AN INCOMPLETE *LU* PRECONDITIONER

FOR SCHUR COMPLEMENT MATRICES

N. GUESSOUS[*] *and O. SOUHAR*[**]

November 29, 2005

[*] *Ecole Normale supérieure, Bensouda B.P. 5206 Fès, Morocco*
ngessous@yahoo.fr
[**] *CEMAGREF Groupement de Lyon- 3 quai Chauveau*
69336 Lyon Cedex 09, France
drsouhar@hotmail.com

Abstract: We recall in this paper a general preconditioning method based on block Incomplete *LU* factorization for solving general sparse linear systems. Two-level factorization methods for 5-point difference matrices are analyzed, these methods use block red-black ordering of the meshes, the basic scheme assumes an exact inversion of the submatrix related to the first block of unknowns then forming the reduced system explicitly. We compare various incomplete *LU* factorizations to approximate the Schur complement matrix. Analytical bounds for the Schur complement and preconditioned error matrices are presented.

Key words: Sparse linear systems, ordering methods, parallel algorithms, block incomplete *LU* factorization, Schur complement, convection-diffusion equations.

AMS subject Classifications: 65F10, 65N06, 65N22.

1 Introduction

This paper deals with the iterative solution of large sparse nonsymmetric systems:

$$Au = b, \tag{1.1}$$

where A is a large and sparse real matrix of order n. In such cases, the Krylov subspace methods combined with a limited preconditioning are choice methods. Among preconditioning methods investigated by various groups, a class of ILU-type techniques has emerged that possesses many of attributes of multilevel solvers [1, 2, 5, 6, 9, 16, 18, 19, 23-26, 29-33].
It is well known that the degree of parallelism in the application of the standard ILU preconditioner (preconditioner arising from ILU factorization without fill-in) is limited. An approach to increase the parallelism in ILU preconditioner methods is to use

domain decomposition methods. In this approach, the physical domain is decomposed into a number of subdomains on each of which an independent incomplete factorization can be computed and applied in parallel. The main idea is to obtain more parallelism at the subdomain level rather than at the grid point level. Usually, the interfaces between the subdomains must be treated in a special manner. This approach is quite general since can be combined with different methods within different subdomains. For different variants of this technique; see, [5-7, 21, 26].
Another approach that also improves the degree of parallelism in the application of the ILU preconditioner is to use the multi-color ordering. The problem addressed by multi-coloring is to determine a coloring of the nodes of the adjacency graph of a matrix such that any two adjacent nodes have different colors.
For a simple two-dimensional finite difference grid (five-point operator), we can easily separate the grid points into two sets, red and black, so that the nodes in one set are adjacent only to nodes from the other set. Assume that the unknowns are labeled by listing the red unknowns first together, followed by the black ones. We will obtain a system of the form

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.2}$$

where B and C are diagonal matrices.
The way to exploit the red-black ordering is to use the standard $SSOR$ or $ILU(0)$ preconditioners for solving (1.2). The preconditioning operations are highly parallel. For example, the linear system that arises from the forward solve in $SSOR$ will have the form

$$\begin{pmatrix} B & \\ E & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{1.3}$$

This system can be solved by performing the following sequence of operations:

1. Solve $Bx = f$.
2. Compute $\tilde{g} = g - Ex$.
3. Solve $Cy = \tilde{g}$.

This consists of two diagonal scaling (operations 1 and 3) and a sparse matrix-by-vector product (operation 2). The situation is identical with the $ILU(0)$ preconditioning. The structure of the ILU factors reveals that many more elements are dropped with the red-black than with the natural ordering, then the result is that the number of iterations to achieve convergence can be much higher with red-black than with the natural ordering [26].
A second method that has been used in connection with the red-black ordering solves the reduced system which involves only the black unknowns. Eliminating the red unknowns from (1.2) results in the reduced system:

$$(C - EB^{-1}F)y = g - EB^{-1}f. \tag{1.4}$$

Note that this new system is again a sparse linear system with about half as many unknowns. In addition, it has been observed that for easy problems, the reduced system can often be solved efficiently with only diagonal preconditioning. The computation of the reduced system is a highly parallel and inexpensive process.
We can also use a multi-level ordering of the grid points, for example Brand and Heinmann proposed a repeated red-black (RRB) ordering; see, [4] and [10] for more details on this technique and the order of the condition number. One notable drawback of multi-coloring is that the efficiency of the preconditioning on the re-ordered system deteriorates, compared with the original system [12].
This paper is organized as follow: Section 2 discusses a LU factorization technique to construct the Schur complement matrix explicitly and gives the global preconditioner. Section 3 shows some bounds for the Schur complement and the preconditioned error matrices, these results will guide to a robust ILU preconditioner. Section 4 contains comparisons numerical experiments which present the quality of this paper. Some concluding comments are included at the end.

2 Block red-black ordering and preconditioners

2.1 Block red-black coloring

In general, in block factorizations, a pivot block is forced to be sparse, and that we need an approximation to its inverse that has a similar structure [10, 12, 29, 30]. Furthermore, this approximation should be easily computable.

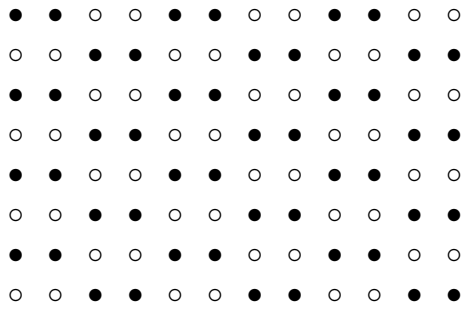

FIG.1. Block Red-Black Coloring of a 12×8 grid.

We now consider an extension of the red-black coloring which consists of transforming the system (1.1) into the following block form

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{2.1}$$

here B and C are block diagonal matrices, where each block is of size 2.
A block of size 2 will be found by coupling a node j with its neighbor $(j+1)$ with the same color (Red) followed by two nodes that are different color (Black), we assume again that the unknowns are labeled by listing the red unknowns first together, followed by the black ones. This block two-by-two colors is illustrated in Figure 1 for a 12×8

grid (with white nodes corresponding to red nodes).
Then we have a 2×2 submatrix of the form

$$\begin{pmatrix} b_{j,j} & b_{j,j+1} \\ b_{j+1,j} & b_{j+1,j+1} \end{pmatrix}. \tag{2.2}$$

One of the motivations of this block red-black coloring is to increase the degree of parallelism compared to the consecutive numbering by rows and columns, since we can invert the block diagonal matrix B exactly and in parallel.

2.2 Block-partitioned preconditioners

We can use the block factorization of A

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} B & \\ E & S \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ & I \end{pmatrix}, \tag{2.3}$$

where S is the Schur complement matrix

$$S = C - EB^{-1}F. \tag{2.4}$$

Since B is a block diagonal matrix, where each block is of small size, then its inverse should be computed exactly and in parallel. We can solve (2.1) by solving the reduced system

$$Sy = \tilde{g} \text{ with } \tilde{g} = g - EB^{-1}f, \tag{2.5}$$

to compute y, and then by backward substitution

$$x = B^{-1}(f - Fy), \tag{2.6}$$

to compute x. It is well known that the above block structure (2.3) can be exploited in different ways to define preconditioners for A. To do so, we define the block preconditioner

$$M = \begin{pmatrix} B & \\ E & \tilde{S} \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ & I \end{pmatrix}, \tag{2.7}$$

where $\tilde{S}$ is some approximation to the Schur complement matrix S, for example in the form of an approximate LU factorization such as $ILUT(\tau, l_{fil})$ [24].
In general; for unstructured matrices, the cost to compute the exact inverse of B is prohibitive, then to approximate the Schur complement S with a sparse matrix, we first approximate B^{-1} by a sparse matrix Y using some approximate inverse techniques; see, [1, 3, 8, 14, 15]. Since Y is sparse, the matrix $\bar{S} = C - EYF$ is also sparse hence its application is more economical since S is often a dense matrix.
We can also use an approximate factorization of $\bar{S}$

$$\bar{S} = L_{\bar{S}} U_{\bar{S}} + R_{\bar{S}}, \tag{2.8}$$

to define a preconditioner $\tilde{S} = L_{\bar{S}}U_{\bar{S}}$ for the Schur complement matrix S. When C is not singular we can choose $\tilde{S} = C$, or $\tilde{S} = I$ i.e. no preconditioning.
We can easily show that the preconditioned system has the particular form

$$M^{-1}A = \begin{pmatrix} I & -B^{-1}F \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & \tilde{S}^{-1}S \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ & I \end{pmatrix} = U_A^{-1}DU_A. \tag{2.9}$$

We conclude that the efficiency of the global preconditioner M depends strongly on how well $\tilde{S}$ approximates the Schur complement matrix S.

3 Analysis

Consider the incomplete factorization

$$A = M + R = LU + R, \tag{3.1}$$

where M is the incomplete LU factorization of A defined by (2.7) and R is the error matrix.
¿From (2.7) we can write

$$S = \tilde{S} + R_S, \tag{3.2}$$

where $\tilde{S}$ is an incomplete LU factorization of S and R_S is the matrix of elements that have been dropped, then

$$R = \begin{pmatrix} 0 & 0 \\ 0 & R_S \end{pmatrix}. \tag{3.3}$$

We suppose that B is of dimension m and C is of dimension $l = n - m$. We use in latter the following notations:
$B = (b_{i,j})_{m\times m}$, $F = (f_{i,j})_{m\times l}$, $E = (e_{i,j})_{l\times m}$, $C = (c_{i,j})_{l\times l}$, $S = (s_{i,j})_{l\times l}$, $B^{-1} = (\beta_{i,j})_{m\times m}$.
An arbitrary element of the Schur complement matrix S is

$$s_{i,j} = c_{i,j} - \sum_{k=1}^{m}\sum_{q=1}^{m} e_{i,k}\beta_{k,q}f_{q,j}.$$

Due to the particular structure of B we can easily show that

$$s_{i,j} = c_{i,j} - \sum_{k=1}^{m}(e_{i,k}\beta_{k,k}f_{k,j} + e_{i,k}\beta_{k,k+1}f_{k+1,j}), \tag{3.4}$$

or

$$s_{i,j} = c_{i,j} - \sum_{k=1}^{m}(e_{i,k}\beta_{k,k-1}f_{k-1,j} + e_{i,k}\beta_{k,k}f_{k,j}). \tag{3.4$'$}$$

Assume that a given element $\beta_{i,j}$ of B^{-1} satisfies the inequality

$$|\beta_{i,j}| \leq \alpha \quad 1 \leq i,j \leq m, \tag{3.5}$$

for some positive real α. Denote $\gamma = \max_{1\leq i,j\leq n}|a_{i,j}|$ the largest elements in absolute value of A.

Proposition 3.1 *The size of the elements of Schur complement matrix S is bounded by $\gamma(1+2m\alpha\gamma)$.*

Proof ¿From (3.4) we have

$$\begin{aligned} |s_{i,j}| &\leq |c_{i,j}| + |\sum_{k=1}^{m}(e_{i,k}\beta_{k,k}f_{k,j} + e_{i,k}\beta_{k,k+1}f_{k+1,j})| \\ &\leq |c_{i,j}| + \sum_{k=1}^{m}(|e_{i,k}\beta_{k,k}f_{k,j}| + |e_{i,k}\beta_{k,k+1}f_{k+1,j}|) \\ &\leq \gamma + 2m\alpha\gamma^2. \end{aligned}$$

■

This result shows that the size of the elements of the Schur complement matrix cannot grow uncontrollably if α is not too large. This shows that the LU factorization is stable.
When a simple dropping strategy is applied with a threshold tolerance τ to Schur complement matrix S, i.e., all entries $(s_{i,j})$ whose absolute values are smaller than τ are dropped, whenever

$$|s_{i,j}| < \tau, \tag{3.6}$$

then we have

$$\|R\|_F = \|R_S\|_F \leq \tau\sqrt{nz(R_S)} \leq \tau l, \tag{3.7}$$

where $nz(R_S)$ is the number of nonzero elements of the submatrix R_S. The bound (3.7) shows that the norm of the factorization error depends on the dropping tolerance and the number of elements dropped.
The next result will require the following inequality:

$$\|XY\|_F \leq \|X\|_F\|Y\|_2, \tag{3.8}$$

where $\|.\|_F$ and $\|.\|_2$ are the matrix Frobenius norm and the 2-norm, respectively.

Proposition 3.2 *Assume that the dropping rule* (3.6) *is applied and S is nonsingular, then*

$$\|(LU)^{-1}R\|_F \leq \tau l(1+\sqrt{2m}\alpha\|F\|_2)\|\tilde{S}^{-1}\|_2. \tag{3.9}$$

Proof Starting from

$$(LU)^{-1}R = \begin{pmatrix} I & -B^{-1}F \\ 0 & I \end{pmatrix}\begin{pmatrix} B^{-1} & 0 \\ -B^{-1}E\tilde{S}^{-1} & \tilde{S}^{-1} \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & R_S \end{pmatrix},$$

then

$$(LU)^{-1}R = \begin{pmatrix} I & -B^{-1}F \\ 0 & I \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & \tilde{S}^{-1}R_S \end{pmatrix},$$

using inequality (3.8)

$$\|(LU)^{-1}R\|_F \leq \|\begin{pmatrix} I & -B^{-1}F \\ 0 & I \end{pmatrix}\|_2\|\begin{pmatrix} 0 & 0 \\ 0 & \tilde{S}^{-1}R_S \end{pmatrix}\|_F,$$

we can show that

$$\begin{aligned} \| \begin{pmatrix} I & -B^{-1}F \\ 0 & I \end{pmatrix} \|_2 &\leq 1 + \|B^{-1}F\|_2 \\ &\leq 1 + \|B^{-1}\|_2 \|F\|_2. \end{aligned}$$

It is well known that

$$\begin{aligned} \|B^{-1}\|_2 &\leq \|B^{-1}\|_F = (\textstyle\sum_{j=1}^{m} \sum_{i=1}^{m} \beta_{i,j}^2)^{\frac{1}{2}} \\ &\leq \sqrt{2m}\alpha \end{aligned}$$

then

$$\| \begin{pmatrix} I & -B^{-1}F \\ 0 & I \end{pmatrix} \|_2 \leq 1 + \sqrt{2m}\alpha \|F\|_2.$$

On the other hand, using inequality (3.8)

$$\begin{aligned} \| \begin{pmatrix} 0 & 0 \\ 0 & \tilde{S}^{-1}R_S \end{pmatrix} \|_F &= \|\tilde{S}^{-1}R\|_F \\ &\leq \|\tilde{S}^{-1}\|_2 \|R\|_F, \end{aligned}$$

from (3.7) we deduct

$$\|\tilde{S}^{-1}R\|_F \leq \tau l \|\tilde{S}^{-1}\|_2.$$

■

Remarks

1. Similarly, the norm of the inverse of the preconditioner is bounded by:
   (3.10)
   $$\|U^{-1}L^{-1}\|_2 \leq (1 + \sqrt{2m}\alpha\|F\|_2) \times (1 + \sqrt{2m}\alpha\|E\|_2) \times \max(\sqrt{2m}\alpha, \|\tilde{S}^{-1}\|_2).$$

   Indeed;

   $$\begin{aligned} (LU)^{-1} &= \begin{pmatrix} I & B^{-1}F \\ & I \end{pmatrix}^{-1} \begin{pmatrix} B & \\ & \tilde{S} \end{pmatrix}^{-1} \begin{pmatrix} I & \\ EB^{-1} & I \end{pmatrix}^{-1} \\ &= \begin{pmatrix} I & -B^{-1}F \\ & I \end{pmatrix} \begin{pmatrix} B^{-1} & \\ & \tilde{S}^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -EB^{-1} & I \end{pmatrix}, \end{aligned}$$

   then

   $$\begin{aligned} \|U^{-1}L^{-1}\|_2 &\leq \| \begin{pmatrix} I & -B^{-1}F \\ & I \end{pmatrix} \|_2 \| \begin{pmatrix} B^{-1} & \\ & \tilde{S}^{-1} \end{pmatrix} \|_2 \| \begin{pmatrix} I & \\ -EB^{-1} & I \end{pmatrix} \|_2 \\ &\leq 1 + \sqrt{2m}\alpha\|F\|_2) \times (1 + \sqrt{2m}\alpha\|E\|_2) \times \max(\|B^{-1}\|_2, \|\tilde{S}^{-1}\|_2) \\ &\leq (1 + \sqrt{2m}\alpha\|F\|_2) \times (1 + \sqrt{2m}\alpha\|E\|_2) \times \max(\sqrt{2m}\alpha, \|\tilde{S}^{-1}\|_2). \end{aligned}$$

2. The inequality (3.5) is difficult to satisfy, but if we assume that B is strictly diagonally dominant with $b_{i,j} \leq 0$, $i \neq j$ and $b_{i,i} > 0$, then each block $\begin{pmatrix} b_{j,j} & b_{j,j+1} \\ b_{j+1,j} & b_{j+1,j+1} \end{pmatrix}$ is nonsingular and its inverse can be written

$$\begin{pmatrix} b_{j,j} & b_{j,j+1} \\ b_{j+1,j} & b_{j+1,j+1} \end{pmatrix}^{-1} = \frac{1}{det} \begin{pmatrix} b_{j+1,j+1} & -b_{j,j+1} \\ -b_{j+1,j} & b_{j,j} \end{pmatrix}, \tag{3.11}$$

where $det = b_{j,j}b_{j+1,j+1} - b_{j+1,j}b_{j,j+1}$, since B is strictly diagonally dominant then there exists $\varepsilon > 0$ such that $det > \varepsilon$ and from (3.11) we show that

$$0 \leq \beta_{i,j} < \frac{\gamma}{\varepsilon} \tag{3.12}$$

4 Numerical Experiments and Conclusion

We have tested the method described above in the case of the Poisson problem,

$$-\Delta u = f \text{ in } \Omega = (0,1)^2, \tag{4.1}$$

and the convection-diffusion equation, who plays a very important role in computational fluid dynamics to simulate flow problems,

$$-\nu\Delta u + \bar{v}.\nabla u = f \text{ in } \Omega = (0,1)^2. \tag{4.2}$$

Both problems are investigated on the unit square with Dirichlet boundary condition

$$u = 0 \text{ on } \Gamma = \partial\Omega.$$

In equation (4.2), $\bar{v}$ is the convective flow and the viscosity parameter ν governs the ratio between convection and diffusion.
We study the two problems [23, 30]:

$$\bar{v} = \begin{pmatrix} cos(\pi(x-\frac{1}{3})sin(\pi(y-\frac{1}{3}) \\ -cos(\pi(y-\frac{1}{3})sin(\pi(x-\frac{1}{3}) \end{pmatrix}, \tag{$P1$}$$

inside the circle of center $(\frac{1}{3}, \frac{1}{3})$ and radius $\frac{1}{4}$, and $\bar{v}(x,y) = 0$ outside.

$$\bar{v} = \begin{pmatrix} exp(xy-1) \\ -exp(-xy) \end{pmatrix}. \tag{$P2$}$$

We use $GMRES(20)$ with right preconditioning [28]. This method minimizes the residual and is therefore mathematically equivalent to several other generalized conjugate gradient methods [1]. The preconditioner is defined by (2.7) where the approach $\tilde{S}$ of S is performed in the latter.
The right hand side was generated by assuming that the exact solution is a vector of

all ones and the initial guess was a vector of some random numbers. The computations were stopped when the 2-norm of the residual was reduced by a factor of 10^6 i.e. $\|r_m\|_2 < 10^{-6}\|r_0\|_2$ or when 200 iterations were reached, indicated by $(-)$.
We will use the following names to denote the methods that we tested on the same matrices. These matrices were reordered using the block red-black coloring described in section 2.1 and five-point upwind finite difference scheme with uniform meshsize h in both x and y directions [34].

$M_{ILU(0)}$: Approximate block LU preconditioner, where the reduced system is solved by backward and forward substitutions, which are performed with $ILU(0)$ of S.

$M_{ILUD}(\tau)$: same as above, but $\tilde{S}$ is performed with $ILUD(\tau)$ factorization [27].

$M_{ILUT}(\tau, l_{fil})$: same as above, but $\tilde{S}$ is performed with $ILUT(\tau, l_{fil})$ factorization [24].
Note that, when we apply the $M_{ILUD}(\tau)$ factorization to approximate the Schur complement matrix S, we do not add the sum of all dropped out elements in a given row.

Table 1. Iterations to convergence for the Poisson problem with various preconditioners for $\tau = 10^{-4}$ and $l_{fil} = 10$.

	$M_{ILU(0)}$		$M_{ILUD}(\tau)$		$M_{ILUT}(\tau, l_{fil})$	
h^{-1}	iter.	spar.	iter.	spar.	iter.	spar.
17	13	2.23	3	3.79	5	2.97
33	20	2.29	4	5.33	7	3.04
65	34	2.32	4	7.20	10	3.07
105	80	2.33	5	8.19	13	3.08

All preconditioners used a safeguard (stabilization) procedure by replacing a zero pivot with $(0.0001 + \tau)r_i$, where r_i was computed as the average nonzero values of the row in question. The storage requirements for each of the above preconditioner is that of $\tilde{S}$. The storage required for $\tilde{S}$ is more difficult to estimate, but it is generally less than $2 \times l_{fil} \times l$ for $ILUT(\tau, l_{fil})$ factorization, where l_{fil} is a fill-in parameter; i.e. each row of $L_{\bar{S}}$ and each row of $U_{\bar{S}}$ will have a maximum of l_{fil} elements (escluding the diagonal elements). We remark that this matrix storage is less than that of $ILUT(\tau, n_{fil})$ factorization applied to the global matrix A (n_{fil} indicates the nonzero elements per row in each of the L and U factors), who requires locations less than $2 \times n_{fil} \times n$.
In tables with numerical results, "iter." shows the number of $GMRES(20)$ iterations; "prec." shows the CPU time in seconds spent in constructing the preconditioners; "solu." shows the CPU time for the solution phase; "spar." shows the sparsity ratio which is the ratio between the number of nonzero elements of the preconditioner in its block ILU factorization to that of the original matrix. The numerical experiments were conducted on a Pentium III CPU at 550 MHZ with 64Mb RAM.
The solution details for the first problem (Poisson problem) are listed in Table 1 for $\tau = 10^{-4}$ and $l_{fil} = 10$. We note that for $M_{ILU(0)}$ preconditioner the convergence is slow. By comparing M_{ILUD} and M_{ILUT}, we can see that the number of iterations for M_{ILUD} is less than that M_{ILUT}, but this profit is to the detriment of the sparsity ratio.

Table 2. Iterations to convergence for the convection-diffusion problem preconditioned by $M_{ILUT}(10^{-4}, 10)$.

ν	1		10^{-1}		10^{-2}		10^{-3}		10^{-4}		10^{-5}	
h^{-1}	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)
17	5	5	5	5	6	4	6	4	8	4	8	9
33	7	7	8	8	9	6	10	5	10	5	12	13
65	10	10	11	11	13	9	14	6	14	6	15	10
105	13	13	14	14	17	12	19	8	20	6	20	6

Table 3. Iterations to convergence for the convection-diffusion problem preconditioned by $M_{ILUD}(10^{-4})$.

ν	1		10^{-1}		10^{-2}		10^{-3}		10^{-4}		10^{-5}	
h^{-1}	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)
17	3	3	3	3	3	3	4	3	5	4	6	12
33	4	4	4	4	4	4	5	4	5	4	7	8
65	4	4	4	4	5	5	5	4	6	4	8	9
105	5	5	5	5	6	5	6	5	7	5	9	4

Table 4. Iterations to convergence for the convection-diffusion problem preconditioned by $M_{ILU(0)}$.

ν	1		10^{-1}		10^{-2}		10^{-3}		10^{-4}		10^{-5}	
h^{-1}	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)
17	13	13	12	11	17	8	18	7	18	8	18	14
33	20	20	22	19	39	13	60	10	70	10	73	19
65	34	34	42	41	74	23	–	14	–	13	–	14
105	80	80	90	94	134	60	–	17	–	15	–	16

Tables 2, 3 and 4 list the number of preconditioned $GMRES(20)$ iterations for solving linear systems arising from convection-diffusion problem with $\bar{v}$ defined by $(P1)$ and $(P2)$. Note that the preconditioner M_{ILUD} yields better than M_{ILUT} and $M_{ILU(0)}$. It can be seen that for the preconditioner $M_{ILU(0)}$, the convergence is very slow and not reached for large problems (with small h), this due to the elements that are dropped since the Schur complement S is also sparse. However for M_{ILUD} preconditioner with $\tau = 10^{-6}$ and for the convection-diffusion problem $(P2)$ the number of iterations decrease to 7 for $h^{-1} = 17$ or 33 and to 3 for $h^{-1} = 65$.
Tables 5 and 6 show that M_{ILUT} needed less than half the storage required for M_{ILUD} to converge. Furthermore, the time "prec." necessary to construct the M_{ILUT} is less than that of M_{ILUD}.
In Table 7, we chose several values for τ and l_{fil} for $M_{ILUT}(\tau, l_{fil})$, one can remark that, we obtain fast convergence at reasonable sparsity ratio if we decrease the dropping tolerance and we increase the parameter l_{fil}. The sparsity ratio is in all cases less than 3.09 for $l_{fil} = 10$, and less 4.05 for $l_{fil} = 15$.

Concluding remarks. We have presented three block ILU preconditioners where the first block was inverted exactly. We have tested these preconditioners on matrices arising from block red-black coloring and five-point upwind finite difference scheme. Our numerical experiments show that to conserve grid and Reynolds number Re ($Re = \frac{1}{\nu}$) independent convergence, we are obliged to increase the parameter l_{fil} and to decrease the dropping tolerance τ but this profit is to the detriment of the sparsity ratio. We have showed that the Frobenius norm of a preconditioning step is directly related to the size of B^{-1}, R_S and $\tilde{S}^{-1}$. Hence a high quality of preconditioner must have a stable block ILU factorization.

Table 5. Comparison of $M_{ILUD}(10^{-4})$ and $M_{ILUT}(10^{-4}, 10)$ for solving the convection-diffusion problem with $\bar{v}$ is defined by $(P1)$.

		$M_{ILUD}(10^{-4})$				$M_{ILUT}(10^{-4}, 10)$			
h^{-1}	ν	iter.	prec.	solu.	spar.	iter.	prec.	solu.	spar.
65	10^{-1}	4	0.33	0.17	7.21	11	0.16	0.22	3.07
	10^{-3}	5	0.33	0.17	6.98	14	0.11	0.27	3.05
	10^{-5}	8	0.22	0.27	6.23	15	0.10	0.38	2.98
105	10^{-1}	5	1.54	0.49	8.20	14	0.39	0.88	3.08
	10^{-3}	6	0.98	0.61	7.97	19	0.39	1.15	3.06
	10^{-5}	9	0.82	0.77	7.12	20	0.38	1.21	3.00

Table 6. Comparison of $M_{ILUD}(10^{-5})$ and $M_{ILUT}(10^{-5}, 15)$ for solving the convection-diffusion problem with $\bar{v}$ is defined by $(P1)$.

		$M_{ILUD}(10^{-5})$				$M_{ILUT}(10^{-5}, 15)$			
h^{-1}	ν	iter.	prec.	solu.	spar.	iter.	prec.	solu.	spar.
65	10^{-1}	3	0.44	0.11	8.82	8	0.15	0.22	4.01
	10^{-3}	4	0.33	0.17	8.49	11	0.15	0.28	3.98
	10^{-5}	5	0.27	0.22	7.87	12	0.11	0.28	3.88
105	10^{-1}	4	1.64	0.50	11.66	10	0.55	0.66	4.05
	10^{-3}	5	1.48	0.60	11.14	14	0.55	0.99	4.02
	10^{-5}	6	1.32	0.66	10.14	15	0.49	1.04	3.91

Table 7. Performance of $M_{ILUT}(\tau, l_{fil})$ for solving the convection-

diffusion problem with $\bar{v}$ is defined by $(P2)$.

Parameters			$h^{-1}=65$				$h^{-1}=105$			
l_{fil}	τ	ν	iter.	prec.	solu.	spar.	iter.	prec.	solu.	spar.
10	10^{-3}	10^{-1}	11	0.11	0.22	3.01	14	0.33	0.88	3.01
		10^{-3}	6	0.11	0.11	2.50	8	0.27	0.50	2.58
		10^{-5}	11	0.10	0.22	2.15	6	0.22	0.33	2.17
15	10^{-3}	10^{-1}	8	0.17	0.22	3.75	10	0.39	0.71	3.80
		10^{-3}	5	0.11	0.11	2.79	6	0.27	0.33	2.91
		10^{-5}	5	0.06	0.11	2.32	5	0.22	0.33	2.36
10	10^{-4}	10^{-1}	11	0.16	0.33	3.07	14	0.38	0.87	3.08
		10^{-3}	6	0.06	0.16	2.71	8	0.28	0.49	2.78
		10^{-5}	10	0.05	0.22	2.23	6	0.22	0.33	2.27
15	10^{-4}	10^{-1}	8	0.17	0.22	3.95	10	0.88	0.66	3.98
		10^{-3}	4	0.16	0.11	3.16	5	0.33	0.27	3.27
		10^{-5}	4	0.05	0.11	2.55	4	0.22	0.27	2.61
10	10^{-5}	10^{-1}	11	0.11	0.27	3.08	14	0.44	0.88	3.09
		10^{-3}	6	0.11	0.17	2.83	8	0.28	0.44	2.88
		10^{-5}	10	0.05	0.16	2.34	6	0.22	0.33	2.38
15	10^{-5}	10^{-1}	8	0.17	0.21	4.02	10	0.50	0.71	4.05
		10^{-3}	4	0.11	0.16	3.38	5	0.33	0.32	3.50
		10^{-5}	9	0.06	0.17	2.74	4	0.22	0.22	2.80

References

[1] O. Axelsson , *Iterative solution methods*, University Press, Cambridge, 1994.

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

[3] M. Benzi and M. Tuma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput. 19(1998) 968-994.

[4] C. Brand and Z. E. Heinemann, *A new iterative solution technique for reservoir simulation equations on locally refined grids*, SPE, (18410), 1989.

[5] T.F. Chan , *Analysis of preconditioners for domain decomposition*, SIAM J. Numer. 24(2) (1987) 382-390.

[6] T.F. Chan and D. Goovaerts, *A note on the efficiency of domain decomposed incomplete factorization*, SIAM J. Sci. Stat. Comput. 11(4) (1990) 794-803.

[7] T.F. Chan and H. A. van der Vorst , *Approximate and incomplete factorization*, Technical Report 871, Department of Mathematics, University of utrecht 1994.

[8] E. Chow and Y. Saad, *Approximate inverse preconditioner via sparse-sparse iterations*, SIAM J. Comput. Vol. 19 (1998) No 3 995-1023.

[9] E. Chow and Y. Saad, *Approximate inverse techniques for block-partitioned matrices*, SIAM J. Sci. Comput. 18 (1997) 1657-1675.

[10] P. Ciarlet Jr., *Repeated red-black orderings: a new approach*, Numer. Alg., 1994.

[11] E. F. D'Azevedo, P. A. Forsyth and Wei-Pai Tang, *Orderings methods preconditioned conjugate gradient methods applied to unstructured grid problems*, SIAM J. Matrix Anal. Appl. 13(3) (1992) 944-961.

[12] I. S. Duff and G. A. Meurant, *The effect of orderings on preconditioned conjugate gradients*, BIT 29 (1989) 635-657.

[13] H. C. Elman, *A stability analysis of incomplete LU factorization*, Math. Comp. 47(175) (1986) 191-217.

[14] N. I. M. Gould and J. A. Scott, *Sparse approximate-inverse preconditioners using normalization techniques*, SIAM J. Sci. Comput. 19 (1998) 605-625.

[15] M. G. Grote and T. Huckle, *Parrallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput. 18 (1997) 838-853.

[16] N. Guessous and O. Souhar, *Multilevel block ILU preconditioner for sparse nonsymmetric M-matrices*, J. Comput. Appl. Math. 162 (2004) 231-246.

[17] M. T. Jones and P. E. Plassman, *A parallel graph coloring heuristic*, SIAM J. Sci. Comput. 14 (1993) 654-669.

[18] L. YU. Kolotilina, *Explicite preconditioning of H-matries, in Numerical Analysis and Mathematical Modelling*, YU. A. Kuznetsov, ed. , Dept. of Numerical Mathematics, USSR Academy of sciences, Moscow, 1989 97-108, (in Russia).

[19] L. YU. Kolotilina and A. Yu. Yermin , *Factorized sparse approximate inverse preconditioning*, SIAM J. Matrix-Anal. 14 (1993) 45-58.

[20] T. A Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Mathematics of computation 32 (1980) 473-497.

[21] G. Meurant, *Domain decomposition methods for partial differential equations on parallel computers*, Int. J. Supercomputing Appls. 2 (1988) 5-12.

[22] G. Meurant, *The block preconditioned conjugate gradient method on vector computers*, BIT 24 (1984) 623-633.

[23] Y. Notay, *A robust algebraic multilevel preconditioner for nonsymmetric M-matrix*, Tech. Rep. GANMN 99-01, Université Libre de Bruxelles, Brussels, Belgium, 1999.

[24] Y. Saad, *ILUT: A dual threshold incomplete LU preconditioner*, Numer. Linear Algebra Appl. 1(4) (1994) 387-402.

[25] Y. SAAD, *ILUM: A multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput. 17(4) (1996) 830-847.

[26] Y. SAAD, *Iterative methods for sparse linear systems*, PWS Publishing, New York, NY, 1996.

[27] Y. SAAD, *SPARSKIT: A basic tool kit for sparse matrix computations*, Technical Report CSRD TR 1029, University of Illinois at Urbana-Champaign, IL, 1990.

[28] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymetric linear systems*, SIAM J. Sci. Statist. Comput. 7(1986) 856-869.

[29] Y. SAAD AND J. ZHANG, *BILUM: Block versions of multi-elimination and multilevel ILU preconditioner for general linear sparse systems*, SIAM J. Sci. Comput. 20(6) (1999) 2103-2121.

[30] Y. SAAD AND J. ZHANG, *BILUTM: A domain based multilevel block ILUT preconditioner for general sparse matrices*, SIAM J. Matrix Anal. Appl. 21(1) (1999) 279-299.

[31] Y. SAAD AND J. ZHANG, *Diagonal threshold techniques in robust multi-level ILU preconditioner for general linear sparse systems*, Numer. Linear Algebra Appl. 6 (1999) 257-280.

[32] J. ZHANG, *A sparse approximate inverse technique for parallel preconditioning of general sparse matrices*, Technical Report 281-98, Department of Computer science, University of Kentucky, Lexington, KY, 1998.

[33] J. ZHANG, *On preconditioning Schur complement and Schur complement preconditioning*, ETNA 10 (2000) 115-130.

[34] J. ZHANG, *Preconditioned iterative methods and finite differences schemes for convection-diffusion*, Appl. Math. and Comput. 109 (2000) 11-30.