chosen the name gradient-matrix, because this type of matrix typically results from the discretization of a pressure gradient in flow equations. It is important to note that the definition allows a gradient-matrix to be non-square. Now we can define the $\mathcal{F}$-matrix.

Definition 2 *A saddle point matrix (2) is called an $\mathcal{F}$-matrix if A is positive definite and B is a gradient-matrix.*

$\mathcal{F}$-matrices occur in various fluid flow problems where Arakawa A-grids (collocated) or C-grids (staggered, see figure 1) are used. For example, in the discretization of Darcy's equation in ground-water flow[7] results in an $\mathcal{F}$-matrix.

Many of the standard algorithms have in common that they compute a fill-reducing ordering for K and then somehow adapt it to make it feasible: a factorization is feasible if it does not break down due to a zero pivot. The delay of elimination (through pivoting) will give an increase in computing time and may lead to increased fill in the factors. To preclude this inefficiency we propose a different approach. Suppose the sets of all velocities and pressures are denoted by V and P, respectively. The respective elements will be called V-nodes and P-nodes. The idea is to first compute an ordering for the V-nodes based on a graph that contains information of the whole matrix, and then insert the P-nodes appropriately. Assume that we have an elimination ordering on V, then we use the following simple rule to insert P-nodes into the ordering:

Rule 1 *During Gaussian elimination with K, whenever a V-node is to be eliminated which is connected to a P-node, these nodes are eliminated together using a 2×2 pivot.*

With this rule we get as many 2×2 pivots as there are P-nodes. Only if due to elimination a V-node becomes totally disconnected from P it can be eliminated on its own.

As all P-nodes are eliminated together with a V-node in pivots of the form

$$\begin{bmatrix} \alpha & \beta \\ \beta & 0 \end{bmatrix}, \tag{3}$$

the factorization is always feasible and additional pivoting is not required.

If we apply this rule to an ordering on V that is constructed as a fill-reducing ordering for A, the resulting ordering for K will not be fill-reducing in general. To ensure that the final ordering is fill-reducing we have to use information about the whole matrix, i.e. the fill patterns of B and B^T have to be taken into account. This is the case if the ordering for V is fill-reducing for the fill pattern $F(A) \cup F(BB^T)$, where $F(A)$ denotes the fill pattern of A. This graph is an envelope for the fill that will be created by elimination of the nodes in P. In many cases this will be equal to $F(A + BB^T)$, but to avoid possible cancellation in the addition we will use the matrix $F(A) \cup F(BB^T)$. Summarizing we get the following algorithm:

Algorithm 1 *To compute a feasible fill-reducing ordering for the saddle point matrix K:*

1. *Compute a fill-reducing ordering for the V-nodes based on $F(A) \cup F(BB^T)$.*
2. *Insert the P-nodes into the ordering according to rule 1.*

The P-nodes (step 2) can be inserted dynamically during Gaussian elimination, which means that we have to adapt the elimination process. The elimination is performed using the fill-reducing ordering on V and applying rule 1. This also takes into account that V-nodes initially coupled to P-nodes become decoupled because of cancellation, which is a rather common phenomenon (see section 3.2). This is different from just combining pressures with velocities beforehand (static pivoting).

The above method has structure preserving properties[1,2], one of which is in the theorem below.

Theorem 1 *If K is an $\mathcal{F}$-matrix, all Schur complements $K^{(l)}$ are $\mathcal{F}$-matrices.*

This means that the A part will remain positive definite and the B part will have at most 2 entries per row in any step of the elimination. The latter allows us to keep the B part exact during the incomplete factorization.

3 STRUCTURE PRESERVING INCOMPLETE FACTORIZATION

In this section we want to develop an incomplete factorization based on the direct method described so far. First we will introduce the domain decomposition we use and then we will illustrate that simply applying a dropping strategy to the A part may not give the desired result when there are couplings to P-nodes. We then proceed to develop a combination of orthogonal transformations and dropping that leads to grid-independent convergence, limits fill-in and keeps the divergence constraint intact.

Assumption For this section we will assume that the entries in B have equal magnitude. This is not a restriction because it can be achieved by scaling the rows of an arbitrary gradient matrix B. If DB gives the desired matrix, our new matrix will be

$$\begin{bmatrix} DAD & DB \\ B^T D & O \end{bmatrix} \tag{4}$$

Observe that the post-scaling means that the V-nodes will be scaled. For Navier-Stokes on a stretched grid (see section 4) the scaling is such that we get as new unknowns the fluxes through the control cell boundaries.

3.1 Domain decomposition

The first step of the proposed method is to construct a non-overlapping decomposition of the physical domain into a number of subdomains. This can be done by applying a graph-partitioning method like Metis[8] or similar libraries to $F(A) \cup F(BB^T)$. Metis has been tested successfully, but for this paper we use a manual partitioning into equally-sized square subdomains. (For the Navier-Stokes equations we used a stretched grid, so in that case they are not square and equally-sized in physical space but in the number of unknowns).

Then we introduce a minimal overlap: two adjacent subdomains share one layer of velocity nodes, whereas pressure nodes are not shared among subdomains. Variables belonging to exactly one subdomain are said to be *interior variables*. Velocities connecting to interior variables in more than one subdomain form separators of the subdomains they connect to. The separator velocities are complemented by an arbitrary single P-node per subdomain. When eliminating the interior variables in the next step, this ensures that the subdomain matrix is non-singular (in physical terms the pressure level inside the subdomain is fixed).

We can now eliminate the interior variables, leading to a Schur-complement problem for the separator velocities and remaining pressures. The remainder of this section is devoted to constructing an incomplete factorization preconditioner for this Schur-complement, so that it can be solved efficiently by a Krylov subspace method.

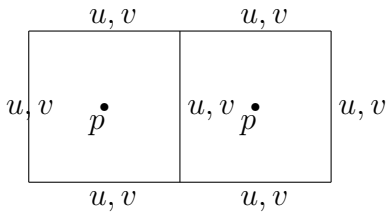

Figure 2: Velocity separators (u, v) and pressure per domain (p) in a 2-domains case.

3.2 The dropping problem

Consider the following matrix, which occurs in any elimination step with a 2×2 pivot:

$$\left[\begin{array}{cc|cc} \alpha & \beta & a^T & b^T \\ \beta & 0 & \hat{b}^T & 0 \\ \hline a & \hat{b} & \hat{A} & \hat{B} \\ b & 0 & \hat{B}^T & O \end{array}\right]. \tag{5}$$

When performing the elimination step, a multiple of $\hat{b}\hat{b}^T$ is added to $\hat{A}$. This does not introduce new fill if $\hat{A}$ is dense. But if we replaced $\hat{A}$ by a sparse matrix by dropping, the matrix would be filled again as $\hat{b}$ is typically dense.

This is a common phenomenon. Consider, for example, the two-domain case in fig. 2. After eliminating the interior variables, many of the V-nodes on the separator are coupled to the two remaining P-nodes. Assume that we drop all connections between the V-nodes on the separator, so in the above matrix (5), $\hat{A}$ is replaced by its diagonal, and a becomes zero; $\hat{b}$ is a dense vector, $\hat{B}$ has an associated dense column with opposite sign, and b^T has a nonzero at the same column position with sign opposite to that of β. When eliminating one "V-node P-node" pair, all the V-nodes on the separator become detached from P and $\hat{A}$ becomes dense.

From the above we learn that we should try to get more zeros into $\hat{b}$. Or stated otherwise, we should try to decouple the V-nodes on the separator from the P-nodes as far as possible.

3.3 Orthogonal operators to decouple V- and P-nodes

One idea to get rid of unwanted pressure couplings is to simply drop them. However, the fill in the B-part is already modest and an exact B-part is attractive, as discussed in section 2. Fortunately we can do better. Consider the square domain decomposition (fig. 2), extended periodically so that every subdomain is bounded by four separators from the neighboring subdomains. The Schur-complement for the separator velocities and remaining pressures has about the following form (the V-nodes in the corners are neglected here, in practice they form 'separators of the separators' and get a block of

their own):

$$\begin{bmatrix} A_{11} & B_1 & A_{12} & A_{13} & O & O \\ B_1^T & O & B_{21}^T & B_{31}^T & O & O \\ A_{21} & B_{21} & A_{22} & O & A_{24} & B_{22} \\ A_{31} & B_{31} & O & A_{33} & A_{34} & B_{32} \\ O & O & A_{42} & A_{43} & A_{44} & B_{42} \\ O & O & B_{22}^T & B_{32}^T & B_{42}^T & O \end{bmatrix} \begin{bmatrix} v_1 \\ p_1 \\ v_2 \\ v_3 \\ v_4 \\ p_2 \end{bmatrix} = \begin{bmatrix} b_{v_1} \\ b_{p_1} \\ b_{v_2} \\ b_{v_3} \\ b_{v_4} \\ b_{p_2} \end{bmatrix}. \tag{6}$$

Here v_1 contains the V-nodes on a certain separator, p_1 contains the two P-nodes from the adjacent subdomains; v_2 and v_3 contain the V-nodes from other separators around these subdomains, respectively. v_4 and p_2 represent the remaining V- and P-nodes in the Schur-complement (separator velocities and pressures not connected to the separator under consideration).

Now B_1 only contains two dense columns, equal up to a sign. So by using an orthogonal transformation H, e.g. a Householder reflection, we can transform B_1 into a matrix with only entries on a certain row, usually the first. Applying H to the first block row and column from left and right, respectively, we obtain the following system (note that the properties of the matrix are preserved by the orthogonal transformation):

$$\begin{bmatrix} H^TA_{11}H & H^TB_1 & H^TA_{12} & H^TA_{13} & O & O \\ (H^TB_1)^T & O & B_{21}^T & B_{31}^T & O & O \\ A_{21}H & B_{21} & A_{22} & O & A_{24} & B_{22} \\ A_{31}H & B_{31} & O & A_{33} & A_{34} & B_{32} \\ O & O & A_{42} & A_{43} & A_{44} & B_{42} \\ O & O & B_{22}^T & B_{32}^T & B_{42}^T & O \end{bmatrix} \begin{bmatrix} H^Tv_1 \\ p_1 \\ v_2 \\ v_3 \\ v_4 \\ p_2 \end{bmatrix} = \begin{bmatrix} H^Tb_{v_1} \\ b_{p_1} \\ b_{v_2} \\ b_{v_3} \\ b_{v_4} \\ b_{p_2} \end{bmatrix}. \tag{7}$$

The Householder matrix is a full matrix (though its application is cheap if its defining form is exploited) and would destroy the sparsity. However, the matrices A_{11}, A_{12} and A_{13} are typically already dense, so not much is lost and we have gained a lot: we decoupled all but one of the V-nodes on the separator from the P-nodes. The decoupled ones can be eliminated on their own now.

The situation depicted in eq. 5 now only occurs once per separator and velocity component, namely for the V-node still coupled to the P-nodes. Because of the transformation $\hat{b}$ is now zero, and no fill is generated.

So far we have not made any approximations, and while we have zeroed out most of the V-node/P-node couplings, a dropping strategy has to be applied in the V-V part to get a sparse preconditioner for the Schur-complement. However, the Householder transformation combined with standard dropping techniques for the SPD case will generally not lead to grid independent convergence. This requires that the approximation is spectrally equivalent to the original matrix. We will consider a new way of dropping in the next section which has this property.

3.4 Dropping strategy

The general idea of the approximation is the following. We replace the flux through grid cell faces forming a separator by the combined flux through that separator. Then we try to reduce the problem of finding all separator velocities by dropping and elimination to the related problem of finding the new fluxes (or summed velocities). This reduced problem can still be understood in terms of conservation of mass and momentum and its form is very similar to the original problem.

Let us consider an orthogonal operator that is more intuitive than the Householder transformation. Suppose e is a vector with all ones and C is an orthogonal extension of e such that the length of every column is the same. Define a square matrix

$$H = [C, e], \tag{8}$$

which is orthogonal up to a constant factor. This operator is applied to the velocity component in normal direction on the separator. These velocities have the same sign for the connection to the pressure and therefore again only one row remains in $H^T B_1$. The first component of $H^T v$ will be the sum of the components of v; we will call this a V_Σ-node from now on.

The following lemma and its corollary play a key role in devising a dropping strategy:

Lemma 1 *Principal submatrices of an (S)PD-matrix are (S)PD.*

Corollary 1

$$\textit{If } \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \textit{ is (S)PD then } \begin{bmatrix} A_{11} & O \\ O & A_{22} \end{bmatrix} \textit{ is (S)PD.} \tag{9}$$

Since we only make approximations in the A part of the matrix K, we have the following lemma.

Lemma 2 *If A is SPD, the condition number of the preconditioned K matrix is bounded by the condition number of the preconditioned A, where as preconditioner an SPD approximation of A is used.*

Proof: see reference 1.

These lemmas set the ground for further reasoning that will lead to grid-independent convergence. In the remainder of this section we assume that A is symmetric and positive definite. Let us extend H with an identity for the unknowns that are not transformed and write $H = [H_1, H_2]$, where

$$H_1 = \begin{bmatrix} C \\ 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} e & 0 \\ 0 & I \end{bmatrix}. \tag{10}$$

The transformed matrix is given by

$$H^T A H = \begin{bmatrix} H_1^T A H_1 & H_1^T A H_2 \\ H_2^T A H_1 & H_2^T A H_2 \end{bmatrix}. \tag{11}$$

Here $H_2^T A H_2$ is a Galerkin approximation of A and hence it can be viewed as a discretization on a coarser grid (in fact it is an aggregation similar to that used by Notay[9], albeit that Notay applies the aggregation directly to the discretized PDE whereas we apply it to its Schur complement on the separators). If A is obtained from a stable discretization of a second-order differential operator, then $H_1^T A H_1$ has a condition number independent of the mesh size if the dimension of C is fixed (i.e. if the length of the separator is fixed). For the approximation we simply drop the off-diagonal blocks in (11). Now assume we have the following strengthened Cauchy-Schwarz inequality[10]

$$|x^T H_1^T A H_2 y| \leq \gamma \{(x^T H_1^T A H_1 x)(y^T H_2^T A H_2 y)\}^{\frac{1}{2}} \tag{12}$$

holding independently of the mesh size. Using Lemma 2 we can prove the following.

Theorem 2 *If a strengthened Schwarz-inequality (12) holds for* $0 \leq \gamma < 1$ *independent of the mesh size, then we have convergence independent of the mesh-size when the dropping process as discussed above is applied. The condition number of the preconditioned* K *matrix is bounded by* $(1+\gamma)/(1-\gamma)$.

The situation above remains the same if we apply the transformation to all separators at once. After the transformation, only the unknowns associated with A_{22} are coupled to pressures. We may still have couplings between various separators in A_{11}, but the condition number of that matrix is independent of the mesh size. To lower the computational cost we also drop couplings between separators in A_{11}.

Note that we have only made approximations in the velocity-velocity part of the matrix, so this means that the "gradient" and "divergence" part is still exact. Hence, we have build a so-called constraint preconditioner[11] and for symmetric K (i.e. for a Stokes flow problem) we can use the Conjugate Gradient method. We were successful with this in reference 1 for the Stokes equation.

4 INCOMPRESSIBLE FLOW IN A LID-DRIVEN CAVITY

As test problem for the Navier-Stokes equations we use the lid-driven cavity. By Tiesinga et al.[12] this problem was studied near the transition point from steady to transient flow. The stability of steady and periodic solutions was investigated using the Newton-Picard method[13] with the θ-method for time stepping (with θ slightly larger than 0.5 in order to damp high-frequency modes which would otherwise show up as spurious eigenvalues near the imaginary axis). The linear systems that have to be solved have a slightly increased diagonal, which improves the conditioning somewhat. The MRILU[14]

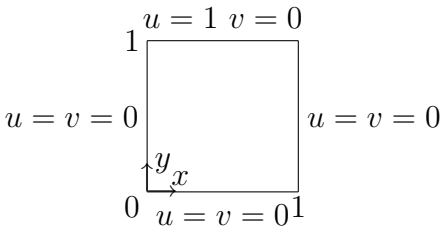

Figure 3: Geometry for the lid-driven cavity problem.

preconditioner used at the time converged slowly and not at a grid-independent rate, which we very much like to improve upon.

In a recent review[15], the performance of a number of block multi-level preconditioners is investigated for the steady problem for Reynolds numbers up to 1000. These methods also solve the coupled equations, but perform inner iterations on the velocity and pressure part separately and hence require many parameters to be tuned. Below we demonstrate robust, grid-independent convergence for the driven cavity problem at Reynolds-numbers of up to 8000.

The problem consists of calculating the flow in a square cavity with uniformly moving lid. The domain and boundary conditions of the lid-driven cavity problem are shown in fig. 3, where u and v denote the velocity in x- and y-direction, respectively.

The equations are given by

$$\left.\begin{aligned} -\mathbf{u}\cdot\nabla\mathbf{u} + \tfrac{1}{Re}\Delta\mathbf{u} - \nabla p &= 0\ , \\ \nabla\cdot\mathbf{u} &= 0\ . \end{aligned}\right\} \tag{13}$$

For the discretization we use a symmetry-preserving space discretization[16], which is stable and does not introduce artificial diffusion. Furthermore, the grid is stretched towards the boundaries in order to resolve the boundary layers. The ratio between largest and smallest mesh size is about 5. This also means that we really need to change to fluxes through grid cell boundaries instead of velocities in order to get the required property that all elements in B have the same magnitude (see the beginning of section 3). The system matrix is the Jacobian from the first step of the Newton method at the current Reynolds number. In order to avoid convergence problems of Newton's method, we use the result at the previous Reynolds-number as a starting solution (the Reynolds numbers used are shown in table 1).

In the tables the following data is displayed: n_x - the grid size (the grid is $n_x \times n_x$), s_x - the subdomain size (the subdomain is $s_x \times s_x$), N - number of unknowns (size of the saddle point matrix), nnz - number of nonzeros in original matrix, $\mathrm{N_S}$ - number of unknowns on the separators and remaining p's (size of the Schur-complement), n - number of $V_\Sigma's$ and remaining p's (size of reduced Schur-complement), iter - number of GMRES iterations performed on the Schur-complement to reduce the residual norm by $1/\mathrm{tol} = 10^6$, fill 1 - grid-independent part of relative fill-in (number of nonzeros in the solver divided by

number of nonzeros in original matrix), and fill 2 - grid-dependent part of relative fill-in, generated when factoring the $n \times n$-dimensional reduced Schur-complement. The grid-independent part fill 1 consists of: (a) fill-in generated while factoring the subdomain matrices, (b) fill-in generated while constructing the Schur-complement, and (c) fill-in generated while factoring the separator-blocks of the preconditioner.

We first focus on the effect of increasing the Reynolds-number (cf. table 1). The convergence is not independent of the Reynolds-number. In our view this is not surprising, because the underlying continuous problem changes with the Reynolds number and more and more eigenvalues are getting close to the origin. This is different from the dependence on the mesh, where the continuous problem stays the same and all eigenvalues near the origin stay at their place.

Next we refine the grid at a high Reynolds-number of 8000, close to the point, such as described in Tiesinga et al.[12], where the steady state becomes unstable; results are shown in table 2. Note that the number of iterations is going down as we decrease the mesh size. This is because with decreasing mesh size the physical size of the subdomains is decreasing if we keep the number of unknowns per subdomain the same. As the physical subdomain decreases, the diffusion plays a more important role than the advection on that scale. Since the approximations take place at the subdomain scale, the convergence behavior tends to that of the Stokes problem.

Re	N	nnz	N_S	n	iter	fill 1	fill 2
500	785 408	6 794 252	129 025	40 069	59	6.41	2.59
1000	785 408	6 794 252	129 025	40 069	73	6.39	2.59
2000	785 408	6 794 252	129 025	40 069	87	6.38	2.65
4000	785 408	6 794 252	129 025	40 069	104	6.35	2.78
8000	785 408	6 794 252	129 025	40 069	130	6.33	2.72

Table 1: 2D Driven cavity - increasing Reynolds-number, grid-size $n_x = 512$

n_x	N	nnz	N_S	n	iter	fill 1	fill 2
64	12 160	103 820	1 793	533	185	6.09	0.418
128	48 896	420 620	7 681	2 341	181	6.22	0.953
256	196 096	1 693 196	31 745	9 797	167	6.29	1.75
512	785 408	6 794 252	129 025	40 069	130	6.33	2.72

Table 2: 2D Driven cavity - grid refinement at $Re = 8000$

We conclude by mentioning that with the resulting preconditioner it was also quite easy to compute eigenvalues using MATLAB's eigs routine (i.e. ARPACK). Hence we can now study the stability problem near the point where the steady state becomes unstable using eigenvalue analysis.

5 DISCUSSION AND CONCLUSIONS

In this paper we have shown the basics of an incomplete factorization for Navier-Stokes Jacobian matrices. This is an incomplete factorization for the whole system, which avoids having to balance inner and outer iterations as in a segregated approach. Depending only on a single parameter (the subdomain size), the method is as easy to use as a direct solver and gives reliable results in a reasonable turn-around time.

For Stokes matrices we were able to prove grid-independent convergence. The total number of operations required is currently not grid-independent since we use a direct solver to handle the reduced system. However, the amount of work required for this step is reduced by about the cube of the subdomain size in 2D and the sixth power in 3D. So increasing the subdomain size by a factor 2 means in 2D a factor 8 and in 3D a factor 64 gain in computing time. For the Navier-Stokes equations we also observed grid-independent convergence. Currently, we are developing a parallel C++ implementation of the method that can be applied recursively, turning it into a multi-level method. Herewith we get rid of the costs for the direct solver on the reduced system.

We showed the robustness of the method for the Stokes equations. For the Navier-Stokes (a generalization of Stokes) equations the results show high robustness. For Reynolds numbers up to 8000 (and even for Reynolds numbers bigger than 8000, which are not shown here), convergence problems only occurred in the Newton method, never in the linear solver. So the method still performs well for cases where eigenvalues pass the imaginary axis somewhere away from the origin (Hopf bifurcations).

Acknowledgements

Part of this work was done during a sabbatical leave of the first author to the Numerical Analysis Group at Rutherford Appleton Laboratory in Chilton (UK). We kindly thank that group for the hospitality and good-fellowship. We also thank the London Mathematical Society for funding this stay under grant 2803. The research of the second author was funded by the Netherlands Organization for Scientific Research, NWO, through the contract ALW854.00.028.

REFERENCES

[1] Wubs, F.W. and Thies., J., A robust two-level incomplete factorization for (Navier-) Stokes saddle point matrices, submitted for publication.

[2] Niet, A.C.de and Wubs, F.W. (2009) Numerically stable LDL^T-factorization of F-type saddle point matrices, *IMA J. Numer. Anal.*, **29**:208-234, 2009.

[3] Hénon, P. and Saad, Y. (2006) A Parallel Multistage ILU Factorization Based on a Hierarchical Graph Decomposition. *SIAM J. of Sci. Comput.,* **28**:2266-2293

[4] Gaidamour, J. and Hénon, P. (2008) A parallel direct/iterative solver based on a Schur complement approach, Proceedings of *IEEE 11th International Conference on Computational Science and Engineering, Sao Paulo, Brazil,* p.98-105.

[5] Tůma, M. (2002) A note on the LDL^T decomposition of matrices from saddle-point problems, *SIAM J. Matrix Anal. Appl.,* **23**(4):903–915.

[6] Arioli, M. and Manzini, G. (2003) Null space algorithm and spanning trees in solving Darcy's equation, *BIT,* **43**(5), 839–848.

[7] Karypis, G. and Kumar, V. (1998) A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM J. Sci. Comput.,* **20**(1):359–392.

[8] Notay, Y (2010) An aggregation-based algebraic multigrid method, *ETNA*,To appear

[9] Axelsson, O. (1994) *Iterative Solution Methods,* Cambridge University Press.

[10] Gould, N. I. M., Hribar, M.E. and Nocedal J. (2001) On the solution of equality constrained quadratic programming problems arising in optimization, *SIAM J. Sci. Comput.*, **23**:13761395.

[11] Tiesinga, G., Wubs, F.W. and Veldman, A.E.P. (2002) Bifurcation analysis of incompressible flow in a driven cavity by the Newton-Picard method, *J. Comput. Appl. Math.*

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

[13] Lust, K., Roose, D., Spence, A. and Champneys, A.R.(1999) An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions *SIAM J. Sci. Comput.*, **19**(4):1188-1209, 1998.

[14] Botta, E.F.F. and Wubs, F.W. (1999) Matrix Renumbering ILU: An effective algebraic multilevel ILU-preconditioner for sparse matrices, *SIAM J. Matrix Anal. Appl.*, **20**(4):1007–1026.

[15] Elman, H., Howle, V.E., Shadid, J., Shuttleworth, R., and Tuminaro, R. A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, **227**:1790-1808, 2008

[16] Verstappen, R.C.W.P. and Veldman, A.E.P. (2003) Symmetry-preserving discretisation of turbulent flow, *J. Comp. Phys.*, **187**:343-368.