Applied Numerical Mathematics 31 (1999) 209–225

APPLIED NUMERICAL MATHEMATICS

www.elsevier.nl/locate/apnum

A multilevel block incomplete factorization preconditioning

Y. Notay [1,2]

Service de Métrologie Nucléaire, Université Libre de Bruxelles (C.P. 165), 50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium

---

Abstract

Incomplete factorization preconditioners based on recursive red–black orderings have been shown efficient for discrete second order elliptic PDEs with isotropic coefficients. However, they suffer for some weakness in presence of anisotropy or grid stretching. Here we propose to combine these orderings with *block* incomplete factorization preconditioning techniques.

For implementation considerations, the latter are extended to the case where the block pivots are generalized tridiagonal matrices, say matrices that have at most one nonzero entry per row in their strictly upper triangular part. On the other hand, a new block method is introduced for the improvement of the performance. This method is called IMBILU (improved modified block ILU).

Numerical results show that the resulting preconditioner is efficient and robust with respect to both discontinuity and anisotropy in the PDE coefficients.

Keywords: Iterative methods; Linear systems; Acceleration of convergence; Preconditioning

---

1. Introduction

This paper deals with the iterative solution of large sparse symmetric positive (nonnegative) definite linear systems

$$Ax = b \tag{1.1}$$

arising from the finite difference discretization of two dimensional second order elliptic PDEs. In such cases, the conjugate gradient method [4] is widely used and increasing attention has been paid to preconditioners based on a MILU factorization of A computed with respect to a recursive red–black ordering of the unknowns, see [6–9,16,28] and also the more general approach developed in [32]. These preconditioners indeed combine the efficiency of multilevel methods with the ease of implementation of approximate factorization preconditioners, and have recently been proved of near optimal order for PDEs with isotropic coefficients [28]. They can further be made optimal by using a proper (W cycle type) inner

---

[1] E-mail: ynotay@ulb.ac.be.

[2] Supported by the “Fonds National de la Recherche Scientifique”, Chercheur qualifié.

PII: S0168-9274(98)00133-0

polynomial acceleration [2,6–9]. Note that using "modified" ILU (where discarded fill entries are lumped to the main diagonal) is essentially motivated here by multigrid like considerations [6–8,16,28] (see also Section 2).

Now, the method suffers from some weakness in presence of anisotropy, most likely because some of the discarded fill entries are then very large. In this paper, we propose to improve the basic scheme by accepting these large fill entries, coping with this extra fill by using a blockwise incomplete factorization scheme instead of a pointwise one.

Motivated by this particular context, two enhancements of standard block ILU algorithms [4,5,10,17, 22] are also introduced.

First, accepting large fill entries while keeping the block pivots tridiagonal (or with small bandwidth) would require a problem dependent ordering strategy which could be difficult to implement for complicated problems. We therefore found it more practical to extend somewhat classical block ILU techniques by letting the block pivots be *generalized* tridiagonal, that is with at most one nonzero per row in the strictly upper triangular part (like usual tridiagonal matrices), but in arbitrary position.

Next, we observed that using in this context the standard modified block ILU (MBILU) factorization scheme does actually not allow to improve much the results obtained with the basic pointwise MILU algorithm. This led us to investigate improvement of standard MBILU, and we propose here a new technique. It reduces to MBILU for the most common case of line partitioning, but, in more general circumstances, it is expected to improve the conditioning properties and in any event not deteriorate them, reasons for which we refer to it as improved MBILU or IMBILU.

Numerical experiments show that the resulting preconditioner, whose basic pointwise version was already found robust with respect to jumps in the PDE coefficients [9,28], is now robust with respect to anisotropy too.

It is worth mentioning some of the previous attempts to improve the considered preconditioning technique in the presence of anisotropy. The method in [9] looks similar to the one presented here, as it is also based on the acceptance of more fill-in. However, only inverse free preconditioners were considered, and this was found to bring only a limited improvement. The methods in [3,23] are more successful, but require a problem dependent ordering strategy. The method in [32] controls the fill-in by value rather than by position, which improves the robustness in presence of anisotropy at the price of a considerable increase of the fill-in. A convincing example on a stretched grid is provided, which however also comes with a change in the ordering policy, the recursive red–black like approach followed for the model problem being apparently left for an ordering inspired by multigrid techniques. Finally, the method in [15] fixes the numbering only during the numeric factorization, as a function of the value of the matrix coefficients in the generated Schur complements. Results are promising, especially because this strategy is also applicable to unstructured matrices. However, this dynamic renumbering makes the preprocessing step time consuming and difficult to implement [29]. Note that with the approach followed in the present paper, the ordering is fixed prior to the factorization according to the standard recursive red–black partitioning (see Section 2). Once the matrix is permuted, the preprocessing cost is similar to that of any block incomplete factorization method (see Section 5).

The remainder of this paper is organized as follows: general terminology and notation are summarized below; (block) incomplete factorizations with recursive red–black ordering are presented in Section 2; basic properties of IMBILU are stated in Section 3; the use of generalized tridiagonal matrices in block incomplete factorization algorithms is discussed in Section 4, and the results of numerical experiments are reported in Section 5.

General terminology and notation. The symbols A^{T} and $\mathcal{N}(A)$ denote the transpose and the null space of the matrix A, respectively. The order relation between real matrices and vectors is the usual componentwise order: if $A=(A_{ij})$ and $B=(B_{ij})$ then $A\leqslant B$ $(A<B)$ if $A_{ij}\leqslant B_{ij}$ $(A_{ij}<B_{ij})$ for all i,j. A is called nonnegative (positive) if $A\geqslant 0$ $(A>0)$. The Hadamard product $A*B$ of the matrices $A=(A_{ij})$ and $B=(B_{ij})$ of the same dimensions is defined through element by element multiplication: $(A*B)_{ij}=A_{ij}\,B_{ij}$. $e=(1,\ldots,1)^{\mathrm{T}}$ is the vector with all components equal to unity and ε is the matrix whose entries are all equal to unity. If $A=(A_{ij})$, we denote by $\operatorname{diag}(A)$ the pointwise (diagonal) matrix whose entries are $A_{ij}\,\delta_{ij}$ and we set $\operatorname{offdiag}(A)=A-\operatorname{diag}(A)$.

When referring to matrices or vectors in block form, we assume that some partitioning $\mathcal{L}=(L_I)_{I=1,\ldots,M}$ of the set of the first n integers is defined according to which n-vectors $x=(x_I)$ are partitioned into block components x_I of dimensions n_I (with $n_I=\#L_I,\ I=1,\ldots,M$) and $n\times n$ matrices A into block components A_{IJ} of dimensions $n_I\times n_J$. Lower case indices refer to scalar entries and capital indices to block entries.

2. Factorizations with recursive red–black ordering

Factorizations with recursive red–black (or other multilevel) ordering are usually described by a recursive procedure, although nothing prevents using the same presentation as for general incomplete LU algorithms. The algorithm below corresponds to the basic pointwise version (e.g., [6–8,16,28]). It includes the “modification”, that is the preconditioner is computed so as to have the same row-sum as A. In the present context, the motivation for this rule is different than with natural orderings, and is essentially inspired by multigrid like considerations. Indeed, when preserving the row-sum, the successive $A^{(I)}$ resemble discretization matrices on coarser grids [6–8,16,28], which would not be true anymore without the “modification” ($A^{(I)}$ becomes then strongly diagonally dominant for all $I>1$).

Algorithm 2.1.

```
Let $A^{(1)}=A$.
For $I=1,\ldots,M-1$:
  (i) partition $A^{(I)}=A$ in a $2\times 2$ block form:
        $$A^{(I)}=\begin{pmatrix}A_{11}^{(I)} & A_{12}^{(I)}\\ A_{21}^{(I)} & A_{22}^{(I)}\end{pmatrix};$$
  (ii) approximate $A_{11}^{(I)}$ by the diagonal matrix $P_{II}$ with same row-sum;
  (iii) form the Schur complement $S^{(I)}=A_{22}^{(I)}-A_{21}^{(I)}\,P_{II}^{-1}\,A_{12}^{(I)}$ so that
        $$A^{(I)}=\begin{pmatrix}A_{11}^{(I)}-P_{II} & 0\\ 0 & 0\end{pmatrix}+\begin{pmatrix}P_{II} & 0\\ A_{21}^{(I)} & S^{(I)}\end{pmatrix}\begin{pmatrix}I_{II} & P_{II}^{-1}\,A_{12}^{(I)}\\ 0 & I_{22}^{(I)}\end{pmatrix}$$
        (where $I_{II}$, $I_{22}^{(I)}$ are the unit matrices of the appropriate dimensions).
  (iv) let $A^{(I+1)}=S^{(I)}$;
Let $P_{MM}=A^{(M)}$.
```

The preconditioner can then be expressed as

$$B = LP^{-1}U, \tag{2.1}$$

where

$$U = L^{\mathrm{T}} = \begin{pmatrix} P_{11} & & A_{12}^{(1)} & & \\ & \ddots & \ddots & & \\ & & P_{II} & A_{12}^{(I)} & \\ & & & \ddots & \ddots \\ & & & & P_{MM} \end{pmatrix}, \qquad P = \begin{pmatrix} P_{11} & & & & \\ & \ddots & & & \\ & & P_{II} & & \\ & & & \ddots & \\ & & & & P_{MM} \end{pmatrix}. \tag{2.2}$$

The procedure above looks like a block incomplete factorization, but it is actually equivalent to a pointwise one because the block pivots $P_{II},\ I = 1, \dots, M-1$, are diagonal. The Schur complement $S^{(I)}$ can thus be computed as in methods based on pointwise elimination. Presentations based on the latter framework also state that the last block P_{MM} has to be factorized exactly. This is not explicit in the scheme above, but it is needed in practice since solving a system with B requires solving a system with P_{MM}.

We now complete the description by defining the partitioning strategy. It corresponds to a multilevel partitioning with a further red–black partitioning of each node set.

More precisely, consider a regular rectangular grid (with boundaries not necessarily regular). Let h be the mesh size on this (finest) grid. One first partitions the nodes set into the usual subsets $S_k,\ k = 0, \dots, k_0$, where S_{k_0} contains the nodes corresponding to grid of mesh size $2^{k_0}h$, S_{k_0-1} those of the remaining nodes corresponding to a grid of mesh size $2^{k_0-1}h$, etc.

The 2×2 partitioning to be used at step I is then as follows. If I is odd, say $I = 2k+1$, it corresponds to the usual red–black partitioning of the rectangular grid of mesh size $2^k h$ defined by the nodes in $S_k \cup \cdots \cup S_{k_0}$, where one selects for the first block the group that contains only nodes in S_k. For $I = 2k+2$, the first block is formed with the nodes in S_k that were not selected at previous step, and the second block is formed with nodes in $S_{k+1} \cup \cdots \cup S_{k_0}$, that is the nodes of the grid of mesh size $2^{k+1}h$. We note L_I the subset of nodes corresponding to block I, that is the first block of the 2×2 partitioning used at step I. Considering then as in Fig. 1 the boxes built on the nodes of the grid of mesh size $2^{k+1}h$, the above procedure implies that the nodes in S_k that correspond to mid edge points are in L_{2k+1}, and that those corresponding to center of boxes are in L_{2k+2}.

Concerning the last level, S_{k_0} is partitioned as previous levels if $M = 2(k_0 + 1)$ is even, but one will use $L_{2k_0+1} = L_M = S_{k_0}$ if M is odd.

This partitioning strategy, referred to as recursive (or repeated) red–black [6–8,16,28], is depicted in Fig. 2 with the induced numbering for a 9×9 grid and $M = 5$ (thus $k_0 = 2$).

Note that the choice of M (or k_0) results from some compromise: smaller values of M imply fewer iterations but larger factorization costs, because of the exact inversion of the last block; $M \approx \log_2 n^{1/2}$ guarantees that the factorization cost is $\mathrm{O}(n)$ [16,28]; slightly smaller values are however generally preferable.

Concerning the incomplete factorization process, it is worth noting that the fill-in remains moderate although Algorithm 2.1 discards only fill entries inside the same diagonal block. Indeed, only four nonzero entries per row are generated in the block offdiagonal part, see [16,28]. This corresponds to the

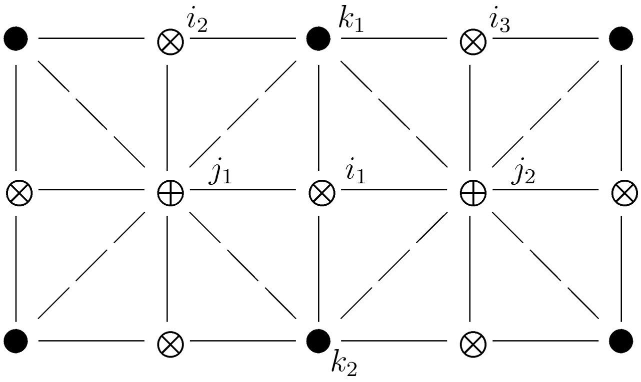


Fig. 1. Part of the grid of mesh size $2^k h$. The filled circles correspond to nodes in $S_{k+1} \cup \cdots \cup S_{k_0}$ and the others to nodes in S_k; among them, those marked $\times$ are in L_{2k+1} and those marked $+$ are in L_{2k+2}.

79	37	**67**	38	**80**	39	**68**	40	**81**
32	*53*	33	*54*	34	*55*	35	*56*	36
64	28	***71***	29	**65**	30	***72***	31	**66**
23	*49*	24	*50*	25	*51*	26	*52*	27
76	19	**62**	20	**77**	21	**63**	22	**78**
14	*45*	15	*46*	16	*47*	17	*48*	18
59	10	***69***	11	**60**	12	***70***	13	**61**
5	*41*	6	*42*	7	*43*	8	*44*	9
73	1	**57**	2	**74**	3	**58**	4	**75**

Fig. 2. Recursive red–black ordering for a 9×9 grid and $M = 5$ ($k_0 = 2$); $S_0 = L_1 \cup L_2$ with $L_1 = \{1, \dots, 40\}$, $L_2 = \{\mathit{41}, \dots, \mathit{56}\}$; $S_1 = L_3 \cup L_4$ with $L_3 = \{\mathbf{57}, \dots, \mathbf{68}\}$, $L_4 = \{\boldsymbol{69}, \dots, \boldsymbol{72}\}$; $S_2 = L_5 = \{\mathbf{73}, \dots, \mathbf{81}\}$.

connections in the five point schemes associated with the successive grids of mesh size $h, 2h, \dots, 2^{k_0}h$, plus the skew connections joining the nodes at the center of the boxes, built on the grid of mesh size $2^k h, k \geqslant 1$ (i.e., the nodes in L_{2k}), to the corner of these boxes (i.e., their nearest neighbors in $L_{2k+1} \cup \cdots \cup L_M$). This is also illustrated in Fig. 1, where the straight lines represent the five point scheme connections for the grid of mesh size $2^k h$, and the dashed lines indicate the additional skew connections between nodes in L_{2k+2} and those in subsequent blocks.

Analyzes have been reported that explain the efficiency of the resulting preconditioner for discrete PDEs with isotropic coefficients (e.g., [6–8,16,28]). All these analyzes fail for anisotropy. See, for instance, [3] where explicit calculations are reported. We want to give here a more intuitive explanation of this phenomenon.

In this view, consider for instance the fill entries generated by the elimination of node i_1 in Fig. 1. The four nonzero offdiagonal entries in the i_1th row of $A_{12}^{(2k+1)}$ connect i_1 to j_1, j_2, k_1 and k_2. Since only j_1 and j_2 belong to the same block, only the fill entry $A_{j_1 i_1}^{(2k+1)} P_{i_1 i_1}^{-1} A_{i_1 j_2}^{(2k+1)}$ between j_1 and j_2 is to be discarded. Now, in presence of anisotropy, this discarded entry may be by far larger than the accepted ones. More precisely, this will happen if the strong coupling corresponds to the horizontal direction. On the other hand, a strong coupling in the vertical direction will entail problems when eliminating nodes like i_2 and i_3. Therefore, in any case, one has to expect the factorization to be less and less accurate when anisotropy becomes stronger.

Motivated by this reasoning, we propose to improve the method by accepting these large fill entries. Noting that at most one of them lies in each row of the upper triangular part of the successive block pivots P_{II}, this is obtained by exchanging step (ii) in Algorithm 2.1 for

(ii) approximate $A_{11}^{(I)}$ by the symmetric generalized tridiagonal matrix P_{II} with same row-sum and whose strictly upper triangular part contains in each row the largest entry in absolute value from the strictly upper triangular part of $A_{11}^{(I)}$.

Note that in this procedure the lower triangular part of P_{II} is fixed from the requirement that P_{II} is symmetric.

Since generalized tridiagonal matrices can, as usual tridiagonal ones, be factorized without fill-in, solving a system with B, which requires two solves of systems with P_{II}, $I = 1, \dots, M-1$, is not much more costly than with the pointwise version. However, P_{II}^{-1} is now full, so that steps (iii) and (iv) are to be updated according to techniques used in (modified) block ILU algorithms. This means that one has to compute some sparse approximation K_{II} to P_{II}^{-1} and let

$$S^{(I)} = A_{22}^{(I)} - A_{21}^{(I)} K_{II} A_{12}^{(I)} - \Omega_{II}^{(S)}, \tag{2.3}$$

where $\Omega_{II}^{(S)}$ is the diagonal matrix such that

$$\Omega_{II}^{(S)} e = A_{21}^{(I)} \left(P_{II}^{-1} - K_{II}\right) A_{12}^{(I)} e \tag{2.4}$$

used to ensure $Be = Ae$ as with the pointwise version; finally, since in general K_{II} will not be just diagonal, to avoid extra fill-in, one has to let now $A^{(I+1)}$ be some sparse approximation to $S^{(I)}$ (with same row-sum) instead of $A^{(I+1)} = S^{(I)}$.

As discussed in the next section, usual block ILU algorithms compute K_{II} such that $0 \leqslant K_{II} \leqslant P_{II}^{-1}$. However, our analysis will show that it is preferable to compute the diagonal of K_{II} such that

$$K_{II} A_{12}^{(I)} e = P_{II}^{-1} A_{12}^{(I)} e, \tag{2.5}$$

and better numerical results are obtained with this novel method. The factorization algorithm is then also simpler since $\Omega_{II}^{(S)} = 0$; moreover, using K_{II} diagonal turned then out to be optimal or near optimal, no significant improvements being obtained with more elaborate approximate inverses computed as indicated in Section 4. We give below the corresponding algorithm.

Algorithm 2.2.

```
Let $A^{(1)} = A$.
For $I = 1, \dots, M-1$:
  (i) partition $A^{(I)} = A$ in a $2 \times 2$ block form:
```

$$A^{(I)} = \begin{pmatrix} A_{11}^{(I)} & A_{12}^{(I)} \\ A_{21}^{(I)} & A_{22}^{(I)} \end{pmatrix};$$

(ii) approximate $A_{11}^{(I)}$ by the symmetric generalized tridiagonal matrix P_{II} with same row-sum and whose strictly upper triangular part contains in each row the largest entry in absolute value from the strictly upper triangular part of $A_{11}^{(I)}$;

(iii) let K_{II} be the diagonal matrix such that

$$K_{II}A_{12}^{(I)}e = P_{II}^{-1}A_{12}^{(I)}e,$$

and form the Schur complement $S^{(I)} = A_{22}^{(I)} - A_{21}^{(I)}K_{II}A_{12}^{(I)}$ so that

$$A^{(I)} = \begin{pmatrix} A_{11}^{(I)} - P_{II} & 0 \\ 0 & A_{21}^{(I)}\left(K_{II} - P_{II}^{-1}\right)A_{12}^{(I)} \end{pmatrix} + \begin{pmatrix} P_{II} & 0 \\ A_{21}^{(I)} & S^{(I)} \end{pmatrix}\begin{pmatrix} I_{II} & P_{II}^{-1}A_{12}^{(I)} \\ 0 & I_{22}^{(I)} \end{pmatrix}$$

(where I_{II}, $I_{22}^{(I)}$ are the unit matrices of the appropriate dimensions);

(iv) let $A^{(I+1)} = S^{(I)}$.

Let $P_{MM} = A^{(M)}$.

Note that using generalized tridiagonal diagonal block pivots raise no additional implementation problem: it suffices to use a real vector to store the values together with an integer vector to store the column indexes, both of which being filled during the numeric factorization. For the remaining, one may use the same implementation techniques as in the pointwise case; in particular, it is not difficult to set up a relevant data structure for the block offdiagonal part of U since the number of nonzero entries per row is known to be equal to 4 (or less for nodes near boundaries) [28].

3. Improved modified block ILU for Stieltjes matrices

Let β be a given symmetric matrix such that $0 \leqslant \beta_{ij} \leqslant 1$ for all i, j and Ω a (pointwise) diagonal matrix. Assuming that A is a Stieltjes matrix, block ILU algorithms (e.g., [1,4,10,22]) compute a preconditioner of the form (2.1), (2.2) such that

$$P - E - F = A - \beta * (EKF) - \Omega, \tag{3.1}$$

where $-F = -E^{\mathrm{T}} = U - P$ is the strictly block upper triangular part of U, whereas K is a nonnegative symmetric block diagonal matrix that approximates P^{-1}.

There are thus three basic ingredients.

First, β is a $(0,1)$ matrix that controls fill-in. We do not refer explicitly to β in the discussion of algorithms in the preceding section, but fill-in control is implicit when we write “Let P_{II} be some (diagonal or generalized tridiagonal) sparse approximation to $A_{11}^{(I)}$” and “Let $A^{(I+1)}$ be some sparse approximation to $S^{(I)}$”.

Next, unmodified block ILU (BILU) factorizations use $\Omega = 0$ whereas modified block ILU (MBILU) algorithms use Ω so as to satisfy

$$Be = Ae, \tag{3.2}$$

that is Ω such that

$$\Omega e = \left(EP^{-1}F - \beta * (EKF)\right)e \tag{3.3}$$

(thus Ω is $\Omega^{(S)}$ referred in Section 2 plus an additional compensation term for the entries discarded because of β).

Finally, some method has to be supplied to compute a relevant sparse approximation K_{II} to P_{II}^{-1}. Little emphasis is put in the literature on the choice of the latter. In practice, a widespread technique consists in using K as a principal band portion (most often tridiagonal) of P^{-1}, assuming that P itself is a band (tridiagonal) matrix. Some other choices are discussed in [5,10,17]. Because P is a Stieltjes matrix and satisfies therefore $P^{-1} \geqslant 0$, most of them, if not all, are such that

$$0 \leqslant K \leqslant P^{-1}, \tag{3.4}$$

and this relation is widely used as assumption in existence and conditioning analysis theorems about block incomplete factorizations, see [4, Section 7.3], for instance.

However, see, e.g., [11], condition (3.4) is actually not necessary to prove the existence (i.e., the positive definiteness) of the preconditioner, at least in the modified case (Ω given by (3.3)). This has been little emphasized, probably because this relation remains anyway necessary to prove by the classical arguments that MBILU factorizations satisfy

$$(z, Az) \geqslant (z, Bz) \quad \forall z \in \mathcal{C}^n, \tag{3.5}$$

i.e., that 1 (which is an eigenvalue by (3.2)) is the lowest eigenvalue of the preconditioned system $B^{-1}A$. This latter property serves as basis for the conditioning analysis of these methods and extends a similar well known result for MILU factorizations.

Now, having a glance at Algorithm 2.2, one can see that (3.5) will still hold although the diagonal entries of K are here larger than the corresponding ones in P^{-1}; indeed, the remainder matrix $R = A - B$ is the sum of 2×2 block diagonal matrices

$$\begin{pmatrix} A_{11}^{(I)} - P_{II} & 0 \\ 0 & A_{21}^{(I)}\left(K_{II} - P_{II}^{-1}\right)A_{12}^{(I)} \end{pmatrix}$$

which are readily found nonnegative definite since both $A_{11}^{(I)} - P_{II}$ and $K_{II} - P_{II}^{-1}$ have nonpositive offdiagonal entries and zero (generalized) row-sum, i.e., are symmetric M-matrices. A more formal proof of this result is given in Theorem 3.1 below, where we address the general case of factorizations defined by (3.1) with K satisfying

$$\bigl(K(Fe)\bigr)_i = \bigl(P^{-1}(Fe)\bigr)_i \quad \text{for all } i \text{ such that } (Fe)_i \neq 0. \tag{3.6}$$

Before stating this theorem, we briefly discuss why adding this rule for the computation of the diagonal entries of K should improve MBILU preconditioning.

First, note that if $\beta * (E\operatorname{diag}(K)F)$ is purely diagonal, this new method leads to exactly the same preconditioner as MBILU; indeed the diagonal of K influences then only the diagonal of P which, via Ω, is anyway recomputed so as to satisfy the row-sum criterion (3.2). This in particular arises when applying block methods with line partitioning to five point matrices. Hence, for most previous applications of block ILU, IMBILU reduces to MBILU.

In more general cases however, the fill entries in the successive Schur complements will be larger in absolute value with IMBILU since the new rule implies an increase of the diagonal entries of K. Therefore, IMBILU factorization of A corresponds to MBILU factorization of some matrix A' which has smaller (larger in absolute value) offdiagonal entries, but same row-sum since (3.2) holds in both cases. Thus $A' = A + \Delta$, where $\mathrm{offdiag}(\Delta) \leqslant 0$ and $\Delta e = 0$, showing that applying IMBILU amounts to applying MBILU to the matrix to which one has added a nonnegative definite perturbation matrix.

Improving the largest eigenvalue behavior associated with modified factorizations by factorizing a perturbed matrix is a common technique (see [20] for an example in the block case) although in general only purely diagonal perturbation matrices are considered (see, however, [12]). In most cases, some benefit is guaranteed on the largest eigenvalues, at the price of some decrease of the lowest eigenvalues. Here, the philosophy is different: we cannot estimate to what extent the largest eigenvalues of the preconditioned system will be decreased by using IMBILU; numerical experiments in Section 5 show that this ranges from zero to something very dramatic. However, this improvement comes essentially for free since both methods have same lowest eigenvalue.

Note that approximate inverses satisfying a generalized row-sum criterion were already considered a.o. in [4] but, to our knowledge, not in the context of block incomplete factorizations of the form (2.1).

For completeness, we also prove in Theorem 3.1 the existence of the preconditioner. It means that we do not assume the positive definiteness of P, but rather show that it is a consequence of the relation (3.1) that defines it. For consistency, we have then to introduce some generalized inverse P^+ of P (satisfying $PP^+P = P$ [13]), and write P^+ instead of P^{-1} in the assumptions of the theorem; how P^+ is determined is however unimportant since the regularity of P, which implies $P^+ = P^{-1}$, is finally proved.

We would like nevertheless to point out a little exception to that rule. When A is a singular weakly diagonally dominant symmetric M-matrix, $\mathcal{N} = \mathrm{span}\{e\}$ [14] and the row-sum criterion (3.2) implies that the preconditioner is also singular. The existence analysis proves however then that it is positive semidefinite with $\mathcal{N}(B) = \mathcal{N}(A)$, which is sufficient for making it a valid preconditioner [21,25]. Moreover, even in such cases $B = (P - E)P^+(P - F) = P - E - F + EP^+F$ does not depend on the choice of the generalized inverse since only P_{MM} is singular whereas the entries in the last diagonal block of P^+ do actually not influence the entries in EP^+F. We refer to [21] for more details about the singular case; see also Section 5 for some practical considerations.

Finally, let us recall that the existence of MBILU factorizations cannot be proved without additional assumption involving the nonzero pattern and/or strict diagonal dominance, see [22] where necessary and sufficient conditions are given. We therefore add the requirement, following the approach in [21,22], that, for all $1 \leqslant I \leqslant M - 1$ the block $A_{12}^{(I)}$ (or equivalently, some of the blocks $F_{IJ},\ J > I$) contains at least one nonzero entry. Besides, it is also assumed in [21,22] that the diagonal blocks of A are irreducible (or 1×1); in fact, only the diagonal blocks of P are to be such and, as an alternate possibility, existence can still be proved if P_{II} is reducible for some I but $(Fe)_I > 0$. Because of this slight extension, we give a very condensed proof, referring to [21,22] for a more thorough development of the arguments.

Theorem 3.1. *Let A be an irreducible (weakly) diagonally dominant symmetric M-matrix, β a symmetric matrix such that $0 \leqslant \beta_{ij} \leqslant 1$ for all i, j, K a nonnegative symmetric block diagonal matrix and Ω a diagonal matrix.*

Let P be the block diagonal matrix and F the strictly block upper triangular matrix satisfying

$$P - E - F = A - \beta * (EKF) - \Omega,$$

where $E = F^{\mathrm{T}}$.

Let

$$B = (P - E)P^+(P - F),$$

where P^+ *is a generalized inverse of* P, *and let* $\overline{\Omega}$ *be the diagonal matrix such that*

$$\overline{\Omega}e = \bigl(EP^+F - \beta * (EKF)\bigr)e.$$

(1) *Assume that for all* $1 \leqslant I \leqslant M-1$, *there is at least one block* F_{IJ}, $I < J \leqslant M$, *which is nonzero and that, for all* $1 \leqslant I \leqslant M$, *either* P_{II} *is a* 1×1 *block or it is irreducible or* $(Fe)_I > 0$.
If $\Omega \leqslant \overline{\Omega}$, *then* P *is a symmetric* M*-matrix, with* P_{II} *regular* (*positive definite*) *for* $I = 1, \ldots, M-1$ *and* P_{MM} *also regular* (*positive definite*), *except if* A *is singular and* $\Omega = \overline{\Omega}$, *in which case* P_{MM} *is singular* (*positive semidefinite*) *and* $\mathcal{N}(B) = \mathcal{N}(A)$.

(2) *If* $\Omega \geqslant \overline{\Omega}$ *and, for* $I = 1, \ldots, M-1$, P_{II} *is regular and* K_{II} *such that*

$$\bigl(K(Fe)\bigr)_i = \bigl(P^{-1}(Fe)\bigr)_i \quad \textit{for all } i \textit{ such that } (Fe)_i \neq 0, \tag{3.7}$$

$$\operatorname{offdiag}(K_{II}) \leqslant \operatorname{offdiag}\bigl(P_{II}^{-1}\bigr), \tag{3.8}$$

then $A - B$ *is nonnegative definite.*

Proof. (1) Obviously, $\operatorname{offdiag}(P) \leqslant 0$ and $F \geqslant 0$ while

$$\bigl((P-F)e\bigr)_I = (Ae)_I + \sum_{S<I} E_{IS}\bigl((I - P^+F)e\bigr)_S + \bigl((\overline{\Omega} - \Omega)e\bigr)_I$$

implies $(P - F)e \geqslant Ae \geqslant 0$ by the classical induction argument (see, e.g., [27]). P is therefore an M-matrix. Moreover, $Pe \geqslant Fe$ proves that P_{II} is regular when either $(Fe)_I > 0$ or P_{II} is irreducible (or 1×1) with $((Fe)_I)_i \neq 0$ for some i [14].

On the other hand, if for any $I < M$, $(((P-F)e)_I)_i > 0$ for some i, one will have necessarily $(((P-F)e)_J)_j > 0$ for some $J > I$, $j \in L_J$, because (a) $(Fe)_I > 0$ means that there is one nonzero entry in row i of F, say entry $(F_{IJ})_{ij}$, whence

$$\bigl(E_{JI}P_{II}^{-1}\bigl((P-F)e\bigr)_I\bigr)_j \geqslant (F_{IJ})_{ij}(P_{II})_{ii}^{-1}\bigl(\bigl((P-F)e\bigr)_I\bigr)_i > 0,$$

and (b) P_{II} irreducible $\Rightarrow P_{II}^{-1} > 0$, whence $((Fe)_I)_i < (P_{II}e_i)_i$ for some i implies $P_{II}^{-1}(Fe)_I < e_I$ and therefore $E_{JI}(e_I - P_{II}^{-1}(Fe)_I)$ is nonzero for at least some $J > I$ by our assumptions. It then follows that $P_{MM}e_M = 0$ may occur only if $((P-F)e)_I = 0$ for all I, that is if $Ae = 0$ and $\Omega = \overline{\Omega}$. Since P_{MM} is irreducible by assumption, this proves the regularity of P_{MM} in any other case, while $\mathcal{N}(B) = \mathcal{N}(A)$ follows otherwise from $Be = 0$ on the one hand and $\dim(\mathcal{N}(B)) = \dim(\mathcal{N}(P_{MM}))$ (see the proof of [21, Theorem 3.2]) with $\dim(\mathcal{N}(P_{MM})) = 1$ because P_{MM} is irreducible on the other hand.

(2) $A - B = E(K - P^+)F + \Omega - (\varepsilon - \beta) * (EKF)$.

Let G be the symmetric block diagonal matrix such that $G_{ij} = 0$ if either $(Fe)_i = 0$ or $(Fe)_j = 0$ and $G_{ij} = (K - P^{-1})_{ij}$ otherwise (which in particular means that $G_{MM} = 0$). Since $(Fe)_i = 0$ implies $E_{ri} = 0$ for all r and $(Fe)_j = 0$ implies $F_{js} = 0$ for all s, it is readily seen that $E(K - P^+)F = EGF$. Now, (3.8) implies $\operatorname{offdiag}(G) \leqslant 0$ and also $\operatorname{offdiag}(G_{II}) \geqslant \operatorname{offdiag}(K_{II} - P_{II}^{-1})$ for all $I < M$. Therefore, (3.7) ensures that $(G_{II}(Fe)_I)_i \geqslant 0$ for all i such that $((Fe)_I)_i \neq 0$ (since then $(G_{II})_{ii} = (K_{II} - P_{II}^{-1})_{ii}$). Hence, G is a symmetric M-matrix, showing that $E(K - P^+)F$ is nonnegative definite. On the other hand, (3.7) readily implies $EKFe = EP^{-1}Fe$ so that $\Omega \geqslant \overline{\Omega}$ may be rewritten $\Omega e \geqslant ((\varepsilon - \beta) * (EKF))e$, showing, together with $\varepsilon - \beta \geqslant 0$, $EKF \geqslant 0$ that $\Omega - (\varepsilon - \beta) * (EKF)$ is also nonnegative definite, whence the required result. □

Remark. Eq. (3.6) fixes only those of the diagonal entries K_{ii} for which $(Fe)_i \neq 0$, but the other ones do actually not influence the computation since the corresponding row in F has no nonzero element.

4. Generalized tridiagonal matrices

As already stated, we say that a symmetric matrix $T = (T_{ij})$ is generalized tridiagonal if, for all i, there is at most one $j > i$ such that $T_{ij} \neq 0$. As standard tridiagonal matrices, they can be factorized without fill-in, i.e., we may write

$$T = \left(I - G^{\mathrm{T}}\right) Q^{-1} (I - G), \tag{4.1}$$

where Q is diagonal and G strictly upper triangular with at most one nonzero per row.

We shall see that IMBILU works fine in the context of Algorithm 2.2, that is with diagonal approximation of the inverse block pivots entirely determined by (2.5). However, for completeness, and to allow comparison with MBILU, we need to extend to generalized tridiagonal matrices algorithms that compute, for P_{II} tridiagonal, K_{II} equal to the tridiagonal band portion of P_{II}^{-1}. We find that the most reasonable extension consists in letting $(K_{II})_{ij} = (P_{II}^{-1})_{ij}$ whenever $(P_{II})_{ij} \neq 0$, and $(K_{II})_{ij} = 0$ otherwise. If P_{II} is permutable to a tridiagonal matrix, this gives the same result as one would obtain with the standard algorithm after explicit permutation, but without requiring this additional step.

Now, computing such an approximation for matrices of the form (4.1) is actually easy. Indeed, the method in [31] (see also [4, Section 8.1.2]) generalizes straightforwardly: from (4.1), we deduce

$$T^{-1} = (I - G)^{-1} Q + T^{-1} G^{\mathrm{T}},$$

and, since T^{-1} is symmetric and $(I - G)^{-1}$ upper triangular with diagonal unity, one easily checks that

$$\begin{array}{ll} \left(T^{-1}\right)_{ii} = Q_{ii} & \text{if } G_{ik} = 0 \text{ for all } k, \\ \left\{ \begin{array}{l} \left(T^{-1}\right)_{ij} = \left(T^{-1}\right)_{ji} = \left(T^{-1}\right)_{jj} G_{ij} \\ \left(T^{-1}\right)_{ii} = Q_{ii} + \left(T^{-1}\right)_{ij} G_{ij} \end{array} \right. & \text{if } G_{ik} = 0 \text{ for all } k \neq j. \end{array}$$

All requested entries can then be computed from theses relations by considering the nodes in decreasing order.

Note for completeness that generalized tridiagonal matrices are treediagonal matrices, defined in [18] as matrices whose associated graph is a tree. The converse is not true: some treediagonal matrices are not generalized tridiagonal, although all are permutable to a generalized tridiagonal form.

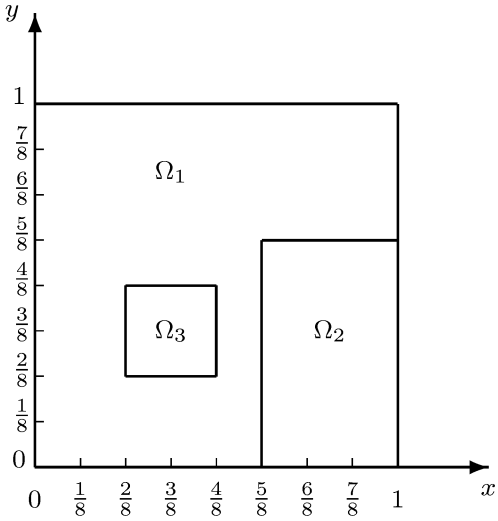
5. Numerical results

We report here the results of numerical experiments performed on the linear system resulting from the five point finite difference approximation with uniform grid of mesh size h of the PDE

$$\begin{array}{l} -\partial_x\, p(x, y)\, \partial_x\, u(x, y) - \partial_y q(x, y) \partial_y u(x, y) = f(x, y) \quad \text{in } \Omega = (0, 1) \times (0, 1), \\ \left\{ \begin{array}{ll} u(x, y) = 0 & \text{on } \Gamma_0, \\ \dfrac{\partial u(x, y)}{\partial n} = 0 & \text{on } \Gamma_1 = \partial\Omega \backslash \Gamma_0 \end{array} \right. \end{array}$$

with the following specification, where d is an anisotropy parameter which we let vary from 10^{-3} to 10^{3}.

Problem 1. $p = d, q = 1, f = 1$ and $\Gamma_0 = \partial\Omega$.



region	p	q	f
Ω_1	1	1	1
Ω_2	1	d	-3
Ω_3	d	1	0

Fig. 3. Specification of the PDE coefficients for Problem 3.

Problem 2.

$$p=\begin{cases}100\cdot d & \text{in } \left(\frac{1}{4},\frac{3}{4}\right)\times\left(\frac{1}{4},\frac{3}{4}\right),\\ d & \text{elsewhere,}\end{cases}\qquad q=\begin{cases}100 & \text{in } \left(\frac{1}{4},\frac{3}{4}\right)\times\left(\frac{1}{4},\frac{3}{4}\right),\\ 1 & \text{elsewhere,}\end{cases}$$

$$f=100 \text{ where } q=100 \text{ and } f=0 \text{ elsewhere,}\qquad \Gamma_0=\{(x,y)\mid 0\leqslant x\leqslant 1,\ y=0\}.$$

Problem 3. $\Gamma_0=\emptyset$ ($\Gamma_1=\partial\Omega$) and p, q and f as specified in Fig. 3.

Note that, for $d=1$, Problem 1 reduces to the model Dirichlet problem, Problem 2 to the problem in [33] and Problem 3 to the model Neumann problem; Problem 3 is inspired from the Stone problem [30] and borrowed from [19,22].

The considered preconditioning techniques are:

- MILU(rrb): preconditioner computed by Algorithm 2.1.
- IMBILU(rrb): preconditioner computed by Algorithm 2.2; K_{II} generalized tridiagonal with offdiagonal part computed as indicated in Section 4 was also tested but not found significantly better.
- MBILU(rrb): preconditioner computed according to the method indicated in the discussion preceding Algorithm 2.2, with $K_{II}\leqslant P_{II}^{-1}$ computed as indicated in Section 4, $S^{(I)}$ defined by (2.3), (2.4) and $A^{(I+1)}$ equal to the matrix with same row-sum as $S^{(I)}$ and same sparsity pattern as $A_{22}^{(I)}-A_{21}^{(I)}\operatorname{diag}(K_{II})A_{12}^{(I)}$; $K_{II}=\operatorname{diag}(P_{II}^{-1})$ was also tested but found still less efficient.
- MBILU(line): standard MBILU for line partitioning (same as MINV(1) in [17]).
- DMBILU*(line): same as above but with “dynamic perturbations” added as recommended in [19] for (partially) anisotropic problems.

For the preconditioners based on recursive red–black partitioning, we let $M=\log_2 h^{-1}$ to guarantee that the factorization cost is similar to that for MBILU(line) and DMBILU*(line) [16,28].

Note that Problem 3 leads to a singular but compatible system. Except possibly DMBILU*, the preconditioners above are then singular too, but admissible because they have same kernel as A [25]. In practice only the last pivot of the pointwise factorization of the last block is zero and it suffices to exchange it for an arbitrary positive value to use these preconditioners trouble free as regular ones [21]. Besides, we projected all direction vectors onto the range of A to improve the stability [26] at the price of 2 more flops per unknown and per iteration.

The results are reported in Fig. 4. For the three test problems with $h^{-1} = 512$, we have graphically represented in function of d the number of floating point operations (flops) per unknown to reduce the two-norm of the residual by 10^{-5} when using the conjugate gradient algorithm with zero vector as initial approximation. The preprocessing cost is not included, but, for all the considered methods it remains fairly small compared with the solution cost. For instance, considering IMBILU(rrb), step I of the Algorithm 2.2 requires about 38 flops plus 2 divisions per node in block 1 to form and factorize P_{II}, and to compute K_{II} and the approximate Schur complement. Hence, since, with the used rule for M, the cost of factorizing P_{MM} is about n flops, the global preprocessing cost is only about 39 flops plus 2 divisions per unknown.

Looking at the results, the reference methods MILU(rrb), MBILU(line) and DMBILU*(line) behave as expected whereas the pictures are self explanatory with respect to MBILU(rrb) and IMBILU(rrb): MBILU(rrb) hardly improves anything over MILU(rrb), while IMBILU(rrb) combines the efficiency of MILU(rrb) in the isotropic cases with the robustness of standard block methods with respect to anisotropy. Note that MILU(rrb) remains better than its block extension IMBILU(rrb) for isotropic to moderately anisotropic problems essentially because both methods present then a similar conditioning, but iterations are somewhat cheaper with MILU(rrb).

Algorithm 2.2 outmatches thus previous incomplete factorization based preconditioners. In Table 1, we report the associated condition number κ for the three test problems and h^{-1} ranging from 64 to 512. We also give the exponent μ from the assumed relationship $\kappa = ch^{-\mu}$, estimated from the largest two problems data (say, $h^{-1} = 256$ and $h^{-1} = 512$). The first observation is that the condition numbers are fairly small for such a cheap preconditioning. For $d = 1$ (isotropic case) the condition number is similar to the one associated with the preconditioner of Algorithm 2.1 (MILU(rrb)) and fits well $\kappa = 0.8 \cdot h^{-0.32}$. For other values of d, the increase of κ with h^{-1} is more erratic. However, if one considers the maximum over d of the observed values, it seems that κ presents in the end a similar nice asymptotic behavior.

Remark. For completeness, we also performed some tests with standard (unmodified) ILU preconditioners in both pointwise [24] and blockwise [5,10,17] versions (using in the latter case a line partitioning). For the model Problem 1, these methods require respectively more than 7000 and more than 3800 flops in the isotropic case ($d = 1$). When $d = 10^3$ ($d = 10^{-3}$), these flop counts reduce to respectively 1600 (1600) and 700 (600). The relative behavior of unmodified methods improves thus in presence of anisotropy, but they nevertheless remain outmatched by DMBILU*(line), which we were therefore right to take as reference (block) incomplete factorization preconditioner. This is confirmed by a further test ran on the more difficult Problem 2: then, with the unmodified block ILU preconditioner, the solution of the problem requires more than 6400 flops when $d = 10^3$, more than 7400 when $d = 1$, and more than 2100 when $d = 10^{-3}$.

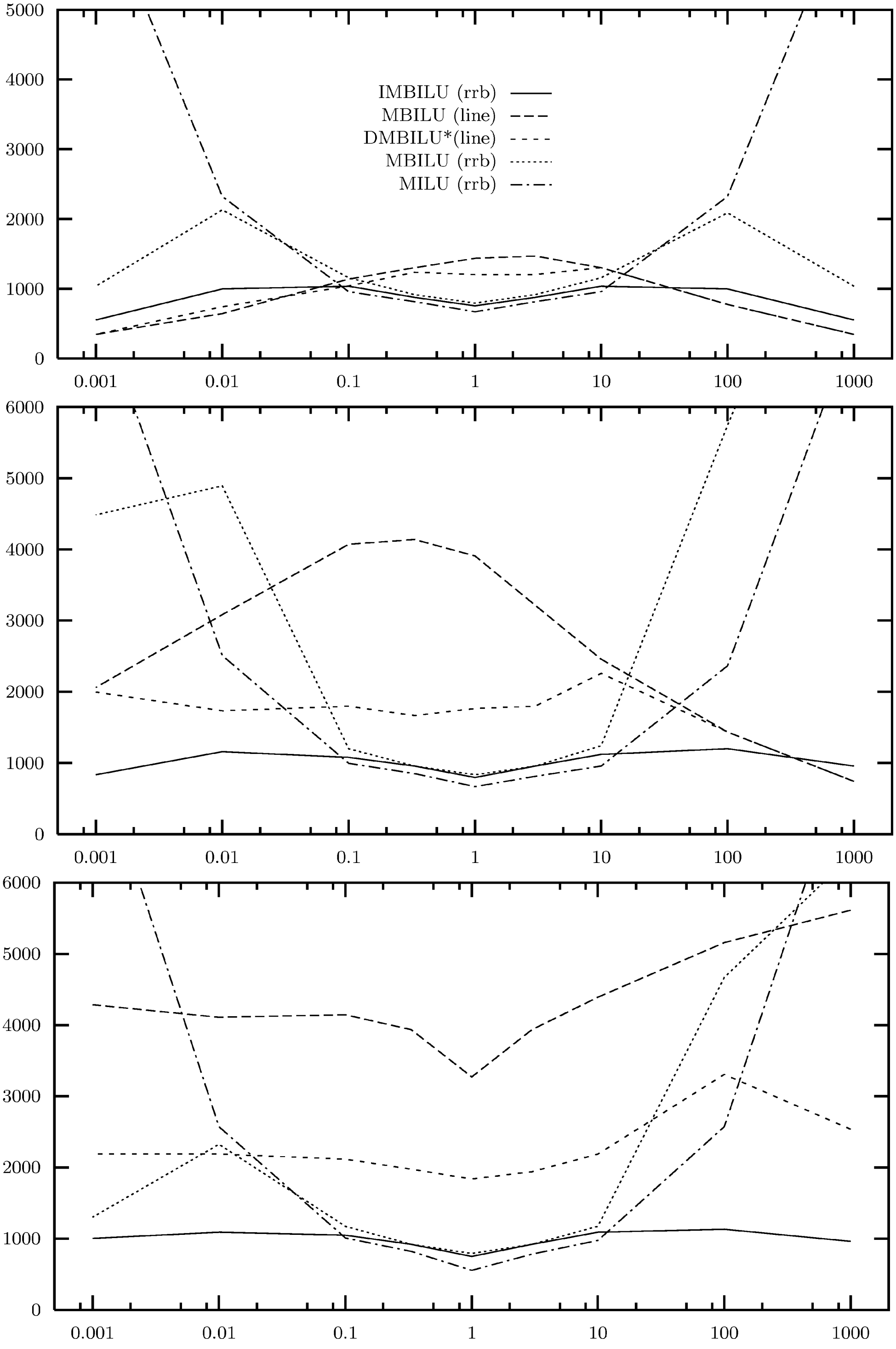


Fig. 4. Number of flops per unknown to achieve solution in function of d.

Table 1
Condition numbers for IMBILU(rrb)

d	0.001	0.01	0.1	1	10	100	1000	$\max_d(\kappa)$
h^{-1}	Condition numbers for Problem 1							
64	1.05	1.56	3.16	2.80	3.16	1.56	1.05	3.16
128	1.23	2.78	5.11	3.62	5.11	2.78	1.23	5.11
256	1.89	5.80	8.37	4.57	8.37	5.80	1.89	8.37
512	3.67	11.1	10.7	5.71	10.7	11.1	3.67	11.1
μ	0.96	0.94	0.36	0.32	0.36	0.94	0.96	0.41
h^{-1}	Condition numbers for Problem 2							
64	1.69	2.64	4.46	2.95	4.48	3.91	1.78	4.48
128	2.22	4.87	5.97	3.74	5.96	5.93	3.00	5.97
256	3.64	9.84	10.2	4.71	10.1	11.2	6.76	11.2
512	7.85	13.4	11.8	5.86	11.7	13.5	11.4	13.5
μ	1.11	0.45	0.20	0.32	0.21	0.28	0.75	0.27
h^{-1}	Condition numbers for Problem 3							
64	3.97	3.89	4.37	3.04	4.73	3.80	3.75	4.73
128	4.94	5.48	5.74	3.78	5.95	5.32	4.69	5.95
256	7.35	11.2	9.19	4.68	9.56	10.8	6.20	11.2
512	9.87	13.8	11.0	5.78	11.1	13.8	8.60	13.8
μ	0.42	0.29	0.25	0.31	0.22	0.35	0.47	0.30

Acknowledgements

The general idea that preconditioners based on recursive red–black orderings can be improved, by combining them with block incomplete factorization techniques, arose from private discussions with Owe Axelsson and Magolu monga-Made.

Magolu monga-Made is also acknowledged for fruitful discussions and useful comments.

References

[1] O. Axelsson, A general incomplete block-matrix factorization method, *Linear Algebra Appl.* 74 (1986) 179–190.

[2] O. Axelsson, The method of diagonal compensation of reduced matrix entries and multilevel iteration, *J. Comput. Appl. Math.* 38 (1991) 31–43.

[3] O. Axelsson, On algebraic multilevel iterations for methods for selfadjoint elliptic problems with anisotropy, in: Fascicolo Speciale, Numerical Methods, *Rend. Sem. Mat. Univers. Politech. Torino*, 1991, pp. 31–61.

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

[5] O. Axelsson, S. Brinkkemper, V.P. Il'in, On some versions of incomplete block-matrix factorization iterative methods, *Linear Algebra Appl.* 84 (1984) 3–15.

[6] O. Axelsson, V. Eijkhout, The nested recursive two level factorization for nine-point difference matrices, Technical Report 8936, Department of Mathematics, Catholic University, Nijmegen, The Netherlands, 1989.

[7] O. Axelsson, V. Eijkhout, Analysis of recursive 5-point/9-point factorization method, in: O. Axelsson, L. Kolotilina, eds., *Preconditioned Conjugate Gradient Methods,* Lectures Notes in Mathematics, Vol. 1457 (Springer, Berlin, 1990) 154–173.

[8] O. Axelsson, V. Eijkhout, The nested recursive two level factorization for nine-point difference matrices, *SIAM J. Sci. Statist. Comput.* 12 (1991) 1373–1400.

[9] O. Axelsson, M. Neytcheva, Algebraic multilevel iterations for Stieltjes matrices, *Numer. Linear Algebra Appl.* 1 (1994) 213–236.

[10] R. Beauwens, M. Ben Bouzid, On sparse block factorization iterative methods, *SIAM J. Numer. Anal.* 24 (1987) 1066–1076.

[11] R. Beauwens, M. Ben Bouzid, Existence and conditioning properties of sparse approximate block factorizations, *SIAM J. Numer. Anal.* 25 (1988) 941–956.

[12] R. Beauwens, B. Tombuyses, Graph perturbations, *Appl. Numer. Math.* 16 (1994) 251–264.

[13] A. Ben Israel, T.N.E. Greville, *Generalized Inverses: Theory and Applications* (Wiley, New York, 1974).

[14] A. Berman, R.J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences* (Academic Press, New York, 1979).

[15] E. Botta, A. van der Ploeg, Preconditioning techniques for matrices with arbitrary sparsity patterns, Preprint, Department of Mathematics, University of Groningen, The Netherlands, 1995.

[16] C.W. Brand, An incomplete factorization preconditioning using repeated red–black ordering, *Numer. Math.* 61 (1992) 433–454.

[17] P. Concus, G.H. Golub, G. Meurant, Block preconditioning for the conjugate gradient method, *SIAM J. Sci. Statist. Comput.* 6 (1985) 220–252.

[18] D.J. Klein, Treediagonal matrices and their inverses, *Linear Algebra Appl.* 42 (1982) 109–117.

[19] M. Magolu, Taking advantage of the potentialities of dynamically modified block incomplete factorizations, *SIAM J. Sci. Comput.* 19 (1998) 1083–1108.

[20] M.M. Magolu, Modified block-approximate factorization strategies, *Numer. Math.* 61 (1992) 91–110.

[21] M.M. Magolu, Sparse block approximate factorizations for singular problems, in: R. Beauwens, P. de Groen, eds., *Iterative Methods in Linear Algebra* (North-Holland, Amsterdam, 1992) 519–529.

[22] M.M. Magolu, Ordering strategies for modified block incomplete factorization, *SIAM J. Sci. Comput.* 16 (1995) 378–399.

[23] S. Margenov, P. Vassilevski, Algebraic multilevel preconditioning of anisotropic elliptic problems, *SIAM J. Sci. Stat. Comput.* 15 (1994) 1026–1037.

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

[25] Y. Notay, Polynomial acceleration of iterative schemes associated with subproper splittings, *J. Comput. Appl. Math.* 24 (1988) 153–167.

[26] Y. Notay, Solving positive (semi)definite linear systems by preconditioned iterative methods, in: O. Axelsson, L. Kolotilina, eds., *Preconditioned Conjugate Gradient Methods*, Lectures Notes in Mathematics, Vol. 1457 (Springer, Berlin, 1990) 105–125.

[27] Y. Notay, Conditioning analysis of modified block incomplete factorizations, *Linear Algebra Appl.* 154–156 (1991) 711–722.

[28] Y. Notay, Z. Ould Amar, A nearly optimal preconditioning based on recursive red–black orderings, *Numer. Linear Algebra Appl.* 4 (1997) 369–391.
[29] Y. Saad, ILUM: a parallel multi-elimination ILU preconditioner for general sparse matrices, *SIAM J. Sci. Statist. Comput.* 17 (1996) 830–847.
[30] H.L. Stone, Iterative solution of implicit approximation of multidimensional partial differential equations, *SIAM J. Numer. Anal.* 5 (1968) 530–558.
[31] K. Takahishi, J. Fagan, M. Chen, Formation of a sparse bus impedance matrix and its application to short circuit study, in: *8th PICA Conference Proceedings*, Minneapolis, 1973, pp. 63–69.
[32] A. van der Ploeg, E. Botta, F. Wubs, Nested grids ILU-decomposition (NGILU), *J. Comput. Appl. Math.* 66 (1996) 515–526.
[33] H.A. van der Vorst, The convergence behaviour of preconditioned CG and CG-S, in: O. Axelsson, L. Kolotilina, eds., *Preconditioned Conjugate Gradient Methods*, Lectures Notes in Mathematics, Vol. 1457 (Springer, Berlin, 1990) 126–136.