Electronic Transactions on Numerical Analysis.
Volume 37, pp. 23-40, 2010.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

BLOCK APPROXIMATE INVERSE PRECONDITIONERS FOR SPARSE NONSYMMETRIC LINEAR SYSTEMS*

J. CERDÁN†, T. FARAJ‡, N. MALLA†, J. MARÍN†, AND J. MAS†

Abstract. In this paper block approximate inverse preconditioners to solve sparse nonsymmetric linear systems with iterative Krylov subspace methods are studied. The computation of the preconditioners involves consecutive updates of variable rank of an initial and nonsingular matrix A_0 and the application of the Sherman-Morrison-Woodbury formula to compute an approximate inverse decomposition of the updated matrices. Therefore, they are generalizations of the preconditioner presented in Bru et al. [SIAM J. Sci. Comput., 25 (2003), pp. 701–715]. The stability of the preconditioners is studied and it is shown that their computation is breakdown-free for H-matrices. To test the performance the results of numerical experiments obtained for a representative set of matrices are presented.

Key words. approximate inverse preconditioners, variable rank updates, block algorithms, Krylov iterative methods, Sherman-Morrison-Woodbury formula

AMS subject classifications. 65F10, 65F35, 65F50

1. Introduction.

In this paper we consider the solution of nonsingular linear systems

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

by preconditioned iterations. We assume the matrix $A \in \mathbb{R}^{n\times n}$ to be sparse and nonsymmetric. For large values of n an approximate solution for (1.1) is frequently obtained by means of iterative Krylov subspace methods. In practice, to accelerate the convergence of these methods either left, right or two-sided preconditioning is applied [23]. For left preconditioning the linear system to solve is

$$MAx = Mb,$$

where the matrix M is the preconditioner.

Usually the matrix M is chosen in such a way that the preconditioned matrix MA is close to the identity I_n in some sense. For instance, the condition number is small and/or the eigenvalues are clustered away from the origin. In general, the more clustered the eigenvalues, the faster the convergence rate. Another desired situation is that the preconditioner should be easy to compute and the cost of the preconditioning step should be of the same order of a matrix-vector product with the coefficient matrix A.

In the last years several preconditioning techniques have been proposed. Roughly speaking they can be grouped in two classes: *implicit preconditioners* and *explicit preconditioners*. Preconditioners of the first class typically compute incomplete factorizations of A, such as incomplete LU, and therefore the preconditioning step is done by solving two triangular linear systems; see for example [18, 19, 22, 23]. By contrast the second class of preconditioners compute and store a sparse approximation of the inverse of A and the preconditioning step is done by a matrix-vector product; see [8, 13, 16, 17]. Since this operation is easy to implement on parallel and vector computers, approximate inverse preconditioners are attractive for parallel computations. In addition, some authors argue that approximate inverse preconditioners

*Received December 10, 2004. Accepted for publication May 21, 2009. Published online on February 28, 2010. Recommended by M. Benzi. Supported by Spanish Grant MTM2007-64477.

†Institut de Matemàtica Aplicada, Universitat Politècnica de València, 46022 València, Spain ({jcerdan,namalmar,jmarinma,jmasm}@imm.upv.es).

‡Dept. Matemàtica Aplicada, Universitat Politècnica de Valencia, 46022, València, Spain (tafael@mat.upv.es).

ETNA
Kent State University
http://etna.math.kent.edu

are more robust than implicit ones [2]. For a comparative study of some of these techniques we refer to [7]. We focus in this paper on sparse approximate inverse preconditioners.

In [11] the authors present a new algorithm based on the Sherman-Morrison formula to compute an inverse decomposition of a nonsymmetric matrix. Given two sets of vectors $\{x_k\}_{k=1}^n$ and $\{y_k\}_{k=1}^n$ in $\mathbb{R}^n$, and a nonsingular matrix A_0 such that $A = A_0 + \sum_{k=1}^n x_k y_k^T$, the algorithm computes a factorization of the matrix $A_0^{-1} - A^{-1}$ of the form $UT^{-1}V^T$ from now on called ISM decomposition. The particular case $A_0 = sI_n$, where I_n is the identity matrix and s is a positive scalar factor, was studied and it was shown that the approximate computation of this decomposition is breakdown-free when A is an M-matrix. We will show that this is also true for the wider class of H-matrices. The ISM decomposition is closely related to the LU factorization as can be seen in the proof of Lemmas 3.2 and 3.5. This fact has been used in [10] to obtain direct and inverse factors of the Cholesky factorization of a symmetric positive definite matrix. This approach differs from AINV, which uses a biconjugation process or SPAI, based on the minimization of the Frobenius norm of $I - AM$.

Once an approximate ISM decomposition has been computed it can be used as an approximate inverse preconditioner as has been done in [11]. It was observed that compared to AINV [8] both performed similarly. In this paper we present results for the block case showing that this approach is able to solve more problems than its pointwise counterpart.

On the other hand, exploiting faster speeds of level 3 BLAS block algorithms is becoming increasingly popular in matrix computations. Since they operate on blocks or submatrices of the original matrix, they are well suited for modern high performance computers. Furthermore, certain problems have a natural block structure that should be exploited to gain robustness. There are a number of papers on block preconditioners; see for instance [3, 6, 9, 12] and the references therein. In all cases an improved efficiency with respect to non-blocking algorithms was observed.

These considerations motivate the present study. We generalize the ISM decomposition by applying successive updates of variable rank which leads to a block-form algorithm. Therefore, the new algorithm is based on the Sherman-Morrison-Woodbury formula [27], which states that the inverse of the matrix $A + XY^T$ is given by

$$A^{-1} - A^{-1}X(I + Y^T A^{-1} X)^{-1} Y^T A^{-1}, \tag{1.2}$$

provided that the matrices $A \in \mathbb{R}^{n\times n}$ and $I + Y^T A^{-1} X$ are nonsingular and $X, Y \in \mathbb{R}^{n\times m}$. The rank of the updates, and hence the size of the blocks, can be chosen in different ways: looking for the particular structure for structured matrices, applying an algorithm to find the block structure [24] and finally by imposing an artificial block structure. The approximate computation of the block ISM decomposition is then used as an inverse block preconditioner.

This paper is organized as follows. Section 2 presents an expression for the block ISM decomposition of a general matrix A using the Sherman-Morrison-Woodbury formula (1.2) which generalizes the one obtained in [11]. Then, in Section 3 this expression is used to obtain block approximate inverse preconditioners based on different choices of the initial matrix A_0 and to show how they relate to each other. Our findings indicate that these preconditioners can be computed without breakdowns for H-matrices. Therefore, since the preconditioner proposed in [11] is a particular case of one of the preconditioners proposed here, we also prove that its computation is breakdown-free for H-matrices. In order to evaluate the performance of the preconditioners, the results of the numerical experiments for a representative set of matrices are presented in Section 4. Finally, the main conclusions are presented in Section 5.

Throughout the paper the following notation will be used. Given two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, we denote $A \geq B$ when $a_{ij} \geq b_{ij}$. A similar convention is used for $\leq$.

ETNA
Kent State University
http://etna.math.kent.edu

Likewise, $|A| = [|a_{ij}|]$. A matrix A is a nonsingular M-matrix if $a_{ij} \leq 0$ for all $i \neq j$ and it is monotone, i.e., $A^{-1} \geq O$. For a given matrix A one can associate its comparison matrix $\mathcal{M}(A) = [\alpha_{ij}]$, where

$$\alpha_{ii} = |a_{ii}|, \text{ and } \alpha_{ij} = -|a_{ij}| \text{ for } i \neq j.$$

The matrix A is an H-matrix if its comparison matrix $\mathcal{M}(A)$ is an M-matrix.

We conclude this section with some well-known properties of M- and H-matrices that will be used later. If $B \geq A$ with $b_{ij} \leq 0$ for $i \neq j$ and A is an M-matrix, then B is also an M-matrix. Moreover, $A^{-1} \geq B^{-1}$ [26]. If A is an H-matrix, then $|A^{-1}| \leq \mathcal{M}(A)^{-1}$ [20].

2. Block ISM decomposition. Let $A_0 \in \mathbb{R}^{n\times n}$ be a nonsingular matrix, and let $X_k, Y_k \in \mathbb{R}^{n\times m_k}$, $k = 1, \ldots, p$, be two sets of rectangular matrices such that,

$$A = A_0 + \sum_{k=1}^{p} X_k Y_k^T. \tag{2.1}$$

Assume that the matrices $T_k = I_{m_k} + Y_k^T A_{k-1}^{-1} X_k$, $k = 1, \ldots, p$, are nonsingular, where $A_{k-1} = A_0 + \sum_{i=1}^{k-1} X_i Y_i^T$, i.e., a partial sum of (2.1), and I_{m_k} denotes the identity matrix of size $m_k \times m_k$. ¿From (1.2) the inverse of A_k is given by

$$A_k^{-1} = A_{k-1}^{-1} - A_{k-1}^{-1} X_k T_k^{-1} Y_k^T A_{k-1}^{-1}, \quad k = 1, \ldots, p. \tag{2.2}$$

Since $A_p^{-1} = A^{-1}$, applying (2.2) recursively one has

$$A^{-1} = A_0^{-1} - \sum_{k=1}^{p} A_{k-1}^{-1} X_k T_k^{-1} Y_k^T A_{k-1}^{-1},$$

which can be written in matrix notation as

$$A^{-1} = A_0^{-1} - \Phi T^{-1} \Psi^T, \tag{2.3}$$

where

$$\Phi = \begin{bmatrix} A_0^{-1}X_1 & A_1^{-1}X_2 & \cdots & A_{p-1}^{-1}X_p \end{bmatrix},$$

$$T^{-1} = \begin{bmatrix} T_1^{-1} & & & \\ & T_2^{-1} & & \\ & & \ddots & \\ & & & T_p^{-1} \end{bmatrix} \quad \text{and} \quad \Psi^T = \begin{bmatrix} Y_1^T A_0^{-1} \\ Y_2^T A_1^{-1} \\ \vdots \\ Y_p^T A_{p-1}^{-1} \end{bmatrix}.$$

To avoid having to compute the matrices A_k, we can define from X_k and Y_k, for $k = 1, \ldots, p$, two new sets of matrices U_k and V_k as in (2.4) and (2.5). The following result is a generalization of [11, Theorem 2.1].

THEOREM 2.1. *Let A and A_0 be two nonsingular matrices, and let $\{X_k\}_{k=1}^p$ and $\{Y_k\}_{k=1}^p$ be two sets of matrices such that condition (2.1) is satisfied. In addition suppose that the matrices $T_k = I_{m_k} + Y_k^T A_{k-1}^{-1} X_k$, for $k = 1, \ldots, p$, are nonsingular. Then*

$$U_k = X_k - \sum_{i=1}^{k-1} U_i T_i^{-1} V_i^T A_0^{-1} X_k, \tag{2.4}$$

ETNA
Kent State University
http://etna.math.kent.edu

$$V_k = Y_k - \sum_{i=1}^{k-1} V_i T_i^{-T} U_i^T A_0^{-T} Y_k \tag{2.5}$$

are well defined for $k = 1, \ldots, p$. *Moreover,*

$$\begin{aligned} A_{k-1}^{-1} X_k &= A_0^{-1} U_k, \\ Y_k^T A_{k-1}^{-1} &= V_k^T A_0^{-1}, \end{aligned}$$

and

$$T_k = I_{m_k} + Y_k^T A_0^{-1} U_k = I_{m_k} + V_k^T A_0^{-1} X_k. \tag{2.6}$$

Proof. Similar to the proof of Theorem 2.1 in [11]. ◻

Denoting by $U = [U_1\ U_2\ \cdots\ U_p]$ and $V = [V_1\ V_2\ \cdots\ V_p]$ the matrices whose block columns are the matrices U_k and V_k, respectively, equation (2.3) can be rewritten as

$$A^{-1} = A_0^{-1} - A_0^{-1} U T^{-1} V^T A_0^{-1}\,, \tag{2.7}$$

which is the block ISM decomposition of the matrix A.

Different choices of the matrices X_k, Y_k and A_0 allow different ways of computing (2.7). Nevertheless, it is convenient that A_0 be a matrix whose inverse is either known or easy to obtain. In the next section we will study two possibilities. The first one is the choice already considered in [11], i.e., $A_0 = sI_n$, where s is a positive scalar and I_n is the identity matrix of size n. The second one is $A_0 = \operatorname{diag}(A_{11}, \ldots, A_{pp})$, where A_{ii} are the main diagonal square blocks of the matrix A partitioned in block form,

$$A = \begin{bmatrix} A_{11} & A_{12} & \ldots & A_{1p} \\ A_{21} & A_{22} & \ldots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \ldots & A_{pp} \end{bmatrix}, \tag{2.8}$$

where $A_{ij} \in \mathbb{R}^{m_i \times m_j}$, $\sum_{k=1}^{p} m_k = n$. In addition we will show the relationship between the two cases when a block Jacobi scaling is applied.

3. Approximate block ISM decompositions. Even if the matrix A is sparse its block ISM decomposition (2.7) is structurally dense. To obtain a sparse block ISM decomposition which can be used as a preconditioner, incomplete factors $\bar{U}_k$ and $\bar{V}_k$ are obtained by dropping off-diagonal block elements during the computation of U_k and V_k. In addition, the inverse of A_0 can be computed approximately and either its exact or its approximate inverse will be denoted by $\bar{A}_0^{-1}$. Once the factors $\bar{U}_k$ and $\bar{V}_k$ have been computed, two different preconditioning strategies can be used,

$$\bar{A}_0^{-1} - \bar{A}_0^{-1} \bar{U} \bar{T}^{-1} \bar{V}^T \bar{A}_0^{-1} \tag{3.1}$$

and

$$\bar{A}_0^{-1} \bar{U} \bar{T}^{-1} \bar{V}^T \bar{A}_0^{-1}. \tag{3.2}$$

In [11] both preconditioners are studied. Although (3.2) requires less computation per iteration than (3.1), the latter tends to converge in fewer iterations, especially for difficult

ETNA
Kent State University
http://etna.math.kent.edu

problems. Therefore, the results presented in the numerical experiments correspond to the block approximate ISM decomposition (3.1). The following algorithm computes the approximate factors.

ALGORITHM 3.1. *Computing the incomplete factors in (3.1) and (3.2).*

(1) ***let*** $U_k = X_k$, $V_k = Y_k$, $(k = 1, \ldots, p)$ ***and*** $\bar{A}_0^{-1} \approx A_0^{-1}$
(2) ***for*** $k = 1, \ldots, p$
(3) ***for*** $i = 1, \ldots, k-1$
$C_i = \bar{U}_i \bar{T}_i^{-1} \bar{V}_i^T$
$U_k = U_k - C_i \bar{A}_0^{-1} X_k$
$V_k = V_k - C_i^T \bar{A}_0^{-T} Y_k$
end for
compute $\bar{V}_k, \bar{U}_k$ ***dropping block elements in*** V_k, U_k
$\bar{T}_k = I_{m_k} + \bar{V}_k^T \bar{A}_0^{-1} X_k$
end for
(4) ***return*** $\bar{U} = [\bar{U}_1\, \bar{U}_2 \cdots \bar{U}_p]$, $\bar{V} = [\bar{V}_1\, \bar{V}_2 \cdots \bar{V}_p]$ *and* $\bar{T} = \operatorname{diag}(\bar{T}_1, \bar{T}_2, \ldots, \bar{T}_p)$

Algorithm 3.1 runs to completion if the pivot matrices $\bar{T}_k$ are nonsingular. It will be shown that this condition holds for H-matrices. To prove these results we first show that the matrices T_k are closely related to the pivots of the block LU factorization applied to A. We will discuss the choice of A_0 separately.

3.1. Case 1: $A_0 = sI_n$**.** Let $X = I_n$ and $Y = (A - sI_n)^T$ be matrices partitioned consistently with (2.8). The kth block column of the matrices Y and X are given explicitly by

$$Y_k = \begin{bmatrix} A_{k1} & \cdots & A_{kk-1} & A_{kk} - sI_{m_k} & \cdots & A_{kp} \end{bmatrix}^T \tag{3.3}$$

and

$$X_k = \begin{bmatrix} 0 & \cdots & I_{m_k} & \cdots & 0 \end{bmatrix}^T. \tag{3.4}$$

With this choice, the expressions (2.4), (2.5), and (2.6) simplify to

$$U_k = X_k - \sum_{i=1}^{k-1} s^{-1} U_i T_i^{-1} V_i^T X_k, \tag{3.5}$$

$$V_k = Y_k - \sum_{i=1}^{k-1} s^{-1} V_i T_i^{-T} U_i^T Y_k, \tag{3.6}$$

and

$$T_k = I_{m_k} + s^{-1} V_k^T X_k = I_{m_k} + s^{-1} V_{kk}^T. \tag{3.7}$$

LEMMA 3.2. *Let A be a matrix partitioned as in (2.8) and let $A_0 = sI$. If the block LU factorization of A can be computed without pivoting, then*

$$T_k = s^{-1} A_{kk}^{(k-1)},$$

where $A_{kk}^{(k-1)}$ is the kth pivot in the block LU factorization of A.

ETNA
Kent State University
http://etna.math.kent.edu

Proof. Observe that the block (i,j) of the matrix $A^{(k)}$ obtained from the kth step of the LU factorization is given by

$$\begin{aligned} A_{ij}^{(k)} &= A_{ij}^{(k-1)} - A_{ik}^{(k-1)}\left(A_{kk}^{(k-1)}\right)^{-1} A_{kj}^{(k-1)} \\ &= A_{ij} - \begin{bmatrix} A_{i1} & A_{i2} & \dots & A_{ik} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1k} \\ A_{21} & A_{22} & \dots & A_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ A_{k1} & A_{k2} & \dots & A_{kk} \end{bmatrix}^{-1} \begin{bmatrix} A_{1j} \\ A_{2j} \\ \vdots \\ A_{kj} \end{bmatrix} \end{aligned}$$

with $i, j > k$. Consider the matrix T_k given by (see section 2):

$$T_k = I_{m_k} + Y_k^T A_{k-1}^{-1} X_k. \tag{3.8}$$

We have

$$\begin{aligned} A_{k-1} &= \left[\begin{array}{ccc|c|c|c} A_{11} & \dots & A_{1,k-1} & A_{1k} & \dots & A_{1p} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} & A_{k-1,k} & \dots & A_{k-1,p} \\ \hline 0 & \dots & 0 & sI_{m_k} & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & sI_{m_p} \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} C_{11} & C_{12} & \dots & C_{1,p-k} \\ \hline 0 & sI_{m_k} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & sI_{m_p} \end{array}\right], \end{aligned}$$

whose inverse is

$$A_{k-1}^{-1} = \begin{bmatrix} C_{11}^{-1} & -s^{-1}C_{11}^{-1}C_{12} & \dots & -s^{-1}C_{11}^{-1}C_{1,p-k} \\ 0 & s^{-1}I_{m_k} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s^{-1}I_{m_p} \end{bmatrix}. \tag{3.9}$$

Then, by substituting (3.9) into (3.8), and bearing in mind (3.3), (3.4), and (2.8), we have

$$\begin{aligned} T_k &= I_{m_k} - s^{-1}\begin{bmatrix} A_{k1} & \dots & A_{k,k-1} \end{bmatrix} \begin{bmatrix} A_{11} & \dots & A_{1,k-1} \\ A_{21} & \dots & A_{2,k-1} \\ \vdots & \ddots & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} \end{bmatrix}^{-1} \begin{bmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{k-1,k} \end{bmatrix} \\ &+ \quad s^{-1}(A_{kk} - sI_{m_k}) \\ &= s^{-1}\left(A_{kk} - \begin{bmatrix} A_{k1} & \dots & A_{k,k-1} \end{bmatrix} \begin{bmatrix} A_{11} & \dots & A_{1,k-1} \\ A_{21} & \dots & A_{2,k-1} \\ \vdots & \ddots & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} \end{bmatrix}^{-1} \begin{bmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{k-1,k} \end{bmatrix} \right) \\ &= s^{-1} A_{kk}^{(k-1)}. \qquad \square \end{aligned}$$

ETNA
Kent State University
http://etna.math.kent.edu

This result generalizes [11, Lemma 3.2] to the case of a block matrix A. If A is either an M-matrix or an H-matrix, the block LU factorization can be done without pivoting [1]. Thus, Algorithm 3.1 runs to completion for these matrices when no dropping strategy is used. The following results show that this situation is also true in the incomplete case.

THEOREM 3.3. *Let A be an M-matrix partitioned as in (2.8). The matrices $\bar{T}_k$ computed by Algorithm 3.1 with $A_0 = sI_n$ are nonsingular M-matrices.*

Proof. The proof proceeds by induction over k, $k = 1, \ldots, p$. We will show that the $\bar{T}_k$ are M-matrices, observing that

$$U_{ik} \geq \bar{U}_{ik} \geq 0, \qquad i \leq k, \tag{3.10}$$

$$V_{ik} \leq \bar{V}_{ik} \leq 0, \qquad i > k, \tag{3.11}$$

$$T_k^{-1} \geq \bar{T}_k^{-1} \geq 0. \tag{3.12}$$

1. For $k = 1$, we have $U_{11} = I_{m_1} \geq 0$. Moreover, $V_{i1} = A_{1i}^T \leq 0$ for $i > 1$ since A is an M-matrix. Therefore, after dropping elements, it follows that $V_{i1} \leq \bar{V}_{i1} \leq 0$. Observe that $\bar{T}_1 = T_1 = s^{-1}A_{11}$ is an M-matrix and equation (3.12) holds.
2. Now, assume that (3.10), (3.11), and (3.12) hold for $k - 1$. For k, we have $U_{kk} = I_{m_k} \geq 0$. For $i < k$, it follows that

$$U_{ik} = -\sum_{j=i}^{k-1} U_{ij} T_j^{-1} V_{kj}^T \geq -\sum_{j=i}^{k-1} \bar{U}_{ij} \bar{T}_j^{-1} \bar{V}_{kj}^T = \bar{U}_{ik} \geq 0.$$

For $i \neq k$,

$$V_{ik} = A_{ki}^T - s^{-1} \sum_{j=1}^{k-1} V_{ij} T_j^{-T} \sum_{l=1}^{j} U_{lj}^T A_{kl}^T = A_{ki}^T - s^{-1} \sum_{j=1}^{k-1} \sum_{l=1}^{j} V_{ij} T_j^{-T} U_{lj}^T A_{kl}^T.$$

Since A is an M-matrix, $A_{kl}^T \leq 0$ for $l \neq k$. Then, for $i > k$,

$$-V_{ij} T_j^{-T} U_{lj}^T A_{kl}^T \leq -\bar{V}_{ij} \bar{T}_j^{-T} \bar{U}_{lj}^T A_{kl}^T \leq 0 \tag{3.13}$$

and

$$V_{ik} \leq A_{ki}^T - s^{-1} \sum_{j=1}^{k-1} \sum_{l=1}^{j} \bar{V}_{ij} \bar{T}_j^{-T} \bar{U}_{lj}^T A_{kl}^T = \bar{V}_{ik} \leq 0.$$

Now dropping elements in $\bar{U}_{ik}$ and $\bar{V}_{ik}$, and mantaining the same notation for the incomplete factors, the inequalities (3.10) and (3.11) hold for k.
Similarly, from (3.7),

$$\begin{aligned} T_k &= s^{-1} \left(A_{kk} - s^{-1} \sum_{j=1}^{k-1} \sum_{l=1}^{j} (V_{kj} T_j^{-T} U_{lj}^T A_{kl}^T)^T \right) \\ &\leq s^{-1} \left(A_{kk} - s^{-1} \sum_{j=1}^{k-1} \sum_{l=1}^{j} (\bar{V}_{kj} \bar{T}_j^{-T} \bar{U}_{lj}^T A_{kl}^T)^T \right) = \bar{T}_k. \end{aligned}$$

By equation (3.13) the matrix $\bar{T}_k$ has non-positive off-diagonal entries. Since T_k is an M-matrix, it follows that $\bar{T}_k$ is also an M-matrix and hence $T_k^{-1} \geq \bar{T}_k^{-1} \geq 0$. ☐

THEOREM 3.4. *Let A be an H-matrix partitioned as in (2.8). The matrices $\bar{T}_k$ computed by Algorithm 3.1 with $A_0 = sI_n$ are nonsingular H-matrices.*

Proof. To simplify the notation, let us denote by B the comparison matrix of A, $\mathcal{M}(A)$, and by (U_j^B, V_j^B, T_j^B) the matrices obtained by applying Algorithm 3.1 to the matrix B.

As before, the proof proceeds by induction over k, $k = 1, \ldots, p$. We will show that $\bar{T}_k$ are H-matrices observing that

$$U_{ik}^B \geq \left|\bar{U}_{ik}\right| \geq 0, \quad i \leq k, \tag{3.14}$$

$$V_{ik}^B \leq -\left|\bar{V}_{ik}\right| \leq 0, \quad i > k, \tag{3.15}$$

$$(T_k^B)^{-1} \geq \left|\bar{T}_k^{-1}\right| \geq 0. \tag{3.16}$$

1. For $k = 1$, we have $\bar{U}_{11} = U_{11} = I_{m_1} \geq 0$. Thus, $\left|\bar{U}_{11}\right| = U_{11}^B$. On the other hand, $V_{i1} = A_{1i}^T$ for $i > 1$ and therefore $-|V_{i1}| = V_{i1}^B$. After dropping elements, it follows that $-\left|\bar{V}_{i1}\right| \geq -|V_{i1}| = V_{i1}^B$. Observe now that $\bar{T}_1 = T_1 = s^{-1}A_{11}$ and $\mathcal{M}(T_1) = T_1^B$ is an M-matrix. Therefore, equation (3.16) holds.
2. Now, assume that (3.14), (3.15), and (3.16) hold until $k-1$. For k, we have $\bar{U}_{kk} = I_{m_k} = U_{kk}^B \geq 0$. For $i < k$, it follows that

$$\begin{aligned}\left|\bar{U}_{ik}\right| &= \left|\frac{1}{s}\sum_{j=1}^{k-1} \bar{U}_{ij}\bar{T}_j^{-1}\bar{V}_{kj}^T\right| \leq \frac{1}{s}\sum_{j=1}^{k-1}\left|\bar{U}_{ij}\right|\left|\bar{T}_j^{-1}\right|\left|\bar{V}_{kj}^T\right| \\ &\leq -\frac{1}{s}\sum_{j=1}^{k-1} U_{ij}^B \left(T_j^B\right)^{-1}\left(V_{kj}^B\right)^T = U_{ik}^B.\end{aligned}$$

In addition, for $i > k$ one has

$$\begin{aligned}-\left|\bar{V}_{ik}\right| &= -\left|A_{ki}^T - \frac{1}{s}\sum_{j=1}^{k-1}\bar{V}_{ij}\bar{T}_j^{-T}\left(\sum_{l=1}^{j}\bar{U}_{lj}^T A_{kl}^T\right)\right| \\ &\geq -\left|A_{ki}^T\right| - \frac{1}{s}\sum_{j=1}^{k-1}\sum_{l=1}^{j}\left|\bar{V}_{ij}\bar{T}_j^{-T}\bar{U}_{lj}^T A_{kl}^T\right| \\ &\geq -\left|A_{ki}^T\right| - \frac{1}{s}\sum_{j=1}^{k-1}\sum_{l=1}^{j}\left|\bar{V}_{ij}\right|\left|\bar{T}_j^{-T}\right|\left|\bar{U}_{lj}^T\right|\left|A_{kl}^T\right|.\end{aligned}$$

Applying (3.14), (3.15), and (3.16) it follows that

$$-\left|\bar{V}_{ik}\right| \geq -\left|A_{ki}^T\right| + \frac{1}{s}\sum_{j=1}^{k-1}\sum_{l=1}^{j} V_{ij}^B\left(T_j^B\right)^{-T}\left(U_{lj}^B\right)^T\left|A_{kl}^T\right| = V_{ij}^B.$$

ETNA
Kent State University
http://etna.math.kent.edu

Now dropping elements in $\bar{U}_{ik}$ and $\bar{V}_{ik}$, and mantaining the same notation for the incomplete factors, the inequalities (3.14) and (3.15) hold for k. In addition,

$$\bar{T}_k = I + \frac{1}{s}\bar{V}_{kk}^T = \frac{1}{s}\left(A_{kk} - \frac{1}{s}\sum_{j=1}^{k-1}\sum_{l=1}^{j}\left(\bar{V}_{kj}\bar{T}_j^{-T}\bar{U}_{lj}^T A_{kl}^T\right)^T\right).$$

We now compare the matrices $\mathcal{M}(\bar{T}_k)$ and T_k^B element by element. We denote by $R_m(\cdot)$ and $C_m(\cdot)$ the mth row and column of a matrix, respectively. Considering the diagonal elements, we have

$$\bar{T}_k(m,m) = \frac{1}{s}A_{kk}(m,m) - \frac{1}{s^2}\sum_{j=1}^{k-1}\sum_{l=1}^{j} R_m\left(\bar{A}_{kl}^T\right)\bar{U}_{lj}\bar{T}_j^{-1}C_m\left(\bar{V}_{kj}^T\right).$$

Then,

$$\begin{aligned}\left|\bar{T}_k(m,m)\right| &\geq \frac{1}{s}\left|A_{kk}(m,m)\right| - \frac{1}{s^2}\sum_{j=1}^{k-1}\sum_{l=1}^{j}\left|R_m\left(A_{kl}^T\right)\bar{U}_{lj}\bar{T}_j^{-1}C_m\left(\bar{V}_{kj}^T\right)\right| \\ &\geq \frac{1}{s}\left|A_{kk}(m,m)\right| - \frac{1}{s^2}\sum_{j=1}^{k-1}\sum_{l=1}^{j} R_m\left(\left|A_{kl}^T\right|\right)\left|\bar{U}_{lj}\right|\left|\bar{T}_j^{-1}\right|C_m\left(\left|\bar{V}_{kj}^T\right|\right).\end{aligned}$$

By applying (3.14), (3.15), and (3.16) it follows that

$$\begin{aligned}\left|\bar{T}_k(m,m)\right| &\geq \frac{1}{s}\left|A_{kk}(m,m)\right| + \frac{1}{s^2}\sum_{j=1}^{k-1}\sum_{l=1}^{j} R_m\left(\left|A_{kl}^T\right|\right)U_{lj}^B\left(T_j^B\right)^{-1}C_m\left(\left(V_{kj}^B\right)^T\right) \\ &= T_k^B(m,m).\end{aligned}$$

Similarly, one has $-\left|\bar{T}_k(m,n)\right| \geq T_k^B(m,n)$ for all $m \neq n$. Then,

$$\mathcal{M}\left(\bar{T}_k\right) \geq T_k^B.$$

By Theorem 3.3 it follows that T_k^B is an M-matrix and hence $\bar{T}_k$ is an H-matrix, which implies that $\left|\bar{T}_k^{-1}\right| \leq \mathcal{M}\left(\bar{T}_k\right)^{-1} \leq \left(T_k^B\right)^{-1}$. $\square$

3.2. Case 2: $A_0 = \mathrm{diag}(A_{11}, \ldots, A_{pp})$**.** Let $X = I_n$ and $Y = (A - A_0)^T$ be matrices partitioned consistently with (2.8). The kth block column of the matrices Y and X are given explicitly by

$$Y_k = \begin{bmatrix} A_{k1} & \cdots & A_{kk-1} & 0_{m_k} & A_{kk+1} & \cdots & A_{kp}\end{bmatrix}^T \tag{3.17}$$

and

$$X_k = \begin{bmatrix} 0 & \cdots & I_{m_k} & \cdots & 0\end{bmatrix}^T.$$

LEMMA 3.5. *Let A be a matrix partitioned as in (2.8). If the block LU factorization of A can be carried out without pivoting, then*

$$T_k = A_{kk}^{(k-1)}A_{kk}^{-1},$$

ETNA
Kent State University
http://etna.math.kent.edu

where $A_{kk}^{(k-1)}$ *is the* k*th pivot in the block LU factorization of* A*.*

Proof. Observe that the block (i,j) of the matrix $A^{(k)}$ obtained from the kth step of LU factorization is given by

$$\begin{aligned} A_{ij}^{(k)} &= A_{ij}^{(k-1)} - A_{ik}^{(k-1)} \left(A_{kk}^{(k-1)}\right)^{-1} A_{kj}^{(k-1)} \\ &= A_{ij} - \begin{bmatrix} A_{i1} & A_{i2} & \dots & A_{ik} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1k} \\ A_{21} & A_{22} & \dots & A_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ A_{k1} & A_{k2} & \dots & A_{kk} \end{bmatrix}^{-1} \begin{bmatrix} A_{1j} \\ A_{2j} \\ \vdots \\ A_{kj} \end{bmatrix} \end{aligned}$$

with $i,j > k$. Consider the matrix T_k given by

$$T_k = I_{m_k} + Y_k^T A_{k-1}^{-1} X_k.$$

With $A_0 = \mathrm{diag}(A_{11}, \dots, A_{pp})$, we have

$$\begin{aligned} A_{k-1} &= \left[\begin{array}{ccc|c|c|c} A_{11} & \dots & A_{1,k-1} & A_{1k} & \dots & A_{1p} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} & A_{k-1,k} & \dots & A_{k-1,p} \\ \hline 0 & \dots & 0 & A_{kk} & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & A_{pp} \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} C_{11} & C_{12} & \dots & C_{1,p-k} \\ \hline 0 & A_{kk} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{pp} \end{array}\right], \end{aligned}$$

whose inverse is

$$A_{k-1}^{-1} = \begin{bmatrix} C_{11}^{-1} & -C_{11}^{-1} C_{12} A_{kk}^{-1} & \dots & -C_{11}^{-1} C_{1,p-k} A_{pp}^{-1} \\ 0 & A_{kk}^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{pp}^{-1} \end{bmatrix}.$$

Then

$$T_k = I_{m_k} - \begin{bmatrix} A_{k1} & \dots & A_{k,k-1} \end{bmatrix} \begin{bmatrix} A_{11} & \dots & A_{1,k-1} \\ A_{21} & \dots & A_{2,k-1} \\ \vdots & \ddots & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} \end{bmatrix}^{-1} \begin{bmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{k-1,k} \end{bmatrix} A_{kk}^{-1}.$$

Post-multiplying by A_{kk}, we have

$$\begin{aligned} T_k A_{kk} &= A_{kk} - \begin{bmatrix} A_{k1} & \dots & A_{k,k-1} \end{bmatrix} \begin{bmatrix} A_{11} & \dots & A_{1,k-1} \\ A_{21} & \dots & A_{2,k-1} \\ \vdots & \ddots & \vdots \\ A_{k-1,1} & \dots & A_{k-1,k-1} \end{bmatrix}^{-1} \begin{bmatrix} A_{1k} \\ A_{2k} \\ \vdots \\ A_{k-1,k} \end{bmatrix} \\ &= A_{kk}^{(k-1)}. \end{aligned}$$

ETNA
Kent State University
http://etna.math.kent.edu

Thus, $T_k = A_{kk}^{(k-1)} A_{kk}^{-1}$. ∎

THEOREM 3.6. *Let A be an M-matrix partitioned as in (2.8). The matrices $\bar{T}_k$ computed by Algorithm 3.1 with $A_0 = \operatorname{diag}(A_{11}, \ldots, A_{pp})$ are nonsingular M-matrices.*

Proof. The proof proceeds by induction over k, $k = 1, \ldots, p$. We will show that $\bar{T}_k$ are M-matrices, observing that

$$U_k \geq \bar{U}_k \geq 0, \tag{3.18}$$

$$V_k \leq \bar{V}_k \leq 0, \tag{3.19}$$

$$T_k^{-1} \geq \bar{T}_k^{-1} \geq 0. \tag{3.20}$$

1. For $k = 1$, we have $U_1 = X_1 \geq 0$. Moreover, $V_1 = Y_1 \leq 0$ since A is an M-matrix. Therefore, after applying a dropping strategy, it follows that $V_1 \leq \bar{V}_1 \leq 0$. Observing that $\bar{T}_1 = T_1 = I_{m_1}$, (3.20) is trivially satisfied.
2. Now, assume that (3.18), (3.19), and (3.20) hold until $k-1$. Then for $i \leq k-1$, we have that
   (a) $-U_i T_i^{-1} V_i^T A_0^{-1} X_k \geq -\bar{U}_i \bar{T}_i^{-1} \bar{V}_i^T A_0^{-1} X_k \geq 0$,
   (b) $-V_i T_i^{-T} U_i^T A_0^{-T} Y_k \leq -\bar{V}_i \bar{T}_i^{-T} \bar{U}_i^T A_0^{-T} Y_k \leq 0$.
   Then, for k, we have
$$U_k = X_k - \sum_{i=1}^{k-1} U_i T_i^{-1} V_i^T A_0^{-1} X_k \geq X_k - \sum_{i=1}^{k-1} \bar{U}_i \bar{T}_i^{-1} \bar{V}_i^T A_0^{-1} X_k = \bar{U}_k \geq 0,$$
$$V_k = Y_k - \sum_{i=1}^{k-1} V_i T_i^{-T} U_i^T A_0^{-T} Y_k \leq Y_k - \sum_{i=1}^{k-1} \bar{V}_i \bar{T}_i^{-T} \bar{U}_i^T A_0^{-T} Y_k = \bar{V}_k \leq 0.$$
   Moreover, $T_k = I_{m_k} + V_{kk}^T A_{kk}^{-1} \leq I_{m_k} + \bar{V}_{kk}^T A_{kk}^{-1} = \bar{T}_k$. Then, $\bar{T}_k$ is an M-matrix and $T_k^{-1} \geq \bar{T}_k^{-1} \geq 0$. ∎

THEOREM 3.7. *Let A be an H-matrix partitioned as in (2.8). The matrices $\bar{T}_k$ computed by Algorithm 3.1 with $A_0 = \operatorname{diag}(A_{11}, \ldots, A_{pp})$ are nonsingular H-matrices.*

Proof. To simplify the notation, let us denote by B the comparison matrix of A, $\mathcal{M}(A)$, and by (U_j^B, V_j^B, T_j^B) the matrices obtained by applying Algorithm 3.1 to the matrix B. Note that $B_0 = \mathcal{M}(A_0)$ and then $|A_0^{-1}| \leq B_0^{-1}$, that is $|A_{kk}^{-1}| \leq B_{kk}^{-1}$ for $k = 1, \ldots, p$. Moreover, $Y_k^B = -|Y_k| \leq 0$.

As before, the proof proceeds by induction over k, $k = 1, \ldots, p$. We will show that the $\bar{T}_k$ are H-matrices by observing that

$$U_k^B \geq \left|\bar{U}_k\right| \geq 0, \tag{3.21}$$

$$V_k^B \leq -\left|\bar{V}_k\right| \leq 0, \tag{3.22}$$

$$(T_k^B)^{-1} \geq \left|\bar{T}_k^{-1}\right| \geq 0. \tag{3.23}$$

1. For $k = 1$, we have $\bar{U}_1 = U_1 = I_{m_1} \geq 0$. Thus, $\left|\bar{U}_1\right| = U_1^B$. On the other hand, $V_1 = Y_1$ and therefore $-\left|V_1\right| = -\left|Y_1\right| = V_1^B$. After applying a dropping strategy, it follows that $-\left|\bar{V}_1\right| \geq -\left|V_1\right| = V_1^B$. Observing that $\bar{T}_1 = T_1 = I_{m_1}$ (3.23) is trivially satisfied.
2. Now, assume that (3.21), (3.22), and (3.23) hold until $k-1$. For k, we have $\bar{U}_k = X_k - \sum_{i=1}^{k-1} \bar{U}_i \bar{T}_i^{-1} \bar{V}_i^T A_0^{-1} X_k$. Then, it follows that

$$\begin{aligned} |\bar{U}_k| &= |X_k - \textstyle\sum_{i=1}^{k-1} \bar{U}_i \bar{T}_i^{-1} \bar{V}_i^T A_0^{-1} X_k| \\ &\leq |X_k| + \textstyle\sum_{i=1}^{k-1} |\bar{U}_i||\bar{T}_i^{-1}||\bar{V}_i^T||A_0^{-1} X_k| \\ &\leq |X_k| - \textstyle\sum_{i=1}^{k-1} U_i^B (T_i^B)^{-1} (V_i^B)^T B_0^{-1} X_k = U_k^B. \end{aligned}$$

In addition, we have $\bar{V}_k = Y_k - \sum_{i=1}^{k-1} \bar{V}_i \bar{T}_i^{-T} \bar{U}_i^T A_0^{-T} Y_k$. Then one has

$$\begin{aligned} -|\bar{V}_k| &= -|Y_k - \textstyle\sum_{i=1}^{k-1} \bar{V}_i \bar{T}_i^{-T} \bar{U}_i^T A_0^{-T} Y_k| \\ &\geq -|Y_k| - \textstyle\sum_{i=1}^{k-1} |\bar{V}_i \bar{T}_i^{-T} \bar{U}_i^T A_0^{-T} Y_k| \\ &\geq -|Y_k| - \textstyle\sum_{i=1}^{k-1} |\bar{V}_i||\bar{T}_i^{-T}||\bar{U}_i^T||A_0^{-T}||Y_k| \\ &= Y_k^B + \textstyle\sum_{i=1}^{k-1} |\bar{V}_i||\bar{T}_i^{-T}||\bar{U}_i^T||A_0^{-T}|Y_k^B \\ &\geq Y_k^B - \textstyle\sum_{i=1}^{k-1} V_i^B (T_i^B)^{-T} (U_i^B)^T B_0^{-T} Y_k^B = V_k^B. \end{aligned}$$

After applying a dropping strategy in the computation of $\bar{U}_k$ and $\bar{V}_k$ and keeping the same notation for the incomplete factors, the inequalities (3.21) and (3.22) hold for k. In addition, $\bar{T}_k = I + \bar{V}_{kk}^T A_{kk}^{-1}$. We now compare the matrices $\mathcal{M}(\bar{T}_k)$ and T_k^B element by element. We denote by $R_m(\cdot)$ and $C_m(\cdot)$ the mth row and column of a matrix, respectively. Considering the diagonal elements, we have

$$\begin{aligned} |T_k(m,m)| &= |1 + R_m(\bar{V}_{kk}^T) C_m(A_{kk}^{-1})| \\ &\geq 1 - |R_m(\bar{V}_{kk}^T) C_m(A_{kk}^{-1})| \\ &\geq 1 + R_m(-|V_{kk}^T|) C_m(B_{kk}^{-1}) \\ &\geq 1 + R_m((V_{kk}^B)^T) C_m(B_{kk}^{-1}) = T_k^B(m,m). \end{aligned}$$

Similarly, one has $-\left|\bar{T}_k(m,n)\right| \geq T_k^B(m,n)$ for all $m \neq n$. Then,

$$\mathcal{M}\left(\bar{T}_k\right) \geq T_k^B.$$

By Theorem 3.6 it follows that T_k^B is an M-matrix and hence $\bar{T}_k$ is an H-matrix, which implies that $\left|\bar{T}_k^{-1}\right| \leq \mathcal{M}\left(\bar{T}_k\right)^{-1} \leq \left(T_k^B\right)^{-1}$. $\square$

3.3. Relation to block Jacobi scaling. In this section we analyze the relation between the two previously studied cases to block Jacobi scaling. In particular, we study the relationship between the factors obtained after applying the exact process to the right block Jacobi scaled matrix AD^{-1}, where $D = \operatorname{diag}(A_{11}, \ldots, A_{pp})$, taking $A_0 = I$ (observe that this is case 1 with $s = 1$), and the factors obtained in the case 2, where $A_0 = D$.

THEOREM 3.8. *Let U, V and T be the factors of the exact block ISM decomposition of the matrix A with $A_0 = \operatorname{diag}(A_{11}, \ldots, A_{pp})$, and let $\hat{U}$, $\hat{V}$ and $\hat{T}$ be the factors corresponding to the block ISM decomposition with $A_0 = I$ of the right scaled matrix AD^{-1}, where*

ETNA
Kent State University
http://etna.math.kent.edu

$D = \mathrm{diag}(A_{11}, \ldots, A_{pp})$. *Then*

$$\hat{U} = U, \qquad \hat{V} = D^{-T} V, \qquad \hat{T} = T.$$

Proof. Observe that for the scaled matrix AD^{-1} equations (3.3) and (3.4) become

$$\hat{Y}_k = \begin{bmatrix} A_{k1}A_{11}^{-1} & \cdots & A_{kk-1}A_{k-1,k-1}^{-1} & 0 & \cdots & A_{kp}A_{pp}^{-1} \end{bmatrix}^T \tag{3.24}$$

and

$$\hat{X}_k = \begin{bmatrix} 0 & \cdots & I_{m_k} & \cdots & 0 \end{bmatrix}^T, \tag{3.25}$$

respectively. Then $\hat{Y}_k = D^{-T} Y_k$ and $\hat{X}_k = X_k$ for all k.

We proceed now by induction. For $k = 1$ it is clear that $\hat{U}_1 = \hat{X}_1 = X_1 = U_1$. On the other hand, $\hat{V}_1 = \hat{Y}_1 = \begin{bmatrix} 0 & \cdots & A_{1t}A_{tt}^{-1} & \cdots & A_{1p}A_{pp}^{-1} \end{bmatrix}^T = D^{-T}Y_1$; see (3.17). Finally, from (3.7), $\hat{T}_1 = I_{m_1} + \hat{V}_{11}^T = I_{m_1} = T_1$.

Assume now that $\hat{U}_i = U_i$, that $\hat{V}_i = D^{-T} V_i$ and $\hat{T}_i = T_i$ for $i = 1, 2, \ldots, k-1$. In this case equation (3.5) becomes

$$\begin{aligned} \hat{U}_k &= \hat{X}_k - \sum_{i=1}^{k-1} \hat{U}_i \hat{T}_i^{-1} \hat{V}_i^T \hat{X}_k \\ &= X_k - \sum_{i=1}^{k-1} U_i T_i^{-1} V_i^T D^{-1} X_k \\ &= U_k, \end{aligned}$$

since the second expression coincides with (2.4) keeping in mind that in this equation A_0 must be replaced by D.

On the other hand equation (3.6) becomes

$$\begin{aligned} \hat{V}_k &= \hat{Y}_k - \sum_{i=1}^{k-1} \hat{V}_i \hat{T}_i^{-T} \hat{U}_i^T \hat{Y}_k \\ &= D^{-T} Y_k - \sum_{i=1}^{k-1} D^{-T} V_i T_i^{-T} U_i^T D^{-T} Y_k \\ &= D^{-T} \big(Y_k - \sum_{i=1}^{k-1} V_i D^{-1} T_i^{-T} U_i^T Y_k \big) \\ &= D^{-T} Y_k, \end{aligned}$$

since the third expression coincides with (2.5) keeping in mind that in this equation A_0 must be replaced by D.

Finally equation (3.7) becomes

$$\hat{T}_k = I_{m_k} + \hat{V}_{kk}^T = I_{m_k} + D^{-T} V_{kk}^T,$$

which is equation (2.6) considering that A_0 must be replaced by D, and the structure of X_k. ∎

Thus, case 2 is equivalent to a left block Jacobi scaling of the block ISM decomposition of the matrix AD^{-1} obtained with $A_0 = I$, i.e., case 1 with $s = 1$.

4. Numerical experiments. In this section the results of numerical experiments performed with the block approximate inverse decomposition (3.1) are presented. We will refer to it as block AISM preconditioner. The test matrices can be downloaded from the University of Florida Sparse Matrix Collection [14]. Table 4.1 provides data on the size, the number of nonzeros and the application area. The block AISM algorithm (Algorithm 3.1) was implemented in Fortran 90 and compiled with the Intel Fortran Compiler 9.1. We compute in each step one block column of the matrices $\bar{U}$ and $\bar{V}$. In order to have a fully sparse implementation, we need to have access to both block columns and rows of these factors. Each matrix is stored in variable block row (VBR) format [21], which is a generalization of the compressed sparse row format. All entries of nonzero blocks are stored by columns, therefore each block can be passed as a small dense matrix to a BLAS routine. In addition, we store at most $lsize$ largest block entries for each row. This additional space is introduced for fast sparse computation of dot products. We note that the row and column structure of $\bar{U}$ and $\bar{V}$ do not necessarily need to correspond each other. In all our experiments we chose $lsize = 5$.

TABLE 4.1
Size (n) and number of nonzero entries (nnz) of tested matrices.

Matrix	n	nnz	Application
SHERMAN2	1,080	23,094	Oil reservoir simulation
UTM1700B	1,700	21,509	Plasma physics
UTM3060	3,060	42,211	Plasma physics
S3RMT3M3	5,357	106,526	Cylindrical shell
S3RMQ4M1	5,489	143,300	Cylindrical shell
UTM5940	5,940	83,842	Plasma physics
CHEM_MASTER1	40,401	201,201	Chemical engineering 2D/3D
XENON1	48,600	1,181,120	Materials
POISSON3DB	85,623	2,374,949	Computational fluid dynamics
S3DKQ4M2	90,499	2,455,670	Cylindrical shell
S3DKT3M2	90,499	1,921,955	Cylindrical shell
XENON2	157,464	3,866,688	Materials
LDOOR	952,203	42,493,817	Large door

Since most of the matrices tested are unstructured, an artificial block partitioning of the matrix is imposed by applying the cosine compressed graph algorithm described in [24]. We will give now a brief overview of this method. The algorithm is based on the computation of the angle between rows of the adjacency matrix C related to A. Rows i and j belong to the same group if their angle is small enough. The cosine of this angle is estimated by computing the matrix CC^T, whose entries (i, j) correspond to the inner product of row i with row j. In order to make the process effective, only the entries (i, j) with $j > i$ are computed, that is, the upper triangular part of CC^T. Moreover, the inner products with the column j that have been already asigned to a group are skipped. Finally, rows i and j are grouped if the cosine of the angle is larger than a parameter τ.

The efficiency of Algorithm 3.1 strongly depends on the method used for finding dense blocks. We found that for unstructured nonsymmetric matrices a value for τ close to 1 leads to very small blocks. In order to evaluate the effect of larger rank updates for some matrices, we choose a small value for this parameter at the price of introducing a large amount of zeros in the nonzero blocks. In the tables the average block size obtained with the cosine algorithm is shown except for the matrix SHERMAN2, for which a natural block size multiple of 6 was the best choice.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 4.2
Effect of the block Jacobi scaling.

Matrix	Avg. block size	Its. (scaled/non-scaled)
S3RMT3M3	11.16	23/76
S3RMQ4M1	15.86	33/56
S3DKQ4M2	17.56	31/60
S3DKT3M2	11.75	23/47
XENON1	2.57	2423/ †
XENON2	2.57	3533/ †

We recall from Theorem 3.8 that applying the block AISM algorithm with $A_0 = I$ to the block Jacobi scaled matrix is almost equivalent to the case 2. In the experiments we did not find significative differences between the two cases. Therefore, we report only results for the block scaled Jacobi matrix. Table 4.2 shows a comparison between scaled and non-scaled block Jacobi for some matrices, that also illustrates the performance differences between case 2 and case 1. We observe that scaling is in general better. Indeed, in some cases (XENON1 and XENON2) it is necessary to scale the matrix to achieve convergence.

Once the block partition has been obtained, the matrix is scaled using block Jacobi and reordered using the minimum degree algorithm applied to the compressed graph of the matrix in order to reduce fill-in.

Exact LU factorization was used to invert the pivot blocks T_k. To preserve sparsity, we follow the strategy recommended in [11], but applied to the matrices partitioned in block form as in (2.8). Fill-in in U is reduced by removing the block entries, whose infinity norm is less than a given drop tolerance Υ. For the factor V, a threshold relative to the infinity norm of the matrix A is used instead, i.e., a block is annihilated if its infinity norm is less than $\Upsilon\|A\|_\infty$. Usually the choice $\Upsilon = 1$ gives good results.

An artificial right-hand side vector was generated such that the solution is the vector of all ones. No significative differences were observed for other choices of the right-hand side vector. The iterative method employed was the BiCGSTAB method [25] with left preconditioning as described in [4]. The initial guess was always $x_0 = 0$, and the iterative solver was stopped when the initial residual was reduced by at least a factor of 10^{-8}, or after 5000 iterations. The tests where performed on a dual Opteron Sun X2200 M2 server.

Table 4.3 shows the results of the block AISM algorithm compared to point AISM for the matrices tested. The block partitioning was obtained with a parameter τ of the cosine algorithm ranging from 0.1 to 0.5 except for the matrix Sherman2. The density of the preconditioner is computed as the ratio between the number of nonzero elements of the preconditioner factors and the number of nonzero elements of the matrix. The number of iterations and the CPU times for computing the preconditioner and to solve the system are also reported.

For the matrices XENON1, XENON2, UTM5940, and SHERMAN2, only the block AISM preconditioner was able to obtain a solution. For the other matrices, we observe that in general block AISM converges in fewer iterations. Concerning the CPU times, both the preconditioner computation and solution times are also smaller. In particular, the preconditioner computation time is reduced significantly.

We show a detailed study for the matrix SHERMAN2 in Figure 4.1. This matrix has been reported as difficult for approximate inverse preconditioning by several authors [2, 7, 15, 16]. In [5] it is solved by applying permutation and scalings in order to place large entries in the diagonal of the matrix. We found that using a block size multiple of 6, the natural structure of the matrix is better exploited and the problem was solved quite easily as can be seen in

TABLE 4.3
Block AISM versus point AISM.

Matrix	Block size	density	Prec. time/Sol. time	Its.
SHERMAN2	1		/	†
	6	0.34	0.0008/0.02	69
S3RMT3M3	1	1.30	0.03/0.055	33
	11.16	0.54	0.01/0.04	23
S3RMQ4M1	1	0.95	0.032/0.046	25
	15.86	0.44	0.010/0.06	33
UTM5940	1	1.45	/	†
	1.45	1.67	0.05/2.40	653
CHEM_MASTER1	1	0.54	0.03/11.8	2523
	3	0.4	0.03/6.63	1138
XENON1	1		/	†
	2.57	0.18	0.03/35.59	2423
POISSON3DB	1	0.19	0.9/18.1	447
	1.17	0.1	0.18/22.16	509
S3DKQ4M2	1	0.91	0.56/1.30	39
	17.56	0.41	0.13/1.05	31
S3DKT3M2	1	1.14	0.51/1.17	37
	11.75	0.51	0.11/0.82	23
XENON2	1		/	†
	2.57	0.18	0.16/154.23	3533
LDOOR	1	0.22	2.10/15.85	63
	10.70	0.40	2.09/10.40	31

Figure 4.1 even without reorderings. We observe that increasing the block size, i.e., the rank of the update, the number of iterations remains more or less constant with some exceptions. Observe that for block size 72, convergence is obtained within very few iterations. However, the best time corresponds to block size 6 and the time tends to increase with the blocksize due to the growth of the density of the preconditioner.

In contrast with the behaviour of the number of iterations exhibited by the SHERMAN2 matrix, usually this number decreases with the rank of the update as we show for the matrices UTM3060, UTM5940, and UTM1700b in Figure 4.2, where a clear decreasing trend can be observed. Since on modern computers the efficient use of the memory cache is fundamental to achieve good performance, one may guess a block size based on cache memory properties of the computer to obtain a good balance between the number of iterations and the performance of the algorithm.

5. Conclusions. In this paper we have presented the block ISM decomposition, which is based on the application of the Sherman-Morrison-Woodbury formula. It is a generalization of the work presented in [11]. Based on the approximate computation of the block ISM decomposition, two different preconditioners have been considered and the relation between them have been established. We have proved the existence of the block ISM decomposition for H-matrices extending results presented in [11].

The performance of the block AISM preconditioners has been studied for a set of matrices arising in different applications. The main conclusion is that block AISM outperforms the point version for the tested problems. In some cases, as for example for the matrices

ETNA
Kent State University
http://etna.math.kent.edu

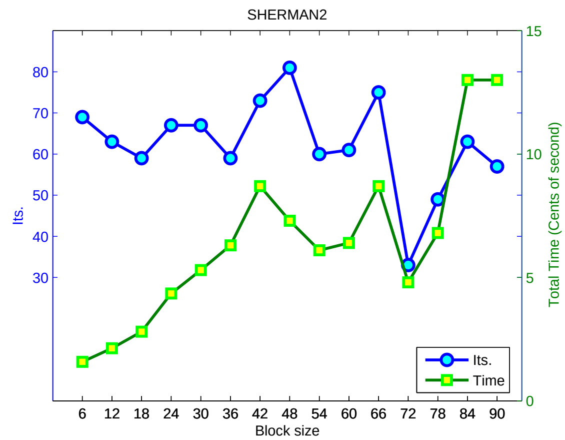

FIG. 4.1. *Number of iterations and total time for the matrix* SHERMAN2*.*

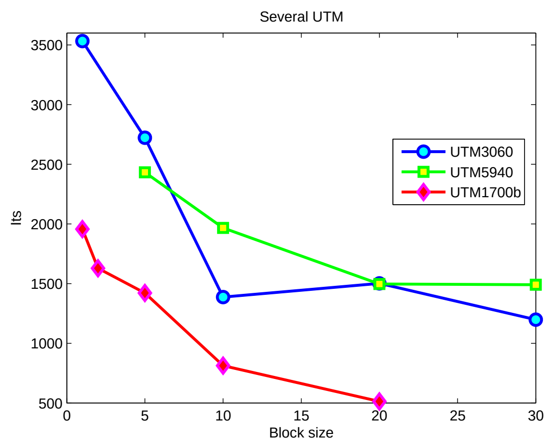

FIG. 4.2. *Number of iterations for the* UTM* *matrices.*

SHERMAN2 and XENON*, convergence was only attained with block AISM.

Except for the matrix SHERMAN2, the size of the block was obtained using a graph compression algorithm. Nevertheless, we found in our experiments that, in general, the larger rank of the update, i.e., the larger the block size, the smaller the number of iterations needed to achieve convergence. However, the overall computational time tends to increase due to the density of the preconditioner. Finally, we note that the block ISM decomposition is the basis for the extension of the preconditioner presented in [10] to block form. This work is currently under study.

REFERENCES

[1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, New York, 1996.
[2] S. T. BARNARD, L. M. BERNARDO, AND H. D. SIMON, *An MPI implementation of the SPAI preconditioner on the T3E*, Internat. J. High Perf. Comput. Applic., 13 (1999), pp. 107–123.
[3] S. T. BARNARD AND M. J. GROTE, *A block version of the SPAI preconditioner*, in Proceedings of the Ninth

SIAM Conference on Parallel Processing for Scientific Computing, B. A. Hendrickson et al., eds., SIAM, Philadelphia, 1999 [CD-ROM].

[4] R. BARRETT, M. BERRY, T. F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE, AND H. VAN DER VORST, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1994.

[5] M. BENZI, J. C. HAWS, AND M. TŮMA, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.

[6] M. BENZI, R. KOUHIA, AND M. TŮMA, *Stabilized and block approximate inverse preconditioners for problems in solid and structural mechanics*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6533–6554.

[7] M. BENZI AND M. TŮMA, *A comparative study of sparse approximate inverse preconditioners*, Appl. Numer. Math., 30 (1999), pp. 305–340.

[8] ———, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 968–994.

[9] R. BRIDSON AND W.-P. TANG, *Refining an approximate inverse*, J. Comput. Appl. Math., 123 (2000), pp. 293–306.

[10] R. BRU, J. MARÍN, J. MAS, AND M. TŮMA, *Balanced incomplete factorization*, SIAM J. Sci. Comput., 30 (2008), pp. 2302–2318.

[11] R. BRU, J. CERDÁN, J. MARÍN, AND J. MAS, *Preconditioning sparse nonsymmetric linear systems with the Sherman-Morrison formula*, SIAM J. Sci. Comput., 25 (2003), pp. 701–715.

[12] E. CHOW AND Y. SAAD, *Approximate inverse techniques for block-partitioned matrices*, SIAM J. Sci. Comput., 18 (1997), pp. 1657–1675.

[13] E. CHOW AND Y. SAAD, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023.

[14] T. A. DAVIS, *University of Florida Sparse Matrix Collection*, available online at http://www.cise.ufl.edu/~davis/sparse/, NA Digest, Vol. 94, issue 42, October 1994.

[15] N. I. M. GOULD AND J. A. SCOTT, *Sparse approximate-inverse preconditioners using norm-minimization techniques*, SIAM J. Sci. Comput., 19 (1998), pp. 605–625.

[16] M. GROTE AND T. HUCKLE, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.

[17] L. Y. KOLOTILINA AND A. Y. YEREMIN, *Factorized sparse approximate inverse preconditionings. I. Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.

[18] T. A. MANTEUFFEL, *An incomplete factorization technique for positive definite linear systems*, Math. Comp., 34 (1980), pp. 473–497.

[19] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[20] A. M. OSTROWSKI, *Über die Determinanten mit Überwiegender Hauptdiagonale*, Commentarii Mathematici Helvetici, 10 (1937), pp. 69–96.

[21] Y. SAAD, *SPARSKIT: A basic tool kit for sparse matrix computations*, Tech. Report 90–20, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffet Field, CA, 1990.

[22] ———, *ILUT: a dual threshold incomplete LU factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.

[23] ———, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

[24] ———, *Finding exact and approximate block structures for ilu preconditioning*, SIAM J. Sci. Comput., 24 (2003), pp. 1107–1123.

[25] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems*, SIAM J. Sci. Stat. Comput., 12 (1992), pp. 631–644.

[26] R. S. VARGA, *Matrix Iterative Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1962.

[27] M. WOODBURY, *Inverting modified matrices*, Tech. Report Memorandum report 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.