SIAM J. SCI. COMPUT.
Vol. 27, No. 5, pp. 1627–1650

MULTILEVEL PRECONDITIONERS CONSTRUCTED FROM INVERSE-BASED ILUs[*]

MATTHIAS BOLLHÖFER[†] AND YOUSEF SAAD[‡]

Abstract. This paper analyzes dropping strategies in a multilevel incomplete LU decomposition context and presents a few strategies for obtaining related ILUs with enhanced robustness. The analysis shows that the incomplete LU factorization resulting from dropping small entries in Gaussian elimination produces a good preconditioner when the inverses of these factors have norms that are not too large. As a consequence a few strategies are developed whose goal is to achieve this feature. A number of "templates" for enabling implementations of these factorizations are presented. Numerical experiments show that the resulting ILUs offer a good compromise between robustness and efficiency.

Key words. incomplete LU decompositions, ILU, preconditioning, multilevel ILU, approximate inverse, algebraic multilevel method, iterative solver

AMS subject classifications. 65F05, 65F10, 65F50, 65Y05

DOI. 10.1137/040608374

1. Introduction.

In recent years, research on "black-box" techniques for solving sparse linear systems of equations has accelerated due to two factors. First, linear systems are becoming ever more difficult to solve due to their bigger sizes. In spite of recent progress in parallel direct solvers [8, 21, 13], practitioners are looking at cheaper alternatives. The second factor is that preconditioning methods have also made some tangible progress in improving robustness. Iterative methods based on preconditioned Krylov subspace techniques are increasingly viewed as attractive alternatives to direct solvers, both for structured problems, such as those arising from discretized partial differential equations (PDEs), and for systems arising from applications areas which yield highly unstructured systems and for which iterative methods were viewed as ineffective in the past.

One essential improvement for the use of Krylov subspace solvers consists of preconditioning methods [18], in particular methods based on incomplete LU decompositions [17]. One of the first attempts at using Krylov subspace methods as a general purpose black-box solver was by Simon [22], who used standard (symmetric) reordering techniques from direct solution methods to preprocess matrices prior to applying level-of-fill ILU. More recent work which employs nonsymmetric reordering strategies [10, 2] indicates that systems arising from a wide range of applications can be successfully solved by this approach in conjunction with ILUs.

Aside from this progress, there still remains a drawback of incomplete LU decompositions, which is that they are sensitive to parameters such as drop tolerances or certain static reordering strategies. Progress has been made in improving the stability of ILUs by taking into account the inverse triangular factors [5, 3, 14]. Also,

[*] Received by the editors May 15, 2004; accepted for publication (in revised form) January 10, 2005; published electronically February 3, 2006.
http://www.siam.org/journals/sisc/27-5/60837.html

[†] Department of Mathematics, MA 4-5, TU Berlin, D-10623 Berlin, Germany (bolle@math.tu-berlin.de, http://www.math.tu-berlin.de/∼bolle/). The work of this author was supported by DFG research center MATHEON "Mathematics for Key Technologies" in Berlin.

[‡] Department of Computer Science and Engineering, University of Minnesota, 200 Union Street S.E., Minneapolis, MN 55455 (saad@cs.umn.edu, http://www.cs.umn.edu/∼saad/). This author's work was supported by National Science Foundation grants NSF/ACI-0305120 and NSF/INT-0003274 and by the Minnesota Supercomputer Institute.

combining incomplete LU decompositions in a hierarchical fashion with preordering pivoting strategies [19] helps in improving stability of the preconditioner.

In this paper we will discuss the rationale for a new multilevel strategy which mainly focuses on keeping the inverse triangular factors bounded. This is justified by an analysis that indicates that stability can be enhanced by this approach.

2. Impact of dropping on the preconditioned system.

This section analyzes how a perturbation introduced by dropping small terms while computing incomplete LU decompositions affects the preconditioned system. This analysis will help design incomplete LU factorization preconditioners with improved robustness.

2.1. The inverse error for a single level.

We start with a partial factorization of our matrix that is terminated after k steps. We assume that the initial $n \times n$ nonsingular matrix A is rearranged as

$$P^\top AQ = \begin{pmatrix} B & F \\ E & C \end{pmatrix},$$

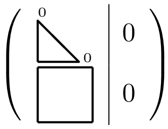
where the leading matrix B is nonsingular and of size $k \times k$ and where P and Q are suitably chosen permutation matrices. Suppose that this matrix is approximately factored as

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} \tilde{L}_B & 0 \\ \tilde{L}_E & I \end{pmatrix} \begin{pmatrix} \tilde{D}_B & 0 \\ 0 & \tilde{S} \end{pmatrix} \begin{pmatrix} \tilde{U}_B & \tilde{U}_F \\ 0 & I \end{pmatrix} + \mathcal{E}_k \equiv \tilde{\mathcal{L}}_k \tilde{\mathcal{D}}_k \tilde{\mathcal{U}}_k + \mathcal{E}_k. \tag{2.1}$$

Here, $\tilde{L}_B$ and $\tilde{U}_B^\top$ are unit lower triangular and $\tilde{D}_B$ is diagonal. The matrix $\tilde{S}$ is an approximation to the Schur complement $S = C - EB^{-1}F$. Assume for simplicity that we do not encounter zero pivots. We note that in principle one could generalize the following analysis using pivoting. To keep the analysis simple we assume that no pivoting is necessary for this partial incomplete factorization.

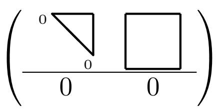
At any given step $l = 1, \ldots, k$, entries of $\tilde{\mathcal{L}}_l$ and $\tilde{\mathcal{U}}_l^\top$ may be dropped in some positions (m, l), where $m > l$. We represent these dropped values by an $n \times n$ matrix

$$\mathcal{V}_l = \left(\begin{array}{ccc|ccc} v_1 & \cdots & v_l & 0 & \cdots & 0 \end{array} \right) = \tag{2.2}$$



for $\tilde{\mathcal{L}}_l$ and

$$\mathcal{W}_l = \left(\begin{array}{c} w_1^\top \\ \vdots \\ w_l^\top \\ \hline 0 \\ \vdots \\ 0 \end{array} \right) = \tag{2.3}$$



for $\tilde{\mathcal{U}}_l$. The elimination process adds one row/column at every step l.

Depending on how the approximate Schur complement $\tilde{S}$ is defined, a different error matrix $\mathcal{E}_k$ will be obtained. We will only distinguish between two choices.

S-*version*: This corresponds to the "simple" approximate Schur complement defined by

$$\tilde{S} = C - \tilde{L}_E \tilde{D}_B \tilde{U}_F. \tag{2.4}$$

T-*version*: This corresponds to the more expensive Schur complement proposed in [23] and defined by

$$\tilde{T} = \begin{pmatrix} -\tilde{L}_E \tilde{L}_B^{-1} & I \end{pmatrix} P^\top AQ \begin{pmatrix} -\tilde{U}_B^{-1}\tilde{U}_F \\ I \end{pmatrix}. \tag{2.5}$$

Note that the T-version of the Schur complement results from applying the inverse factors $\tilde{\mathcal{L}}_k^{-1}$ to the left and $\tilde{\mathcal{U}}_k^{-1}$ to the right to $P^\top AQ$ and taking the lower right block. Obtaining $\tilde{T}$ does not really require inverting $\tilde{L}_B$ and $\tilde{U}_B$. It can be computed by updating $\tilde{T}$ by a low-rank matrix at each step $l = 1, \ldots, k$.

The next result shows how the error matrix $\mathcal{E}_k$ can be characterized for the two choices of the approximate Schur complement.

LEMMA 2.1. *Using the above notation we obtain for the S-version* (2.4)

$$\mathcal{E}_k = \mathcal{V}_k \tilde{\mathcal{D}}_k + \tilde{\mathcal{D}}_k \mathcal{W}_k, \tag{2.6}$$

and for the T-version (2.5)

$$\mathcal{E}_k = \mathcal{V}_k \tilde{\mathcal{D}}_k \tilde{\mathcal{U}}_k + \tilde{\mathcal{L}}_k \tilde{\mathcal{D}}_k \mathcal{W}_k. \tag{2.7}$$

Proof. We will show this by induction. Initially at step $l = 0$ there is no error present and we set $\mathcal{V}_0 = \mathcal{W}_0 = 0$ and $\tilde{\mathcal{L}}_0 = \tilde{\mathcal{U}}_0 = I$.

Now going from step $l-1$ to step l $(l \leqslant k)$ we have

$$P^\top AQ = \tilde{\mathcal{L}}_{l-1} \tilde{\mathcal{D}}_{l-1} \tilde{\mathcal{U}}_{l-1} + \mathcal{E}_{l-1}.$$

To distinguish between step $l-1$ and l we add a subscript l to all matrices. We partition the approximate Schur complement $\tilde{S}_{l-1}$ from $\tilde{\mathcal{D}}_{l-1}$ as

$$\tilde{S}_{l-1} = \begin{pmatrix} \beta & f^\top \\ e & \hat{C} \end{pmatrix}.$$

At step l we get

$$\tilde{\mathcal{L}}_l = \tilde{\mathcal{L}}_{l-1} \left(I + [y - v]\, e_l^\top \right),$$

where

$$y = \begin{pmatrix} 0 \\ 0 \\ \beta^{-1} e \end{pmatrix} \begin{array}{l} \leftarrow \text{size} = l-1, \\ \leftarrow \text{size} = 1, \\ \leftarrow \text{size} = n-l \end{array}$$

is the leading column of the approximate Schur complement divided by the diagonal entry and

$$v = \begin{pmatrix} 0 \\ 0 \\ \beta^{-1} \varepsilon_e \end{pmatrix}$$

denotes the vector of those entries from y that are dropped at step l. For the U-part, we similarly obtain

$$\tilde{\mathcal{U}}_l = (I + e_l(z - w)^\top)\tilde{\mathcal{U}}_{l-1},$$

where

$$z^\top = \begin{pmatrix} 0 & 0 & f^\top/\beta \end{pmatrix}$$

and

$$w^\top = \begin{pmatrix} 0 & 0 & \varepsilon_f^\top/\beta \end{pmatrix}$$

denotes the vector of those entries from z being dropped.

We now consider the S-version and T-version separately, beginning with the S-version. For the S-version the next approximate Schur complement $\tilde{S}_l$ is set as

$$\tilde{S}_l = \hat{C} - (e - \varepsilon_e)\beta^{-1}(f - \varepsilon_f)^\top.$$

It follows that

$$\begin{aligned}
&\tilde{\mathcal{L}}_l\tilde{\mathcal{D}}_l\tilde{\mathcal{U}}_l \\
&= \begin{pmatrix} L_{B,l-1} & 0 \\ L_{E,l-1} & \begin{pmatrix} 1 & 0 \\ \frac{e-\varepsilon_e}{\beta} & I \end{pmatrix} \end{pmatrix} \begin{pmatrix} D_{B,l-1} & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \tilde{S}_l \end{pmatrix} \begin{pmatrix} U_{B,l-1} & U_{F,l-1} \\ 0 & \begin{pmatrix} 1 & \frac{(f-\varepsilon_f)^\top}{\beta} \\ 0 & I \end{pmatrix} \end{pmatrix} \\
&= \begin{pmatrix} L_{B,l-1} & 0 \\ L_{E,l-1} & I \end{pmatrix} \begin{pmatrix} D_{B,l-1} & 0 & 0 \\ 0 & \beta & (f - \varepsilon_f)^\top \\ 0 & e - \varepsilon_e & \hat{C} \end{pmatrix} \begin{pmatrix} U_{B,l-1} & U_{F,l-1} \\ 0 & I \end{pmatrix} \\
&= \tilde{\mathcal{L}}_{l-1}\tilde{\mathcal{D}}_{l-1}\tilde{\mathcal{U}}_{l-1} - v\beta e_l^\top - e_l\beta w^\top.
\end{aligned}$$

By induction we already have

$$\tilde{\mathcal{L}}_{l-1}\tilde{\mathcal{D}}_{l-1}\tilde{\mathcal{U}}_{l-1} = P^\top AQ - \mathcal{E}_{l-1} = P^\top AQ - \mathcal{V}_{l-1}\tilde{\mathcal{D}}_{l-1} - \tilde{\mathcal{D}}_{l-1}\mathcal{W}_{l-1},$$

and by definition we have $v = \mathcal{V}_l e_l$ and $w^\top = e_l^\top \mathcal{W}_l$. The leading $l \times l$ part of $\tilde{\mathcal{D}}_l$ is diagonal and the (l, l) entry is just β. Thus we obtain

$$\tilde{\mathcal{L}}_l\tilde{\mathcal{D}}_l\tilde{\mathcal{U}}_l = P^\top AQ - \mathcal{V}_{l-1}\tilde{\mathcal{D}}_{l-1} - \tilde{\mathcal{D}}_{l-1}\mathcal{W}_{l-1} - v\beta e_l^\top - e_l\beta w^\top = P^\top AQ - \mathcal{V}_l\tilde{\mathcal{D}}_l - \tilde{\mathcal{D}}_l\mathcal{W}_l,$$

which shows (2.6).

We now consider the T-version. In contrast with the S-version, for the T-version we get for $\tilde{T}_l$

$$\tilde{T}_l = \left(-(e - \varepsilon_e)\frac{1}{\beta},\ I\right)\tilde{S}_{l-1}\begin{pmatrix} -(f - \varepsilon_f)^\top\frac{1}{\beta} \\ I \end{pmatrix} = \hat{C} - e\frac{1}{\beta}f^\top + \varepsilon_e\frac{1}{\beta}\varepsilon_f^\top.$$

From this it follows that

$$\begin{aligned}
\tilde{\mathcal{L}}_l^{-1}P^\top AQ\tilde{\mathcal{U}}_l^{-1} &= \tilde{\mathcal{L}}_l^{-1}\left(\tilde{\mathcal{L}}_{l-1}\tilde{\mathcal{D}}_{l-1}\tilde{\mathcal{U}}_{l-1} + \mathcal{E}_{l-1}\right)\tilde{\mathcal{U}}_l^{-1} \\
&= (I - (y - v)e_l^\top)\tilde{\mathcal{D}}_{l-1}(I - e_l(z - w)^\top) \\
&\quad + \tilde{\mathcal{L}}_l^{-1}\mathcal{V}_{l-1}\tilde{\mathcal{D}}_{l-1}(I - e_l(z - w)^\top) \\
&\quad + (I - (y - v)e_l^\top)\tilde{\mathcal{D}}_{l-1}\mathcal{W}_{l-1}\tilde{\mathcal{U}}_l^{-1} \\
&= \tilde{\mathcal{D}}_l + v\beta e_l^\top + e_l\beta w^\top \\
&\quad + \tilde{\mathcal{L}}_l^{-1}\mathcal{V}_{l-1}\tilde{\mathcal{D}}_{l-1} \\
&\quad + \tilde{\mathcal{D}}_{l-1}\mathcal{W}_{l-1}\tilde{\mathcal{U}}_l^{-1}.
\end{aligned}$$

For the last equation we used the definition of $\tilde{\mathcal{D}}_l$ and, in particular, that of $\tilde{T}_l$. Also note that only the leading $l-1$ columns of $\mathcal{V}_{l-1}\tilde{\mathcal{D}}_{l-1}$ are nonzero (analogously, $e_l^\top \tilde{\mathcal{D}}_{l-1}\mathcal{W}_{l-1} = 0$).

It is easy to verify that $v\beta e_l^\top = \tilde{\mathcal{L}}_l^{-1} v\beta e_l^\top$ since the leading l entries of v are zero. A similar argument can be made for $e_l\beta w^\top$. Altogether we have

$$\tilde{\mathcal{L}}_l^{-1} P^\top A Q \tilde{\mathcal{U}}_l^{-1} = \tilde{\mathcal{D}}_l + \tilde{\mathcal{L}}_l^{-1}\mathcal{V}_l\tilde{\mathcal{D}}_l + \tilde{\mathcal{D}}_l\mathcal{W}_l\tilde{\mathcal{U}}_l^{-1},$$

since $v\beta e_l^\top$ and $e_l\beta w^\top$ are the l column/row of $\mathcal{V}_l\tilde{\mathcal{D}}_l$ and $\tilde{\mathcal{D}}_l\mathcal{W}_l$. This completes the proof. □

A corollary of the representation of the error is obtained when we consider the inverse error

$$\mathcal{F}_k = \tilde{\mathcal{L}}_k^{-1}\mathcal{E}_k\tilde{\mathcal{U}}_k^{-1}. \tag{2.8}$$

Corollary 2.2. *Under the assumptions of Lemma* 2.1 *we have for the S-version*

$$\mathcal{F}_k = \tilde{\mathcal{L}}_k^{-1}\mathcal{V}_k\tilde{\mathcal{D}}_k\tilde{\mathcal{U}}_k^{-1} + \tilde{\mathcal{L}}_k^{-1}\tilde{\mathcal{D}}_k\mathcal{W}_k\tilde{\mathcal{U}}_k^{-1}, \tag{2.9}$$

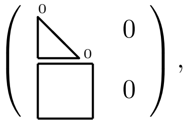
and for the T-version we obtain

$$\mathcal{F}_k = \tilde{\mathcal{L}}_k^{-1}\mathcal{V}_k\tilde{\mathcal{D}}_k + \tilde{\mathcal{D}}_k\mathcal{W}_k\tilde{\mathcal{U}}_k^{-1}. \tag{2.10}$$

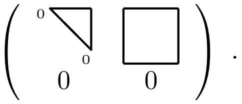
An important consequence of Corollary 2.2 is that the inverse triangular factors $\tilde{\mathcal{L}}_k^{-1}$ and $\tilde{\mathcal{U}}_k^{-1}$ amplify the size of the entries being dropped during the incomplete LU decomposition. For the S-version this impact is likely to be stronger since both inverse factors contribute to the error at the same time.

An interesting observation can be made when taking a detailed look at the patterns of the inverse errors. To do this, partition $\mathcal{V}_k$ and $\mathcal{W}_k$ as

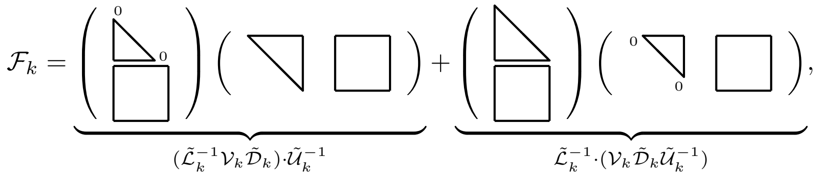
$$\mathcal{V}_k = \begin{pmatrix} V_B & 0 \\ V_E & 0 \end{pmatrix} \equiv \tag{2.11}$$



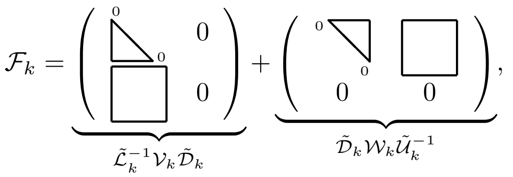
$$\mathcal{W}_k = \begin{pmatrix} W_B & W_F \\ 0 & 0 \end{pmatrix} \equiv \tag{2.12}$$



Then the inverse error of the S-version can be sketched with a pattern


$\mathcal{F}_k =$
0
0
0
0
$(\tilde{\mathcal{L}}_k^{-1}\mathcal{V}_k\tilde{\mathcal{D}}_k)\cdot\tilde{\mathcal{U}}_k^{-1}$
$\tilde{\mathcal{L}}_k^{-1}\cdot(\mathcal{V}_k\tilde{\mathcal{D}}_k\tilde{\mathcal{U}}_k^{-1})$

which shows that the error produced by dropping in the lower triangular part also contributes to the upper triangular part and vice versa. However, this is different for the T-version. Here we have


$\mathcal{F}_k =$
0
0
0
0
0
0
0
0
$\tilde{\mathcal{L}}_k^{-1}\mathcal{V}_k\tilde{\mathcal{D}}_k$
$\tilde{\mathcal{D}}_k\mathcal{W}_k\tilde{\mathcal{U}}_k^{-1}$

showing that dropping in the lower/upper triangular part produces errors in these parts only, and by the definition of the approximate Schur complement there clearly will be no perturbation in the lower right block. Note also that for the T-version, there are no errors in the (2,2) block, which is not surprising since the Schur complement is computed exactly from applying the inverses of the (inexact) factors $\mathcal{L}_k$ and $\mathcal{U}_k$ to $P^\top AQ$.

Let $\tilde{d}_1, \dots, \tilde{d}_k$ be the leading k diagonal entries of $\tilde{\mathcal{D}}_k$. If we keep in mind that the entries being dropped at step l are the entries in column l of $\mathcal{V}_l$ and $\mathcal{W}_l^\top$, then we can see that for the S-version we have

$$\mathcal{F}_k = \sum_{l \leqslant k} \left(\tilde{\mathcal{L}}_k^{-1} v_l \tilde{d}_l e_l^\top \tilde{\mathcal{U}}_k^{-1} + \tilde{\mathcal{L}}_k^{-1} e_l \tilde{d}_l w_l^\top \tilde{\mathcal{U}}_k^{-1} \right) \tag{2.13}$$

and, analogously for the T-version,

$$\mathcal{F}_k = \sum_{l \leqslant k} \left(\tilde{\mathcal{L}}_k^{-1} v_l \tilde{d}_l e_l^\top + e_l \tilde{d}_l w_l^\top \tilde{\mathcal{U}}_k^{-1} \right). \tag{2.14}$$

Equation (2.13) implies that the perturbation introduced at step l of the incomplete LU decomposition is given by

$$\tilde{\mathcal{L}}_k^{-1} v_l \tilde{d}_l e_l^\top \tilde{\mathcal{U}}_k^{-1}$$

from the L-part and

$$\tilde{\mathcal{L}}_k^{-1} e_l \tilde{d}_l w_l^\top \tilde{\mathcal{U}}_k^{-1}$$

from the U-part. For the T-version one can skip one of the inverse factors, as can be seen from (2.14). This indicates that it is wise to incorporate the norm $\|\tilde{\mathcal{L}}_k^{-1}\|$ into the dropping strategy at step l when entries of $\mathcal{L}_l$ are being dropped. In principle one should also include the norm $\|\tilde{\mathcal{U}}_k^{-1}\|$ in the S-version case when dropping in $\mathcal{L}_k$. Similar arguments apply when dropping in $\mathcal{U}_k$.

Unless we control the growth of $\tilde{\mathcal{L}}_k^{-1}$ and $\tilde{\mathcal{U}}_k^{-1}$ and keep their norm below a constant κ, the error introduced by dropping at step l can be arbitrarily amplified by the inverse factors.

COROLLARY 2.3. *Assume that* $\|L_l^{-1}\| \leqslant \kappa$ *and* $\|U_l^{-1}\| \leqslant \kappa$ *for steps* $l = 1, \dots, k$, *where* $\kappa > 0$ *is a prescribed bound. Denote by* $\tilde{d}_l$ *the leading diagonal entry of the approximate Schur complement at step* l.

1. *Let* $\tilde{s}_{lj}$ *and* $\tilde{s}_{il}$ *for* $i, j > l$ *be the entries in the leading column and row of the approximate Schur complement at step* l. *Suppose that the S-version is used and entries* $\tilde{s}_{lj}$, $\tilde{s}_{il}$ *are dropped only if*

$$\kappa^2 |\tilde{s}_{lj}| \leqslant \varepsilon |\tilde{d}_l|, \ \kappa^2 |\tilde{s}_{il}| \leqslant \varepsilon |\tilde{d}_l|;$$

then

$$\mathcal{F}_k = \sum_{l=1}^{k} \mathcal{F}_{L,l} + \sum_{l=1}^{k} \mathcal{F}_{U,l},$$

where $F_{L,l}$ *and* $F_{U,l}$ *are rank*-1 *matrices such that any entry is bounded by* $\varepsilon |\tilde{d}_l|$.

2. *Let $\tilde{t}_{lj}$ and $\tilde{t}_{il}$ for $i,j>l$ be the entries in the leading column and row of the approximate Schur complement at step l. In case of the T-version assume that for $i,j>l$, $\tilde{t}_{lj}$ and $\tilde{t}_{il}$ are dropped only if*

$$\kappa|\tilde{t}_{lj}| \leqslant \varepsilon|\tilde{d}_l|,\ \kappa|\tilde{t}_{il}| \leqslant \varepsilon|\tilde{d}_l|;$$

then

$$\mathcal{F}_k = \mathcal{F}_{L,k}\tilde{\mathcal{D}}_k + \tilde{\mathcal{D}}_k\mathcal{F}_{U,k},$$

where

$$\max_{i,j} e_i^\top|\mathcal{F}_{L,k}|e_j \leqslant \varepsilon, \qquad \max_{i,j} e_i^\top|\mathcal{F}_{U,k}|e_j \leqslant \varepsilon.$$

Proof. The results follow immediately from the assumptions and (2.13) for the S-version case and (2.14) for the T-version case. ∎

2.2. The multilevel framework. Multilevel ILU procedures do not usually proceed until $k=n$; stop at a certain step k and postpone factoring the remaining part to the next level. Thus, one can write at a given level η

$$P^\top A_\eta Q = \begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} \tilde{L}_B & 0 \\ \tilde{L}_E & I \end{pmatrix}\begin{pmatrix} \tilde{D}_B & 0 \\ 0 & \tilde{S} \end{pmatrix}\begin{pmatrix} \tilde{U}_B & \tilde{U}_F \\ 0 & I \end{pmatrix} + \mathcal{E}_k \to A_{\eta+1} = \tilde{S}.$$

Postponing the factorization of $A_{\eta+1} \equiv S$ may be mandatory because the prescribed bound may otherwise be exceeded. When considering the next level matrix $A_{\eta+1}$, specific techniques, such are nonsymmetric permutations, can be invoked to ensure that the factorization at this level is more accurate and reliable. The simple version of the Schur complement is often preferred since it leads to less fill-in than the T-version. However, the stability analysis seen above has shown that this version is much more sensitive to perturbations. On the other hand, the T-version gains from its higher robustness, but it will suffer from the fact that it is more expensive to compute and it may have significantly more fill-in.

A natural way of combining the S-version and the T-version in a multilevel framework could be to approximate B only using the S-version. This could still give a good approximation as long as B is, say, close to being diagonal dominant. But for the next level which requires the remaining Schur complement, one could switch to the T-version. This leads to the M-version.

M-*version*: Define the approximate Schur complement via

$$\tilde{M} = \begin{pmatrix} -\tilde{L}_E\tilde{L}_B^{-1} & I \end{pmatrix} P^\top AQ \begin{pmatrix} -\tilde{U}_B^{-1}\tilde{U}_F \\ I \end{pmatrix} \tag{2.15}$$

at step k and via the S-version prior to step k.

The motivation for this variant is that in a multilevel approach, it sometimes is possible to ensure that the factors of the B block are nicely bounded. In this situation it pays to compute the Schur complement more accurately, and the T-version may be ideal for this.

COROLLARY 2.4. *Suppose that the partial incomplete LU decomposition is computed via the M-version, i.e., from step $l=1,\dots,k$ the S-version is used, but at step k the remaining approximate Schur complement is defined via the T-version. Set*

$$\Pi = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}.$$

Then the inverse error satisfies

$$\begin{aligned}\mathcal{F}_k = &\left(\tilde{\mathcal{L}}_k^{-1}\mathcal{V}_k\tilde{\mathcal{D}}_k\tilde{\mathcal{U}}_k^{-1} + \tilde{\mathcal{L}}_k^{-1}\tilde{\mathcal{D}}_k\mathcal{W}_k\tilde{\mathcal{U}}_k^{-1}\right)\\ &-\Pi\left(\tilde{\mathcal{L}}_k^{-1}\mathcal{V}_k\tilde{\mathcal{D}}_k\tilde{\mathcal{U}}_k^{-1} + \tilde{\mathcal{L}}_k^{-1}\tilde{\mathcal{D}}_k\mathcal{W}_k\tilde{\mathcal{U}}_k^{-1}\right)\Pi.\end{aligned} \tag{2.16}$$

Proof. Since we compute until step k the incomplete LU decomposition via the S-version, we will have until that step

$$\mathcal{E}_k = \mathcal{V}_k\tilde{\mathcal{D}}_k + \tilde{\mathcal{D}}_k\mathcal{W}_k.$$

As we now switch to the T-version, it means that we precisely have

$$\tilde{M} = \left(\begin{array}{cc} 0 & I \end{array}\right)\tilde{\mathcal{L}}_k^{-1}(P^\top AQ)\tilde{\mathcal{U}}_k^{-1}\left(\begin{array}{c} 0 \\ I \end{array}\right) \Rightarrow \left(\begin{array}{cc} 0 & I \end{array}\right)\mathcal{F}_k\left(\begin{array}{c} 0 \\ I \end{array}\right) = 0.$$

This means that there will be no error in the lower right block. □

The inverse error that has been computed so far is based on the assumption that the approximate factorization from (2.1) is used with either the S-version or the T-version as approximate Schur complement. Since we would like to use this ILU in a multilevel context, we do not have to keep the factors $\tilde{L}_E$ and $\tilde{U}_F$. Instead we can substitute them with $E\tilde{U}_B^{-1}\tilde{D}_B^{-1}$ and $\tilde{D}_B^{-1}\tilde{L}_B^{-1}F$. Keeping in mind that we will not compute these factors explicitly but only solve systems with them, this leads to the approximate factorization

$$\left(\begin{array}{cc} B & F \\ E & C \end{array}\right) = \underbrace{\left(\begin{array}{cc} \tilde{L}_B & 0 \\ E\tilde{U}_B^{-1}\tilde{D}_B^{-1} & I \end{array}\right)}_{\tilde{L}_k}\underbrace{\left(\begin{array}{cc} \tilde{D}_B & 0 \\ 0 & \tilde{S} \end{array}\right)}_{\tilde{D}_k}\underbrace{\left(\begin{array}{cc} \tilde{U}_B & \tilde{D}_B^{-1}\tilde{L}_B^{-1}F \\ 0 & I \end{array}\right)}_{\tilde{U}_k} + \mathcal{E}_k. \tag{2.17}$$

In this case one can easily see that the error and the inverse error only show up in the B-part and the C-part.

Theorem 2.5. *Suppose that we use the partial incomplete LU decomposition from* (2.17). *Let* $\mathcal{V}_k$ *and* $\mathcal{W}_k$ *be partitioned as in* (2.11)–(2.12). *Then the following holds for the inverse error.*

1. *In the case of the S-version we obtain*

$$\mathcal{F}_k = \tilde{\mathcal{L}}_k^{-1}\left(\begin{array}{cc} V_B\tilde{D}_B + \tilde{D}_BW_B & 0 \\ 0 & 0 \end{array}\right)\tilde{\mathcal{U}}_k^{-1}.$$

2. *For the M-version we obtain*

$$\begin{aligned}\mathcal{F}_k = &\quad\tilde{\mathcal{L}}_k^{-1}\left(\begin{array}{cc} V_B\tilde{D}_B + \tilde{D}_BW_B & 0 \\ 0 & 0 \end{array}\right)\tilde{\mathcal{U}}_k^{-1}\\ &-\Pi\tilde{\mathcal{L}}_k^{-1}\left(\begin{array}{cc} V_B\tilde{D}_B + \tilde{D}_BW_B & 0 \\ 0 & 0 \end{array}\right)\tilde{\mathcal{U}}_k^{-1}\Pi,\end{aligned}$$

where Π *is chosen as in Corollary* 2.4.

Proof. From (2.17) one can immediately see that

$$\mathcal{E}_k = \left(\begin{array}{cc} * & 0 \\ 0 & * \end{array}\right).$$

Then the assertion follows immediately from Corollaries 2.2 and 2.4. □

Remark. In principle, for the T-version the same approach leads to analogous bounds which are more complicated. But one essential property will be lost: Entries dropped in the lower triangular part would only contribute to the perturbation in this part (similar for the upper triangular part).

2.3. The inverse error contribution from lower levels. In the multilevel context the best we can achieve is to find permutation matrices P_C and Q_C for the approximate Schur complement $\tilde{S}$ such that

$$\left.\begin{array}{c} P_C^\top \tilde{S} Q_C \\ P_C^\top \tilde{M} Q_C \\ P_C^\top \tilde{T} Q_C \end{array}\right\} = L_C D_C U_C + \mathcal{E}_C,$$

where D_C is diagonal and $\|L_C^{-1}\|, \|U_C^{-1}\| \leqslant \kappa$ with a reasonable small inverse error $\mathcal{F}_C = L_C^{-1}\mathcal{E}_C U_C^{-1}$. In this case we finally end up with an approximate factorization:

$$\begin{aligned} &\begin{pmatrix} I & 0 \\ 0 & L_C^{-1}P_C^\top \end{pmatrix} \mathcal{L}_k^{-1} P^\top A Q \mathcal{U}_k^{-1} \begin{pmatrix} I & 0 \\ 0 & Q_C U_C^{-1} \end{pmatrix} \\ &= \begin{pmatrix} D_B & 0 \\ 0 & D_C \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & L_C^{-1}P_C^\top \end{pmatrix} \mathcal{F}_k \begin{pmatrix} I & 0 \\ 0 & Q_C U_C^{-1} \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & \mathcal{F}_C \end{pmatrix}. \end{aligned}$$

This result indicates that the inverse error might be amplified by the additional inverse factors L_C^{-1} and U_C^{-1} from the lower levels. Although this is an extremal case and we expect this to rarely happen in practice, this suggests again that it is strongly advisable to also keep the inverse factors L_C^{-1} and U_C^{-1} bounded.

2.4. Perturbation of the approximate Schur complement. The results discussed so far dealt with perturbations introduced by dropping compared with the diagonal part and the remaining approximate Schur complement

$$\tilde{\mathcal{D}}_k = \begin{pmatrix} \tilde{D}_B & 0 \\ 0 & \tilde{X} \end{pmatrix},$$

where $\tilde{X}$ is one of $\tilde{S}$, $\tilde{M}$, or $\tilde{T}$. Nothing has been said so far about the error between the approximate Schur complements $\tilde{S}$, $\tilde{M}$, and $\tilde{T}$ and the exact Schur complement $S = C - EB^{-1}F$. This information is of great importance since the approximate Schur complement will be used as the input matrix for the next level. The perturbation results deal only with the approximate Schur complement, and as long as the inverse triangular factors are kept bounded it is likely that the errors are small. But in a relative sense even small perturbations may have a serious impact on the preconditioned system if the diagonal entries of $\mathcal{D}_k$ become small in absolute value. This may be a property of the underlying original system. But this may in particular happen if the approximate Schur complement becomes ill conditioned.

For this reason we will investigate the error introduced to the approximate Schur complement by computing the incomplete LU factorization. An incomplete LU decomposition results in an approximate factorization from (2.1) of the type

$$A = \tilde{\mathcal{L}}_k \tilde{\mathcal{D}}_k \tilde{\mathcal{U}}_k + \mathcal{E}_k,$$

where it is assumed that no pivoting is necessary to ensure that $\|\tilde{\mathcal{L}}_k\|, \|\tilde{\mathcal{U}}_k\| \leqslant \kappa$. Suppose that we can also exactly factor A as

$$A = \mathcal{L}_k \mathcal{D}_k \mathcal{U}_k,$$

where

$$\mathcal{L}_k = \begin{pmatrix} L_B & 0 \\ L_E & I \end{pmatrix}, \quad \mathcal{U}_k = \begin{pmatrix} U_B & U_F \\ 0 & I \end{pmatrix}, \quad \mathcal{D}_k = \begin{pmatrix} D_B & 0 \\ 0 & S \end{pmatrix},$$

and $L_B, U_B^\top$ are unit lower triangular; D_B is diagonal, and S is the exact Schur complement.

Comparing the exact Schur complement and the approximate Schur complements we find that

$$\tilde{\mathcal{L}}_k^{-1}\mathcal{L}_k\mathcal{D}_k\mathcal{U}_k\tilde{\mathcal{U}}_k^{-1} = \tilde{\mathcal{D}}_k + \mathcal{F}_k.$$

Looking only at the lower right block yields

$$\text{(2.18)}\quad \left.\begin{array}{c}\tilde{S}\\ \tilde{M}\\ \tilde{T}\end{array}\right\} = S - \tilde{L}_E\tilde{L}_B^{-1}(L_B-\tilde{L}_B)D_B(U_B-\tilde{U}_B)\tilde{U}_B^{-1}\tilde{U}_F - (0,I)\mathcal{F}_k\begin{pmatrix}0\\ I\end{pmatrix}.$$

From this equation we can see that the error between the exact and approximate Schur complements is driven by

1. the previous error $L_B - \tilde{L}_B$ and $U_B - \tilde{U}_B$ between both factorizations,
2. the inverse error $(0,I)\mathcal{F}_k\binom{0}{I}$ produced by dropping and the choice of the approximate Schur complement, and
3. the norm of the inverse triangular factors $\tilde{\mathcal{L}}_k^{-1}$ and $\tilde{\mathcal{U}}_k^{-1}$ ($\|\tilde{L}_E\tilde{L}_B^{-1}\|, \|\tilde{U}_B^{-1}\tilde{U}_F\| \leqslant \kappa$).

Suppose that the error from previous steps is bounded by ε. If we ensure that at step k entries are dropped such that $\|(0,I)\mathcal{F}_K\binom{0}{I}\| \leqslant \varepsilon$ and that $\varepsilon^2\kappa^2 \leqslant \varepsilon$, then the remaining error in (2.18) will reveal the same order ε. We will summarize this observation in a theorem.

Denote the entries of the approximate Schur complements $\tilde{S}^{(l)}, \tilde{T}^{(l)}$ after $l = 1,\ldots,k$ steps of incomplete Gaussian elimination by $\tilde{s}_{ij}^{(l)}$ ($\tilde{t}_{ij}^{(l)}$, respectively). To simplify the analysis we assume that

$$\text{(2.19)}\qquad \Gamma \geqslant |\tilde{s}_{ll}^{(m)}|, |\tilde{t}_{ll}^{(m)}| \geqslant \gamma,$$

$m = 0,\ldots,k$, $l = 1,\ldots,k$. For the leading k diagonal entries this means that they are approximately of the same order. Initially this can be achieved by scaling and reordering, which will in fact be done in the section on numerical examples.

THEOREM 2.6. *Under the assumption* (2.19) *the following holds provided that the inverse triangular factors are bounded by some prescribed constant κ. Assume that the prescribed drop tolerance ε is less than $\frac{1}{\kappa^2}$.*

1. *For the S-version we suppose that in every step m, $m = 1,\ldots,k$, the entries l_{im} and u_{mj} of $\tilde{\mathcal{L}}_m$ and $\tilde{\mathcal{U}}_m$ are dropped only if*

$$|\tilde{l}_{im}|, |\tilde{u}_{mj}| \leqslant \frac{\varepsilon}{\kappa^2}.$$

Then there exists a constant K such that for any entry i,j of the Schur complements we have

$$|s_{ij} - \tilde{s}_{ij}| \leqslant K\varepsilon.$$

2. *For the M-version we consider the same conditions as for the S-version. Then there exists a constant K such that for any entry i,j of the Schur complements we have*

$$|s_{ij} - \tilde{m}_{ij}| \leqslant K(\kappa\varepsilon)^2.$$

3. *Finally we require for the T-version that in every step m, $m = 1, \ldots, k$, the entries l_{im} and u_{mj} of $\tilde{\mathcal{L}}_m$ and $\tilde{\mathcal{U}}_m$ are dropped at most if*

$$|\tilde{l}_{im}|, |\tilde{u}_{mj}| \leqslant \varepsilon.$$

Then there exists a constant K such that for any entry i, j of the Schur complement we have

$$|s_{ij} - \tilde{t}_{ij}| \leqslant K(\kappa\varepsilon)^2.$$

Proof. Suppose that at step m we have

$$|s_{ij}^{(m)} - \tilde{s}_{ij}^{(m)}| \leqslant K\varepsilon$$

for the S- and M-versions, respectively, and $|s_{ij}^{(m)} - \tilde{t}_{ij}^{(m)}| \leqslant K\varepsilon$ for the T-version. Because of (2.19) it follows that there exists a further constant C such that

$$|l_{ij} - \tilde{l}_{ij}|, |u_{ji} - \tilde{u}_{ji}| \leqslant C\varepsilon, \qquad j = 1, \ldots, m, \quad i > j.$$

From (2.18) we obtain that after step $m+1$, the error $|s_{ij}^{(m+1)} - \tilde{s}_{ij}^{(m+1)}|$ and $|s_{ij}^{(m+1)} - \tilde{t}_{ij}^{(m+1)}|$ can be bounded by

$$\max |\tilde{L}_E \tilde{L}_B^{-1}(L_B - \tilde{L}_B) D_B (U_B - \tilde{U}_B) \tilde{U}_B^{-1} \tilde{U}_F| + \max \left| (0, I) \mathcal{F}_k \begin{pmatrix} 0 \\ I \end{pmatrix} \right|.$$

For the first part we obtain the bound

$$\max |\tilde{L}_E \tilde{L}_B^{-1}(L_B - \tilde{L}_B) D_B (U_B - \tilde{U}_B) \tilde{U}_B^{-1} \tilde{U}_F| \leqslant \Gamma \varepsilon^2 \kappa^2.$$

Since $\varepsilon\kappa^2 < 1$, we see that

$$\max |\tilde{L}_E \tilde{L}_B^{-1}(L_B - \tilde{L}_B) D_B (U_B - \tilde{U}_B) \tilde{U}_B^{-1} \tilde{U}_F| \leqslant \Gamma \varepsilon.$$

Suppose we can show that the entries $|s_{ij}^{(m+1)} - \tilde{s}_{ij}^{(m+1)}|$ and $|s_{ij}^{(m+1)} - \tilde{t}_{ij}^{(m+1)}|$ are bounded by $K\varepsilon$. Then the same has to hold for the $(m+1)$ column of $\mathcal{L}_{m+1} - \tilde{\mathcal{L}}_{m+1}$ and $\mathcal{U}_{m+1}^\top - \tilde{\mathcal{U}}_{m+1}^\top$, since by hypothesis (2.19) the diagonal entries of the approximate Schur complement are uniformly bounded away from zero.

It remains to show that we can bound the error $|s_{ij}^{(m+1)} - \tilde{s}_{ij}^{(m+1)}|$, $|s_{ij}^{(m+1)} - \tilde{t}_{ij}^{(m+1)}|$ by $K\varepsilon$; i.e., we have to show that

$$\max |(0, I) \mathcal{F}_k \begin{pmatrix} 0 \\ I \end{pmatrix}| \leqslant L\varepsilon$$

for some constant L.

1. According to Theorem 2.5, for the S-version we will have

$$\mathcal{F}_{m+1} = \tilde{\mathcal{L}}_{m+1}^{-1} \begin{pmatrix} V_B \tilde{D}_B + \tilde{D}_B W_B & 0 \\ 0 & 0 \end{pmatrix} \tilde{\mathcal{U}}_{m+1}^{-1}.$$

Thus, if the entries in L_B and U_B in every step are dropped at most if they are less than ε/κ^2, then the inverse error is also bounded by a constant L times ε. From this it follows that the error $|s_{ij}^{(m+1)} - \tilde{s}_{ij}^{(m+1)}|$ remains below a constant times ε.

2. For step $1,\dots,k$, the M-version and the S-version are the same. The difference between the M-version and the S-version is that after the final step k the remaining Schur complement is computed in a different way; i.e., the lower right block of the inverse error $\mathcal{F}_k$ vanishes.
3. For the T-version we note that by Corollary 2.2 the inverse error $\mathcal{F}_m$ is given by

$$\mathcal{F}_m = \tilde{\mathcal{L}}_m^{-1}\mathcal{V}_m\tilde{\mathcal{D}}_m + \tilde{\mathcal{D}}_m\mathcal{W}_m\tilde{\mathcal{U}}_m^{-1}.$$

That is, the entries that are dropped in $\tilde{\mathcal{L}}_m$ and $\tilde{\mathcal{U}}_m$ are only amplified by one inverse triangular factor. But by construction there is no inverse error in those positions, where the entries of the approximate Schur complement $\tilde{T}$ are located. Thus we don't have to divide by κ (or κ^2). Since $\varepsilon\kappa^2 < 1$, the error between the exact and the approximate Schur complements can be estimated by some constant K times ε. □

Remarks.

- For Theorem 2.6 it is essential to require that (2.19) be fulfilled. This is needed to ensure that the leading k diagonal entries of the approximate Schur complement are uniformly bounded away from zero, as well as to protect $\tilde{L}_E\tilde{L}_B^{-1}(L_B - \tilde{L}_B)D_B(U_B - \tilde{U}_B)\tilde{U}_B^{-1}\tilde{U}_F$ from being amplified by too large diagonal entries in D_B.

  This certainly indicates a strong need for preordering and scaling algorithms that properly prepare the leading block B. To safeguard this process, small diagonal pivots should be moved to the end. This is a side effect of keeping the inverse triangular factors below a bound κ, since $\kappa|s_{mm}^{(m)}| \geqslant \max_{i,j}\{|s_{im}^{(m)}|, |s_{mj}^{(m)}|\}$ is necessary to keep the inverse triangular factors below κ.
- From Theorem 2.6 we may conclude that the entries of $\tilde{\mathcal{L}}_k$ and $\tilde{\mathcal{U}}_k$ outside the leading $k \times k$ block are not needed at all. This is only true if the approximate Schur complement is not computed from these entries but by using $\tilde{\mathcal{L}}_k$, $\tilde{\mathcal{U}}_k$ from (2.17). Otherwise, using Corollary 2.2 we need to keep

$$\max|\mathcal{F}_{m+1}| = \max|\tilde{\mathcal{L}}_{m+1}^{-1}\mathcal{V}_{m+1}\tilde{\mathcal{D}}_{m+1}\tilde{\mathcal{U}}_{m+1}^{-1} + \tilde{\mathcal{L}}_{m+1}^{-1}\tilde{\mathcal{D}}_{m+1}\mathcal{W}_{m+1}\tilde{\mathcal{U}}_{m+1}^{-1}| \leqslant L\varepsilon,$$

  which shows that even the entries in the E and F block need to be less than ε/κ^2.

3. Inverse-based multilevel ILUs.

Corollary 2.2 suggests that, whenever possible, incomplete LU factorizations should be computed in such a way that the inverse triangular factors remain bounded. While keeping the entries of L and U small is straightforward, e.g., with the help of pivoting, making the same demand for the inverse matrices is more delicate. We have adopted a strategy to achieve these features, which is based on combining the following three ingredients.

1. A static preordering of the system that puts the original matrix A in the form

$$P^\top AQ = \begin{pmatrix} B & F \\ E & C \end{pmatrix},$$

   where the leading block B is likely to have nicely bounded inverse triangular factors.
2. A partial incomplete LU factorization which approximately factors $P^\top AQ$ and uses pivoting to keep the inverse triangular factors below a given bound κ. The factorization is partial in that it only proceeds with the elimination of the unknowns corresponding to the B block. Details are given below.

3. A multilevel setting, possibly a recursive one, which completes the partial incomplete factorization of step 2. Indeed, this amounts to (recursively) repeating the previous two steps on the Schur complement system resulting from step 2.

A few clarifications are now given for steps 2 and 3. Step 2 starts with a new reordering $P^\top AQ \to \hat{P}^\top A\hat{Q}$ of A with a smaller leading block. This reordering consists of moving to the end rows and columns that are responsible for undesired large inverse factors. In other words, we obtain

$$\hat{P}^\top A\hat{Q} = \tilde{P}^\top(P^\top AQ)\tilde{Q} = \tilde{P}^\top \begin{pmatrix} B & F \\ E & C \end{pmatrix} \tilde{Q} = \left(\begin{array}{c|cc} B_{11} & B_{12} & F_1 \\ \hline B_{21} & B_{22} & F_2 \\ E_1 & E_2 & C \end{array}\right) \equiv \left(\begin{array}{c|c} \hat{B} & \hat{F} \\ \hline \hat{E} & \hat{C} \end{array}\right) .$$

The incomplete factorization is then computed for

$$\hat{P}^\top A\hat{Q} = \left(\begin{array}{c|c} \hat{B} & \hat{F} \\ \hline \hat{E} & \hat{C} \end{array}\right) \approx \underbrace{\left(\begin{array}{c|c} L_{\hat{B}} & 0 \\ \hline L_{\hat{E}} & I \end{array}\right)}_{\mathcal{L}} \left(\begin{array}{c|c} D_{\hat{B}} & 0 \\ \hline 0 & S \end{array}\right) \underbrace{\left(\begin{array}{c|c} U_{\hat{B}} & U_{\hat{F}} \\ \hline 0 & I \end{array}\right)}_{\mathcal{U}}, \tag{3.1}$$

where $L_{\hat{B}}, U_{\hat{B}}^\top$ are unit lower triangular factors, $D_{\hat{B}}$ is diagonal, and S is the approximate Schur complement. The partial incomplete factorization is performed ensuring that the inverse triangular factors fulfill $\|\mathcal{L}^{-1}\|, \|\mathcal{U}^{-1}\| \leqslant \kappa$.

Further comments on these three pieces of the factorization are given next.

3.1. Template 1: Preorderings.

Standard static ordering techniques can be used to give some desirable properties to the original matrix. Among these are all the graph-based routines to reduce fill-in and more recent approaches to improve diagonal dominance. A few popular examples of fill-reducing methods are the reverse Cuthill–McKee (RCM), the multiple minimum degree ordering (MMD), the nested dissection (ND), and the approximate minimum fill (AMF). See [11] for an overview of some of these reorderings and [16, 1] for the AMF ordering.

The nonsymmetric orderings used in MC64 [10] and in the ddPQ strategy [18] attempt to improve the diagonal dominance of the matrix in different ways. The objective is to rearrange the columns and/or the rows so that the diagonal part gets a maximum weight. In [10] scaling is part of this approach but fill reduction is performed by a static (symmetric) postordering of the resulting matrix. In contrast, ddPQ assumes that some scaling is done a priori and the reordering is performed as a compromise between diagonal dominance and fill. A more detailed description of MC64 is provided in section 4.

3.2. Template 2: Partial incomplete LU decomposition.

A partial (I)LU of a partitioned matrix is a factorization of the matrix in the form of the right-hand side of (3.1). Any Gaussian elimination-based procedure can be adapted to yield a partial factorization. However, the most attractive method exploits the Crout version of Gaussian elimination; see [9, p. 50] and [14]. The Crout (sometimes referred to as the Crout–Doolittle) algorithm computes the kth column of L and the kth row of U at step k. A partial factorization consists of performing a number, say p, steps of this algorithm and then computing the Schur complement. There are several advantages of Crout versions of ILU, some of which are discussed in [14].

In the context of the partial ILU factorization needed for a multilevel scheme, diagonal pivoting is necessary. In fact pivoting is an integral part of the reordering scheme. Diagonal pivoting relies on various estimates for the inverse triangular factors; see, e.g., [6, 12]. When the estimated norm of the inverse triangular factor exceeds the prescribed bounds, then the associated column and row are pushed to the end (see Figure 3.1). From the point of view of implementation, a diagonal pivoting strategy of this type can be easily added to the Crout factorization.

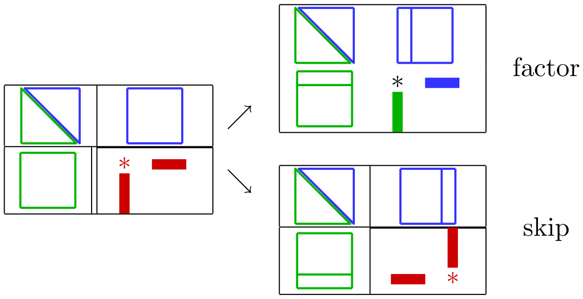


FIG. 3.1. *Diagonal pivoting in the Crout version.*

3.3. Template 3: The multilevel scheme. The next part of the preconditioner involves recursion by repeating the first two steps on the approximate Schur complement resulting from step 2. This leads naturally to a multilevel strategy. To solve a linear system using this multilevel ILU, we could simply proceed as in [20]. Based on the underlying approximate factorization

$$P^\top AQ = \begin{pmatrix} B & F \\ E & C \end{pmatrix} \approx \begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \begin{pmatrix} B & 0 \\ 0 & \tilde{S}_C \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ 0 & I \end{pmatrix},$$

the associated preconditioner would require the application of

$$(P^\top AQ)^{-1} \approx \begin{pmatrix} \tilde{B}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -\tilde{B}^{-1}F \\ I \end{pmatrix} \tilde{S}_C^{-1} \begin{pmatrix} -E\tilde{B}^{-1} & I \end{pmatrix}. \tag{3.2}$$

Here $\tilde{B}$ is the approximation to B corresponding to an ILU of B, and a solve with $\tilde{S}_C$ in (3.2) corresponds to a recursive solve invoking the next "coarser" level.

4. Numerical tests. This section presents numerical experiments to test the inverse-based multilevel ILU approach. We begin by pointing out that the algorithms described in this paper are part of a recently released package called ILUPACK [4], which is available online. Along with the package, the ILUPACK website [4] also posts a rather exhaustive set of experiments with various publicly available test matrices. The numerical computations were performed on an IBM RISC 6000 with four Power 3-II processors (375 MHz). Sixty-four bit address length and up to 4 GB memory were used. All codes use optimization.

The discussion of the experiments centers around two main questions:

- How does the reliability depend on the drop tolerance? (sensitivity)
- How many problems (statistically) can be solved with an incomplete LU factorization (with or without multilevel) if we fix the number of nonzeros of L, U relative to the nonzeros of A?

We will compare three approaches:

1. ILUTP [17, 18] using a binary search tree;

2. inverse-based ILUTC [14], which computes a single ILU without pivoting and uses the inverse triangular factors within dropping;
3. inverse-based multilevel ILU as described in section 3, hereafter referred to as ILUPACK. According to the theory presented here we will use the S-version (SIMPLE version) and the M-version (DEFAULT version).

These methods are combined with several reordering strategies:

(a) AMF [1],
(b) MMD [11],
(c) RCM [11], and
(d) ddPQ, a strategy recently presented in [19], which we will also use for the multilevel strategy.

A recent algorithm called MC64 [10] that is designed to improve the diagonal dominance is also included in the numerical results for comparison and in order to see how MC64 can help improve performance. The MC64 algorithm computes a nonsymmetric permutation via a maximum weight matching based on the bipartite graph representation of the input matrix A. This is a well-known assignment problem in operations research and combinatorial analysis. In the solution of the assignment problem, a permutation matrix P and two vectors $u = (u_i)_{i=1,\dots,n}$ and $v = (v_i)_{i=1,\dots,n}$ are generated (so-called dual variables). $D_r = \text{diag}\ (u_1, \dots, u_n)$ and $D_c = \text{diag}\ (v_1, \dots, v_n)$ can be used to rescale the input matrix and P is used to reorder its columns. The matrix D_rAD_cP resulting from these operations has diagonal entries which are one in modulus and off-diagonal entries which do not exceed one in modulus. Note that in our numerical experiments the time for MC64 is almost negligible relative to that of the incomplete LU factorization and the iterative solution process. This confirms earlier observations made in [2].

All codes that are used refer to their versions as they were implemented in ILUPACK [4]. As iterative solver restarted GMRES(30) [18] is used. The iteration is stopped whenever the residual is reduced by $\sqrt{\text{eps}} \approx 1.5 \cdot 10^{-8}$. The iterative process is deemed to have failed if more than 500 steps are required or a breakdown occurs.

For the inverse-based multilevel ILU we prescribe the bound κ for the inverse factors. However, dropping is not performed with respect to some drop tolerance divided by this bound κ, but by the estimated maximum norm $\tilde{\kappa}$ of L_k^{-1} and U_k^{-1} at step k of the algorithm, which may be less than κ. This is done because one cannot predict a priori whether this bound κ is ever attained, so using $\tilde{\kappa} = \max(\|L_k^{-1}\|, \|U_k^{-1}\|)$ is a good compromise. As mentioned earlier the inverse norms are estimated by essentially using the results of [6].

4.1. Sensitivity with respect to the drop tolerance. The first set of numerical experiments investigates how the success of the incomplete LU decomposition is affected by the choice of the drop tolerance. To measure this sensitivity to drop tolerance, we use 81 test problems from PDEs available at the Davis collection [7] and from the Matrix Market [15]. We chose several larger matrices (at least a few thousand unknowns) that were available at the time of the numerical tests.

Figures 4.1, 4.2, and 4.3 show the relative number (percentage) of problems solved using a specific ILU (ILUTP, inverse-based ILUTC, and inverse-based multilevel ILU) and one of the above orderings (AMF/MMD/RCM/PQ). We show the results for the S-version of the inverse-based multilevel ILU (referred to as ILUPACK) and $\kappa = 10$.

Figures 4.1, 4.2, and 4.3 show clearly that the multilevel ILU is the least sensitive to the given drop tolerance. Using MC64 [10] one could even improve the results further. However, we noticed that for this class of problems only slight improvements

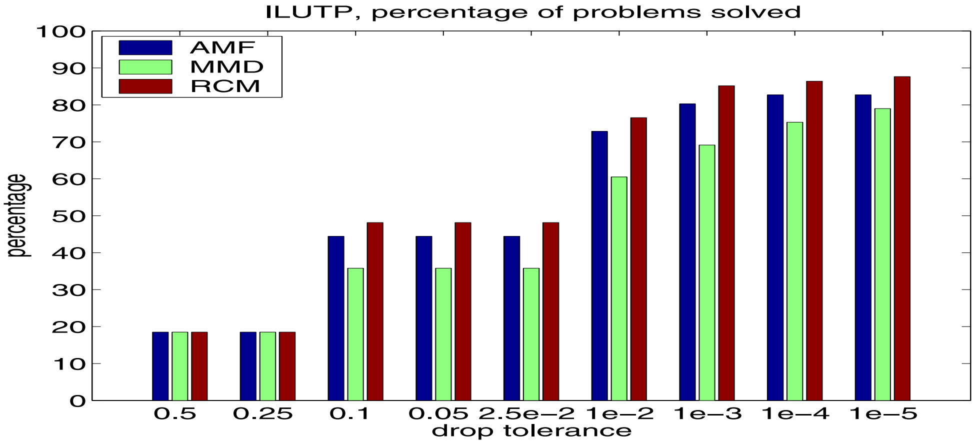


Fig. 4.1. *Numerical sensitivity of ILUTP (PDE problems).*

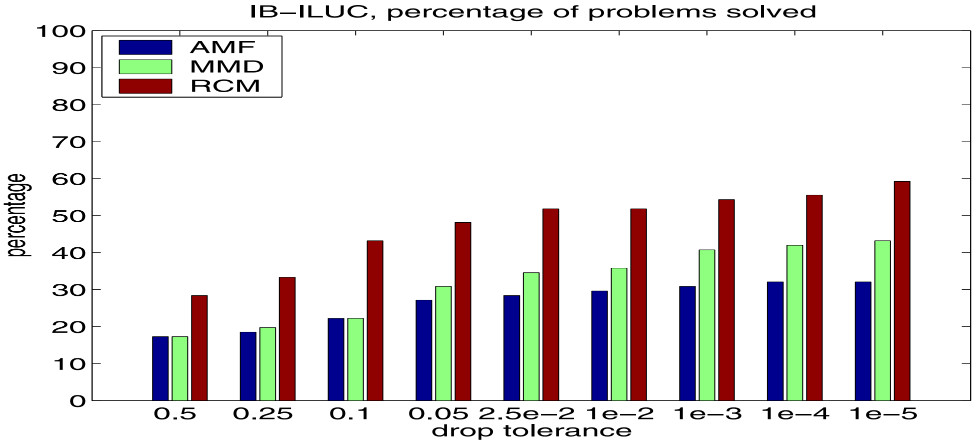


Fig. 4.2. *Numerical sensitivity of inverse-based ILUTC (PDE problems).*

were made. Note that matrices arising from PDEs are typically symmetrically structured and that orderings like AMF, MMD, and RCM, and ILUs such as ILUTC and the inverse-based multilevel ILU preserve symmetry.

4.2. Efficiency via fill-in. The above statistics regarding the dependence of the drop tolerance give only a partial idea of the efficiency of the methods. To get a more detailed view on the fill-in of the triangular factors resulting from the incomplete factorizations, we now study how the number of successfully solved systems is related to the fill-in resulting from the factorization. In a typical situation, an ILU which allows more fill-in is more accurate and should therefore yield faster convergence.

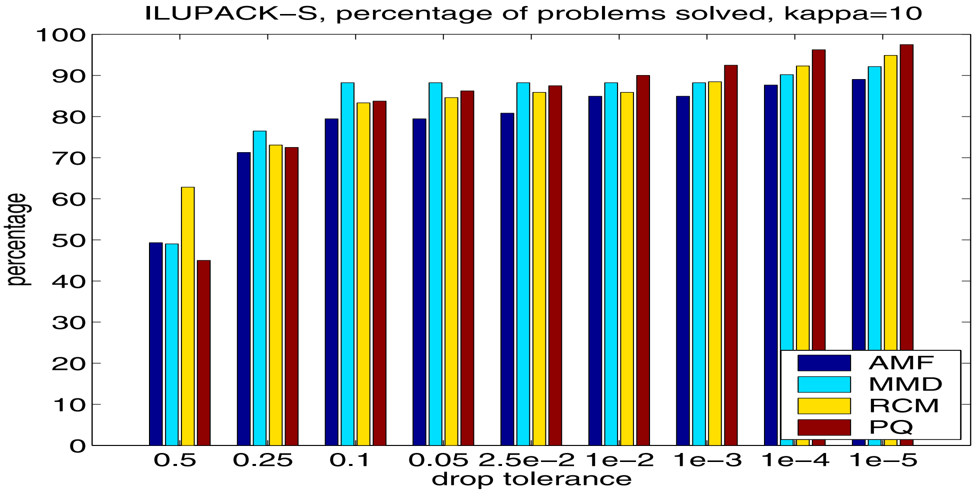


FIG. 4.3. *Numerical sensitivity of ILUPACK, S-version, $\kappa = 10$ (PDE problems).*

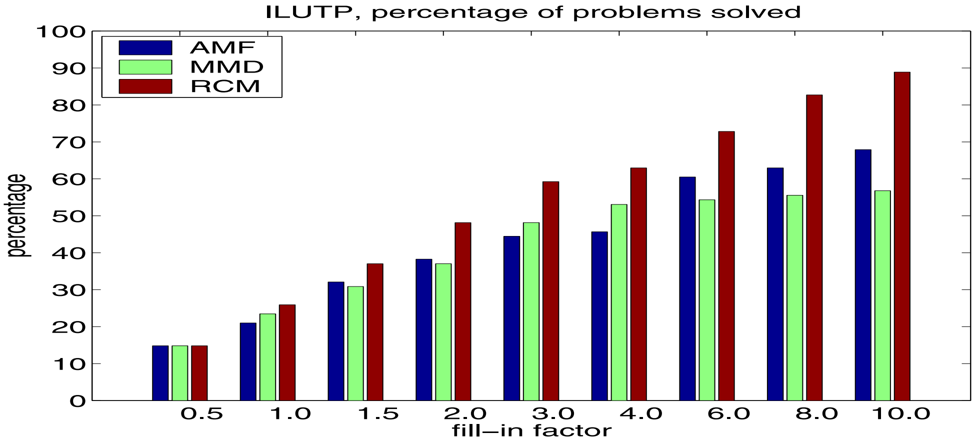


FIG. 4.4. *ILUTP, successful computation versus fill-in (PDE problems).*

The relevant results are in Figures 4.4, 4.5, and 4.6.

It turns out again that the fill-in of the triangular factor of the inverse-based multilevel ILU is the least related to the number of problems that could be solved. Better results can again be obtained by using MC64. Compared with the upcoming section 4.3 on unstructured problems, the improvements by MC64 here are not that impressive and show only a slight improvement.

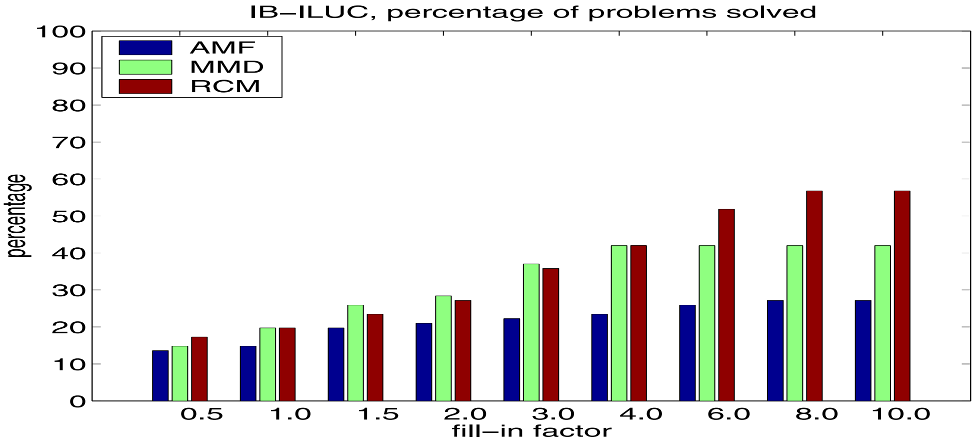


FIG. 4.5. *Inverse-based ILUTC, successful computation versus fill-in (PDE problems).*

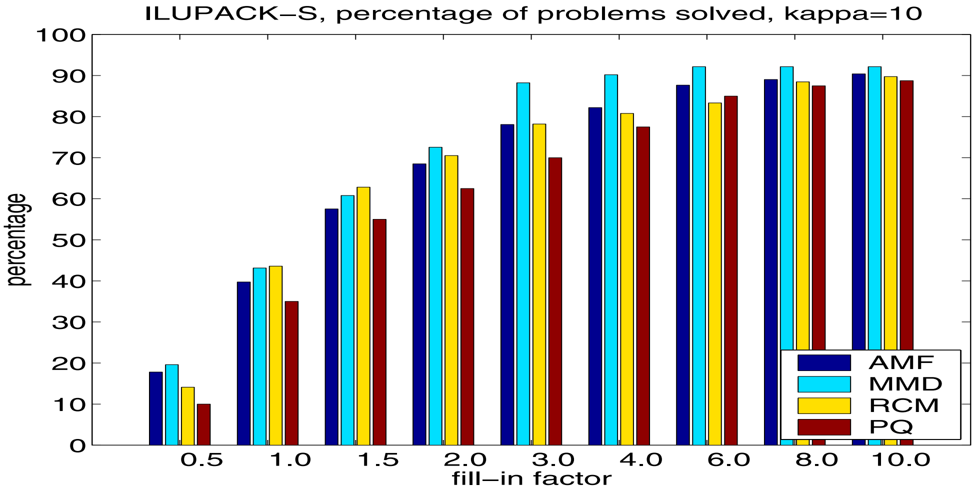


FIG. 4.6. *ILUPACK, S-version, $\kappa = 10$, successful computation versus fill-in (PDE problems).*

4.3. Unstructured problems. To give an idea of how the inverse-based multilevel ILU performs on highly unstructured problems we consider a set of 33 test problems from chemical engineering, available at [7]. These matrices are typically highly indefinite and pivoting, or an a priori nonsymmetric reordering, is often helpful. To show this we first present in Figures 4.7, 4.8, and 4.9 the statistics on the sensitivity with respect to the drop tolerance (analogous to the set of PDE problems) when the original matrices are considered.

It is important to mention that ILUPACK does not implement pure versions

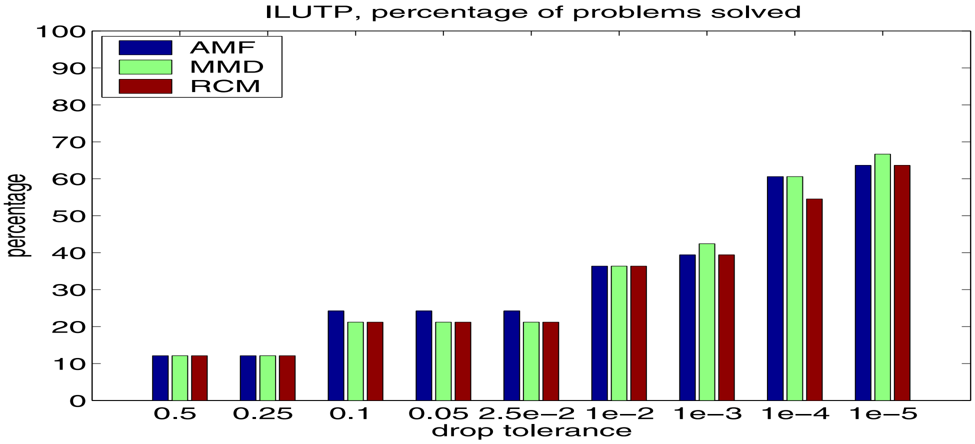


FIG. 4.7. *Numerical sensitivity of ILUTP (chemical engineering).*

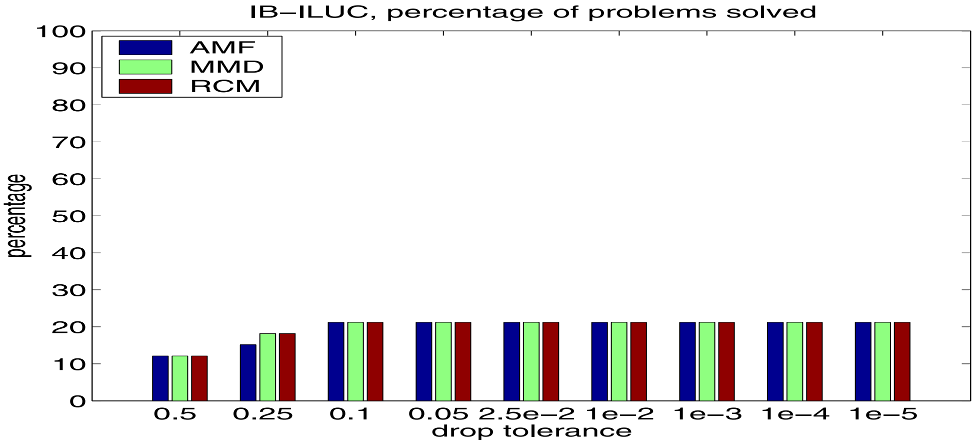


FIG. 4.8. *Numerical sensitivity of inverse-based ILUTC (chemical engineering).*

of the standard symmetric orderings like AMF, MMD, or RCM. Since these do not include any pivoting they are likely to fail on highly unstructured problems. The multilevel ILU in ILUPACK uses only diagonal pivoting, so it was configured so as to automatically switch to the ddPQ ordering when the other simple orderings (AMF, MMD, and RCM) fail to produce a sufficiently large leading diagonal block B. This explains why in Figures 4.7, 4.8, and 4.9 the orderings like AMF, MMD, and RCM perform quite well despite the high indefiniteness of the test problem class.

Due to the high indefiniteness of this problem class, MC64 dramatically improves the numerical performance. On the other hand, relative to ILUTP and ILUTC, the inverse-based multilevel ILU is only mildly affected by this reordering (cf. Figures

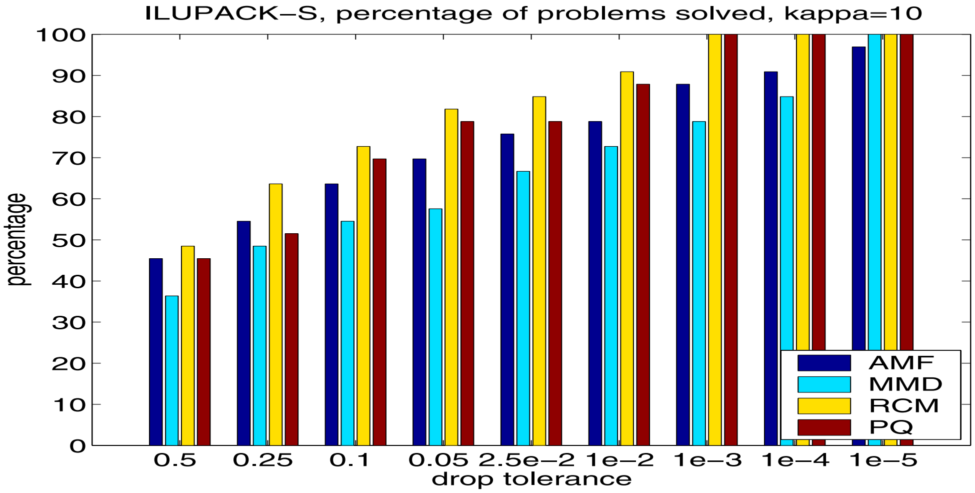


FIG. 4.9. *Numerical sensitivity of ILUPACK, S-version, $\kappa = 10$ (chemical engineering).*

4.10, 4.11, and 4.12).

We should point out again that ILUTC does not include any form of pivoting. This explains why it fails without MC64 for most problems but works very well after MC64 has been applied.

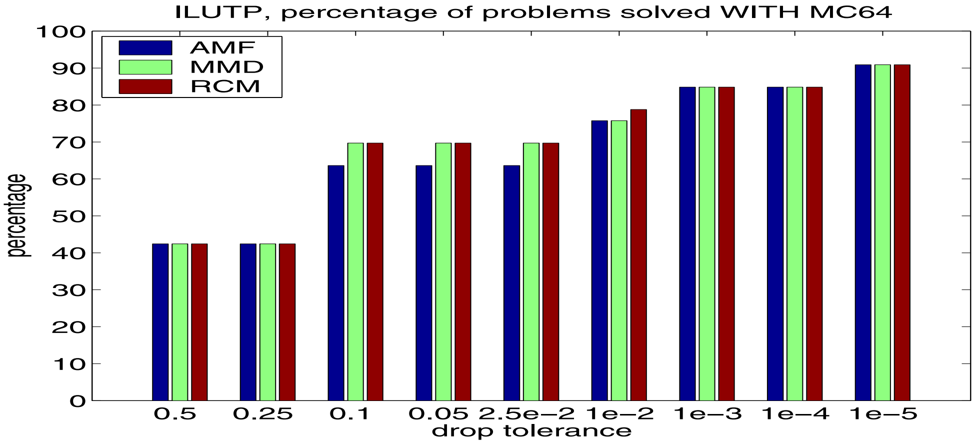


FIG. 4.10. *Numerical sensitivity of $ILUTP + MC64$ (chemical engineering).*

4.4. Variants of the inverse-based multilevel ILU. The previous results compare the new inverse-based multilevel ILU with existing ILU approaches in general terms. In this section we briefly comment on the impact of parameters such as the

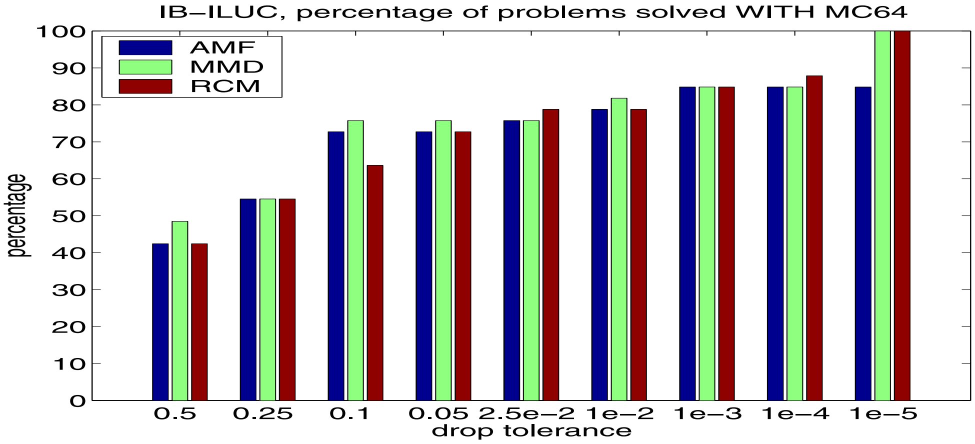


FIG. 4.11. *Numerical sensitivity of inverse-based $ILUTC + MC64$ (chemical engineering).*

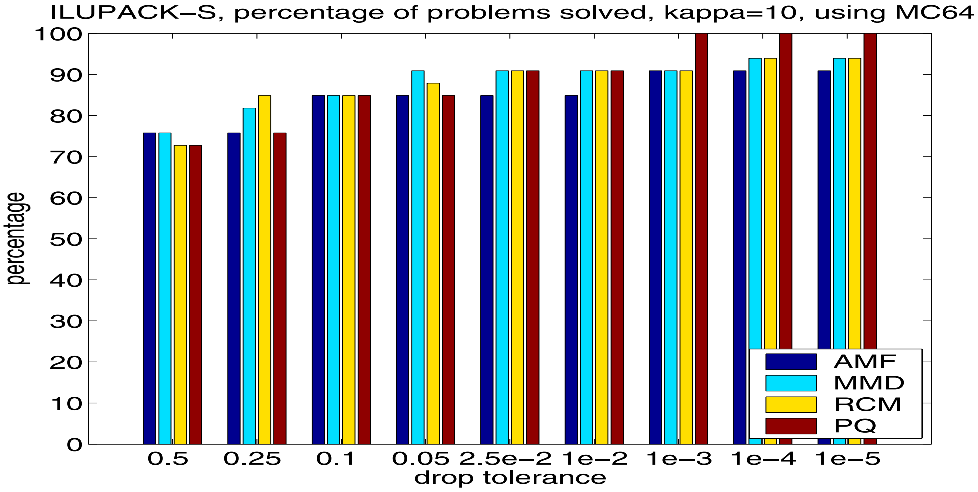


FIG. 4.12. *Numerical sensitivity of $ILUPACK + MC64$, S-version, $\kappa = 10$ (chemical engineering).*

bound κ for the inverse triangular factors, as well as the choice between the S-version and the M-version.

Recall that the norms of the inverse factors are kept below a threshold κ, which is a parameter of the algorithm. In our numerical tests we found that the influence of the prescribed bound κ on performance is relatively mild. This is caused by the use of the maximal norm of the estimates $\tilde{\kappa}$ for $\|L_k^{-1}\|$ and $\|U_k^{-1}\|$ as a substitute for κ. Clearly κ remains an upper bound, but since it is not clear whether this bound will ever be reached by the inverse triangular factor, the estimate $\tilde{\kappa}$ seems to be a good compromise. Only $\tilde{\kappa}$ is used for dropping in conjunction with the drop tolerance.

The theory developed in earlier sections suggests that the M-version will be less sensitive to the drop tolerance since the corresponding approximate Schur complement is more accurate. However, this is likely to comes at a higher expense in terms of fill-in. By using a coarser approximation of the Schur complement, the S-version is likely to require smaller drop tolerances. This was confirmed in the numerical examples.

We now show, in a few tables, test examples where fixed parameters are used. For ILUTP and ILUTC a drop tolerance of $\varepsilon = 10^{-3}$ is used. For the inverse-based multilevel ILU, a significantly coarser drop tolerance $\varepsilon = 10^{-1}$ and $\kappa = 10$ are used. We examine some sample problems which arise in the numerical treatment of PDEs. Therefore, we use RCM as a fixed ordering. The results are summarized in Tables 1, 2, 3, and 4. The results show that ILUTP with RCM is fairly robust, but this robustness comes at an exceedingly high memory cost. On the other hand, the multilevel inverse-based ILU from ILUPACK consumes very little memory, though at least the S-version fails to solve two of the problems.

Table 1
ILUTP+RCM with $\varepsilon = 10^{-3}$ for selected PDE problems.

Name	$\frac{nnz(L+U)}{nnz(A)}$	GMRES	ILU	GMRES	Total
		steps	time	time	time
kim1	1.7	7	3.9e+0	1.4e+0	5.30e+0
rma10	4.0	17	3.5e+1	5.4e+0	4.04e+1
garon2	7.9	12	1.1e+1	1.1e+0	1.21e+1
rim	3.2	49	7.6e+0	6.2e+0	1.38e+1
raefsky3	3.5	7	2.7e+1	1.1e+0	2.81e+1
venkat01	4.2	5	1.6e+1	1.5e+0	1.75e+1
wang4	6.1	12	1.4e+0	7.4e-1	2.14e+0
e40r0000	8.4	11	1.9e+1	1.4e+0	2.04e+1
e40r5000	10.2	7	2.7e+1	9.9e-1	2.80e+1

Table 2
Inverse-based ILUC+RCM with $\varepsilon = 10^{-3}$ for selected PDE problems.

Name	$\frac{nnz(L+U)}{nnz(A)}$	GMRES	ILU	GMRES	Total
		steps	time	time	time
kim1	1.8	6	1.4e+1	1.2e+0	1.52e+1
rma10	5.3	6	3.4e+1	2.6e+0	3.66e+1
garon2	8.1	12	7.7e+0	1.3e+0	9.00e+0
rim	5.4	—	1.2e+1	—	—
raefsky3	4.4	6	2.7e+1	1.1e+0	2.81e+1
venkat01	4.6	5	1.6e+1	1.7e+0	1.77e+1
wang4	7.3	10	2.1e+0	7.1e-1	2.81e+0
e40r0000	10.0	6	1.9e+1	9.8e-1	2.00e+1
e40r5000	13.0	—	2.6e+1	—	—

5. Conclusion.

While it is clear that iterative solvers are unlikely to ever compete with direct solution methods in terms of robustness and generality, one can

Table 3

Inverse-based multilevel ILU+RCM, S-version with $\varepsilon = 10^{-1}$ and $\kappa = 10$ for selected PDE problems.

Name	Levels	$\frac{nnz(L+U)}{nnz(A)}$	GMRES	ILU	GMRES	Total
			steps	time	time	time
kim1	1	0.8	78	6.6e+0	1.3e+1	1.96e+1
rma10	7	1.7	135	2.9e+1	3.4e+1	6.30e+1
garon2	4	1.6	49	3.3e+0	2.3e+0	5.60e+0
rim	13	1.6	—	1.2e+1	—	—
raefsky3	5	0.5	—	4.8e+0	—	—
venkat01	4	1.4	10	7.6e+0	3.0e+0	1.06e+1
wang4	3	1.6	54	9.7e-1	2.1e+0	3.07e+0
e40r0000	3	1.3	101	3.0e+0	7.6e+0	1.06e+1
e40r5000	5	3.2	28	1.8e+1	2.6e+0	2.06e+1

Table 4

Inverse-based multilevel ILU+RCM, M-version with $\varepsilon = 10^{-1}$ and $\kappa = 10$ for selected PDE problems.

Name	Levels	$\frac{nnz(L+U)}{nnz(A)}$	GMRES	ILU	GMRES	Total
			steps	time	time	time
kim1	1	0.8	78	6.7e+0	1.2e+1	1.87e+1
rma10	6	2.4	31	1.0e+2	1.0e+1	1.10e+2
garon2	3	1.9	27	1.4e+1	1.2e+0	1.52e+1
rim	10	2.2	—	3.1e+1	—	—
raefsky3	3	0.5	81	1.3e+1	7.8e+0	2.08e+1
venkat01	4	1.4	10	2.2e+1	2.9e+0	2.49e+1
wang4	4	1.6	45	2.8e+0	1.6e+0	4.40e+0
e40r0000	3	1.4	37	9.2e+0	2.1e+0	1.13e+1
e40r5000	4	4.4	22	8.4e+1	2.4e+0	8.64e+1

certainly argue that recent progress in the field has considerably shortened the gap between the two classes of methods. The ingredients in this progress came in large part from a better understanding of the impact of dropping on the inverse factors. As long as ILU techniques were only applied to easy problems arising from elliptic-type PDEs, little attention was paid to these inverses because their condition numbers are often within reasonable range. When the success of ILUs grew and their range of applicability expanded, it was realized that the classical ideas that were behind the development of ILU basically failed. Recent work [3] showed that it becomes critical in this situation to analyze the effect of dropping on the inverse factors, and a further analysis of this was given in section 2. In particular, it becomes critical to design the preconditioners in such a way that the perturbations of the inverse factors, caused by dropping, remain small. From a practical implementation viewpoint, this calls for a multilevel strategy which strives to yield bounded inverse factors of the B block corresponding to the "fine" level. Then, recursion can be invoked to deal with the Schur complement resulting from the fine-coarse partitioning.

A package based on three "templates" for implementing the basic ideas has been implemented and thoroughly tested. Numerical experiments indicate that this multilevel ILU can be a good alternative to direct solvers in most cases. They also show that nonsymmetric reorderings, diagonal pivoting, and efficient Crout implementations can contribute to a better robustness and efficiency of the preconditioner. The code and a large set of experiments are available online; see [4].

Still lacking is a parallel implemention of the algorithms. While the algorithms have not been designed with parallelism in mind, many of the ideas are extensible to parallel environments, at a minimum via a domain decomposition viewpoint.

REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[2] M. Benzi, J. C. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.

[3] M. Bollhöfer, *A robust and efficient ILU that incorporates the growth of the inverse triangular factors*, SIAM J. Sci. Comput., 25 (2003), pp. 86–103.

[4] M. Bollhöfer and Y. Saad, *ILUPACK—Preconditioning Software Package*, http://www.math.tu-berlin.de/ilupack/.

[5] M. Bollhöfer and Y. Saad, *On the relations between ILUs and factored approximate inverses*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 219–237.

[6] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16 (1979), pp. 368–375.

[7] T. Davis, *University of Florida Sparse Matrix Collection*, http://www.cise.ufl.edu/research/sparse/matrices.

[8] T. A. Davis, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 165–195.

[9] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, UK, 1986.

[10] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.

[11] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[13] A. Gupta, *Improved symbolic and numerical factorization algorithms for unsymmetric sparse matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 529–552.

[14] N. Li, Y. Saad, and E. Chow, *Crout versions of ILU for general sparse matrices*, SIAM J. Sci. Comput., 25 (2003), pp. 716–728.

[15] National Institute of Standards, *Matrix Market*, http://math.nist.gov/MatrixMarket.

[16] E. Rothberg and S. C. Eisenstat, *Node selection strategies for bottom-up sparse matrix ordering*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 682–695.

[17] Y. Saad, *ILUT: A dual threshold incomplete ILU factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.

[18] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.

[19] Y. Saad, *Multilevel ILU with reorderings for diagonal dominance*, SIAM J. Sci. Comput., 27 (2005), pp. 1032–1057.

[20] Y. Saad and B. J. Suchomel, *ARMS: An Algebraic Recursive Multilevel Solver for General Sparse Linear Systems*, Technical Report umsi–1999-107, University of Minnesota at Minneapolis, Dept. of Computer Science and Engineering, 1999.

[21] O. Schenk and K. Gärtner, *Solving unsymmetric sparse systems of linear equations with PARDISO*, J. Future Generation Comput. Syst., 20 (2004), pp. 475–487.

[22] H. D. Simon, *Incomplete LU preconditioners for conjugate gradient type iterative methods*, in Proceedings of the SPE 1985 Reservoir Simulation Symposium, Dallas, TX, Society of Petroleum Engineers of AIME, 1988, paper number 13533, pp. 302–306.

[23] M. Tismenetsky, *A new preconditioning technique for solving large sparse linear systems*, Linear Algebra Appl., 154–156 (1991), pp. 331–353.