CONDITIONING ANALYSIS OF INCOMPLETE CHOLESKY FACTORIZATION WITH ORTHOGONAL DROPPING*

ARTEM NAPOV†

Abstract. We analyze preconditioners based on incomplete Cholesky factorization in which each neglected (dropped) component is orthogonal to the approximation being kept. We present a general estimate for the condition number of the preconditioned system which only depends on the accuracy of individual approximations. The estimate is further improved if given rows of the factor are not modified by consecutive approximations. In particular, if only the newly computed rows are modified by each approximation, the estimate is further shown to be sharp. The analysis is illustrated with some existing factorizations in the context of discretized elliptic partial differential equations.

Key words. incomplete Cholesky, conditioning analysis, convergence analysis, iterative methods, preconditioner

AMS subject classifications. 65F08, 65F35

DOI. 10.1137/120870414

1. Introduction. We consider incomplete Cholesky factorizations for the iterative solution of symmetric positive definite (SPD) $N \times N$ linear systems

\begin{equation}
Au = b.
\end{equation}

Incomplete Cholesky factorization is commonly described with the help of Cholesky version of Gaussian elimination, which amounts to computing an upper triangular matrix $R$ such that $A = R^T R$. Incomplete factorization is then obtained by introducing approximation into the elimination process (see, e.g., [1, 12, 14, 23, 25]).

Here we are concerned with incomplete factorizations based on low-rank approximation. These techniques have been successfully applied to linear systems arising from discretized partial differential equations (PDEs) [3, 4, 13, 27, 26, 28, 16]. Unlike classical incomplete factorization methods [20, 17, 22], which rely on the dropping of individual entries, the new methods approximate some dense blocks in the factors with low-rank matrices. The resulting approximate factors may then remain dense but acquire some structure, and the term data-sparse is often used to describe them. It is now known that for systems arising from discretization of PDEs, individual low-rank approximations are often possible with almost arbitrary accuracy for a rank which is independent of, or slowly varying with, the block size [4, 13, 9]. (See also [5, 6] for the related results.) Whereas substantial effort has been invested in finding the applications where the low-rank property is present, the impact of the individual approximations on the quality of the preconditioner is less well understood.

In this paper, we present the analysis which covers two such preconditioners: [16] and, under a slightly modified form, [28]. Both methods are variants of incomplete

*Received by the editors March 19, 2012; accepted for publication (in revised form) by S. C. Eisenstat June 4, 2013; published electronically August 1, 2013. This work was supported by Director, Office of Science, Office of Advanced Scientific Computing Research of the U.S. Department of Energy under contract DE-AC02-05CH11231. The work on the revised version was supported by Université Libre de Bruxelles (“Premier assistant”).

†Service de Métrologie Nucléaire, Université Libre de Bruxelles (C.P. 165/84), B-1050 Brussels, Belgium (anapov@ulb.ac.be, ulb.ac.be/~anapov).
Cholesky factorization. Whereas they are conceptually different and produce factors in different data-sparse formats, they both exploit the low-rank property by “dropping” components which are orthogonal to the low-rank approximations being kept. This is motivated by the observation that the successively formed Schur complements then do not decrease (in the SPD sense) when approximation is performed, which in turn guarantees that the preconditioner can be constructed for any SPD matrix $A$.

The analysis in the present paper also applies to any SPD matrix $A$. The starting point is a generic incomplete Cholesky factorization algorithm which forms a common framework for the aforementioned factorizations. It allows approximation within an off-diagonal block of the factor and also requires the orthogonality between the component rejected during each approximation step and the approximation that is kept.

The resulting factorization is then considered as a preconditioner for the original system. We present a general upper bound on the corresponding spectral condition number (i.e., the quotient of the largest and smallest eigenvalues of the preconditioned system matrix), as required to estimate the convergence rate of iterative methods such as conjugate gradient [1, 14, 23]. The bound involves quantities which only depend on the individual orthogonal approximations and in a way measure their accuracy. A related accuracy measure is introduced in [28]; the analysis there allows, however, for only one approximation step.

Although both incomplete factorizations [16, 28] proceed by progressively computing the rows of the factor, they modify different groups of computed rows during the approximation steps. This may further lead to an improved condition number estimate involving the same accuracy measures if the modified rows (and the rows with lower indices) are no longer modified during the consecutive steps. The ideal case where only the newly computed rows are modified at each step (and, hence, each row is modified at most once) is of particular interest. This case is referred to as one-level and the related estimate is shown to be sharp in that for every set of accuracy values there exist a matrix and a corresponding incomplete factorization preconditioner, obtained using orthogonal dropping, that allow to reach the bound.

A particular one-level algorithm arises when the off-diagonal blocks of the factor are set to zero, the resulting preconditioner being then the block-diagonal part of $A$. The accuracy measures then coincide with so-called Cauchy–Bunyakovsky–Schwarz (CBS) constants, which are commonly used in the eigenvalue estimates of a block-diagonally preconditioned system. The case when the preconditioner has $2 \times 2$ block diagonal form is well understood [2, 1]; however, the extension to block-diagonal preconditioners with multiple blocks stemming form our analysis leads to a better bound than one could obtain by recursively applying the $2 \times 2$ estimate. In addition, the abovementioned sharpness property carries over to the block-diagonal case.

Eventually, our analysis is illustrated with the factorization algorithms based on [28], [16] and the one-level variant in the context of model problem arising from a low-order discretization of a second-order PDE. For this example all the considered preconditioners have similar conditioning properties, and further, the bound for the one-level variant allows an accurate prediction of their condition number. We further show how the approximation threshold can be modified to keep the condition number bounded independently of the problem size.

The reminder of the paper is organized as follows. In section 2 we introduce our generic incomplete Cholesky factorization with orthogonal dropping and relate it to the existing methods. The bounds on the condition number are presented in section 3 and illustrated on a model problem in section 4. Concluding remarks are stated in section 5.
Notation. \([i,j] = \{i, i+1, \ldots, j\}\) represents the ordered set of integers ranging from \(i\) to \(j\). \(I\) stands for identity matrix and \(O\) for zero matrix (matrix with all entries being zero). For any vector \(v\), \(\|v\|\) is its Euclidian norm. For any matrix \(C\), the induced matrix norm is
\[
\|C\| = \max_{v \neq 0} \frac{\|Cv\|}{\|v\|}.
\]
For any SPD matrix \(D\), \(\lambda_{\text{max}}(D)\) and \(\lambda_{\text{min}}(D)\) are, respectively, its largest and its smallest eigenvalue, both eigenvalues being real and positive; the spectral condition number \(\kappa(D) = \lambda_{\text{max}}(D)/\lambda_{\text{min}}(D)\) is then well defined. For any \(n \times n\) block matrix \(E = (E_{i,j})\) and any \(i \leq k\),
\[
E_{i:k,j} = \begin{pmatrix} E_{i,j}^T & \cdots & E_{k,j}^T \end{pmatrix}^T,
\]
and for any \(j \leq m\),
\[
E_{i:k,j:m} = \begin{pmatrix} E_{i:k,j} \cdots E_{i:k,m} \end{pmatrix}.
\]

2. Factorization algorithm.

2.1. General setting. We consider that an SPD system matrix \(A\) is partitioned into a \(n \times n\) block form
\[
A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1}^T & \cdots & A_{n,n} \end{pmatrix}
\]
with diagonal blocks \(A_{i,i}, i = 1, \ldots, n\), being square.

Let us first briefly recall with Algorithm 2.1 the exact (as opposed to incomplete) version of Cholesky factorization; it is stated below under the “block dot product” form [12]. The algorithm returns an upper triangular factor
\[
R = \begin{pmatrix} R_{1,1} & \cdots & R_{1,n} \\ \vdots & \ddots & \vdots \\ R_{n,1} & \cdots & R_{n,n} \end{pmatrix}
\]
with the same block partitioning as \(A\) and such that
\[
A = R^T R.
\]

Note that the diagonal blocks \(R_{i,i}\) in (2.2) are upper triangular and for \(i > j\) there holds \(R_{i,j} = O\), \(i, j = 1, \ldots, n\).

The main stages of the algorithm, repeated for each block \(R_{i,j}\) with \(i \leq j\), are the update operation (line 1a) followed by either factorization (line 1b) or solution (line 1c). Each execution of the inner loop (lines 1–1c) yields a new block row of the factor.

Now, the incomplete Cholesky factorization as given by Algorithm 2.2 differs from the exact version mainly in two respects: it performs approximation operations (line 2), and its outer loop runs through a given sequence of block indices \(i_k, k = 1, \ldots, \ell\), which may differ from \(i = 1, \ldots, n-1\) (block row \(n\) is computed separately, at line 3). The second difference has no effect in a particular situation when \(i_k = k\) \(k = 1, \ldots, n-1\): step \(k\) of the outer loop of Algorithm 2.2 then amounts to computing
one block row of the factor (lines 1-1c) followed by one approximation operation (line 2), also known as dropping; the factorization is completed by computing the remaining block \( R_{n,n} \) (line 3). This is further illustrated with Figures 2.1 and 2.2. In a general situation, we allow for multiple approximations in between two consecutive block row computations. The index value \( i_k \) then represents the number of block rows of the factor available at the end of step \( k \), and this value may therefore remain unchanged for multiple steps; see Figure 2.3 for an example. We therefore set \( i_1 = 1 \) and allow either \( i_k = i_{k-1} \) or \( i_k = i_{k-1} + 1 \), \( k = 2, \ldots, \ell \), with \( i_\ell = n - 1 \). Since one approximation operation is performed at every step, the number of steps \( \ell \) also corresponds to the overall number of approximations.

Regarding the approximation operation, it only applies to the “off-diagonal” part of the computed block rows, that is, at step \( k \), \( k = 1, \ldots, \ell \), it only modifies the submatrix \( R_{1:i_k, i_k+1:n} \) corresponding to the block columns \( i_k + 1 \) through \( n \), and leaves the “diagonal” part \( R_{1:i_k, 1:i_k} \) unchanged. Further, we only consider dropping schemes that yield an approximation orthogonal to the approximation error, that is, denoting by \( \tilde{R}_{12}^{(k)} \) the off-diagonal submatrix \( R_{1:i_k, i_k+1:n} \) prior to the approximation operation at step \( k \) and by

\[
(2.3) \quad \tilde{R}_{12}^{(k)} = \text{APPROX}_k \left( R_{12}^{(k)} \right)
\]

the same submatrix after this operation, we require the equality

\[
(2.4) \quad \tilde{R}_{12}^{(k)T} \left( R_{12}^{(k)} - \tilde{R}_{12}^{(k)} \right) = O
\]
\[ k = 1 \ (i_1 = 1) \quad k = 2 \ (i_2 = 2) \quad k = 3 \ (i_3 = 3) \quad k = 4 \ (i_4 = 4) \]

Fig. 2.1. Block rows of the factor available at the end of step \( k \) of the incomplete Cholesky algorithm with \( n = 5 \), \( \ell = 4 \), and \( i_k = k \), \( k = 1, \ldots, 4 \); dark blue (dark gray) blocks correspond to a particular set of block rows modified by approximation operation at step \( k \), light blue (light gray) blocks remain unmodified, and thick contour delimits the block row added during the step \( k \). The considered set of modified block rows corresponds to the SSS pattern, see section 2.2.

\[ k = 1 \ (i_1 = 1) \quad k = 2 \ (i_2 = 2) \quad k = 3 \ (i_3 = 3) \quad k = 4 \ (i_4 = 4) \]

Fig. 2.2. Block rows of the factor for \( n = 5 \), \( \ell = 4 \), and \( i_k = k \), \( k = 1, \ldots, 4 \). Modified rows correspond to the one-level pattern from section 2.2.

\[ k = 1 \ (i_1 = 1) \quad k = 2 \ (i_2 = 2) \quad k = 3 \ (i_3 = 2) \quad k = 4 \ (i_4 = 3) \]

\[ k = 5 \ (i_5 = 4) \quad k = 6 \ (i_6 = 4) \quad k = 7 \ (i_7 = 4) \]

\[ \{P_1, P_2, P_3, P_4, P_5, P_6, P_7\} \]

Fig. 2.3. Block rows of the factor for \( n = 5 \) and \( \ell = 7 \) with indices of the modified rows being \( P_1 = \{1\}, P_2 = \{2\}, P_3 = \{1, 2\}, P_4 = \{3\}, P_5 = \{4\}, P_6 = \{3, 4\}, \) and \( P_7 = \{1, 2, 3, 4\} \). Modified rows correspond to the HSS pattern from section 2.2 based on the tree in the right bottom corner.

to hold. Some examples of approximation operations that fulfill the above orthogonality condition are discussed in section 2.3. Of course, not all the block rows in \([1, i_k]\) have to be (and are in practice) modified; some of them may remain unchanged, and, as shown latter, this may further improve the analysis. Some known patterns of modified block rows are illustrated with Figures 2.1–2.3 and further described in section 2.2.
Here we consider incomplete factorization $R^T R$ produced by the above algorithm as a preconditioner for the original system (1.1). Our primary concern is therefore the analysis of the corresponding spectral condition number

$$\kappa(R^{-T} AR^{-1}) = \frac{\lambda_{\max}(R^{-T} AR^{-1})}{\lambda_{\min}(R^{-T} AR^{-1})}$$

that governs the convergence of iterative methods, such as conjugate gradient. Note that the above quantity is well defined; indeed, as shown in section 3, if the approximation operation in Algorithm 2.2 satisfies the orthogonality condition (2.3), (2.4), then this algorithm always terminates and the resulting preconditioner $R^T R$ is SPD.

2.2. Relation to existing methods. The incomplete Cholesky factorization described in Algorithm 2.2 provides a generic framework that covers several existing preconditioners. This is, for instance, the case of incomplete Cholesky factorizations in [16, 28] (the latter being considered here under a slightly different form; see below). Both methods are similar in that they exploit individual low-rank approximations to reduce both the storage requirement and the operation complexity of the factorization. More precisely, they require with a proper block partitioning at most $O(r_{\max} N^2)$ operations to factorize a $N \times N$ matrix, where $r_{\max} = \max_{1 \leq k \leq \ell} r_k$ is the maximal rank from the approximation step, and need at most $O(r_{\max} N)$ storage for the factor; these estimates may further be improved if the matrix is sparse. Hence, if $r_{\max} \ll N$, they compare favorably to the (exact) Cholesky factorization, which requires $O(N^3)$ operations and $O(N^2)$ storage.

Now, assuming the same block partitioning (2.1) of $A$, these algorithms mainly differ by the set of block rows that are effectively modified during each step of the incomplete Cholesky algorithm. This difference is motivated by the data-sparse structure of the resulting factors: sequentially semiseparable (SSS) in [16] versus hierarchically semiseparable (HSS) in [28]. Below we present the block row modification patterns that are inspired by these two factorizations.

Considering first the SSS pattern, it amounts to performing the approximation on the whole off-diagonal part of the factor, that is, setting $\ell = n - 1$ and $i_k = k, k = 1, \ldots, \ell$, the submatrix $R_{\cdot : k, k+1: n}$ is modified at every step. This situation is illustrated in Figure 2.1. Note that the factorization in [16] corresponds to the SSS pattern and, moreover, uses an orthogonal dropping scheme which preserves a given set of vectors.

Regarding the HSS pattern, it requires an auxiliary tree with $n - 1$ leaves and with nodes ordered in a postorder, i.e., children are numbered before their parent; see the bottom right corner of Figure 2.3 for an example. Denoting with $P_k \in [1, i_k]$ the set of block rows modified during the step $k$, one obtains the HSS pattern by setting $P_k = \{i\}$ if $k$ is the tree index of the $i$th leaf, and $P_k = \bigcup_{i \in \text{children}(k)} P_i$ if the node $k$ is not a leaf. For a perfect binary tree of height $t$ (the only considered here) this further entails $n = 2^t + 1$ and $\ell = 2^{t+1} - 1$. This pattern is illustrated in Figure 2.3 for $t = 2$; note in particular that the rows modified during step 1, 3, or 4 (and those having lower indices) are not modified during the following step.

Note that the preconditioner in [28] differs from the Algorithm 2.2 with HSS pattern in that some dropping may also occur in the diagonal part $R_{\cdot i_k; 1: i_k}$ of the factor. All in all, this additional dropping slightly increases the cost of the factorization (in fact, dropping applies to larger matrices) but is likely to improve the storage requirements. Last but not least, the numerical experiments below indicate that the performance of both methods is similar. We therefore recommend using the method...
in [28], while stating that the analysis of HSS pattern provides some insight for this method as well.

It should also be noted that the rank values \( r_k \), \( k = 1, \ldots, \ell \), may sometimes differ substantially between the SSS and HSS data structures (usually favoring the HSS one). This difference may often be determinant for the cost of the preconditioner; it is, however, less important for the condition number analysis undertaken in this work.

Now, we also consider the one-level pattern where only the newly computed block row is approximated, that is, setting \( \ell = n - 1 \) and \( i_k = k \), \( k = 1, \ldots, \ell \), only the block row \( R_{k,k+1:n} \) is modified at step \( k \). (See Figure 2.2 for an example.)

Note that a situation when all the entries in the off-diagonal part are set to zero, that is, when \( \text{APPROX}_k(\cdot) = O \) for all \( k \), is also compatible with our orthogonality condition (2.3), (2.4). The approximate factor then reduces to a block-diagonal form \( R = \text{blockdiag}(R_{k,k}) \). Moreover, line 1a of Algorithm 2.2 then amounts to \( T = A_{i_k,j} \), whereas for \( j = i_k \), line 1b further yields \( R_{i_k,i_k}^T R_{i_k,i_k} = T = A_{i_k,i_k} \); hence \( R^T R = \text{blockdiag}(A_{k,k}) \) and the preconditioner corresponds to a block-diagonal part of \( A \). On the other hand, one may equivalently zero out only the newly computed block row, and hence this block-diagonal preconditioner may also be obtained with one-level modification pattern.

2.3. Orthogonal dropping. In the context of data-sparse incomplete factorizations as described above, dropping occurs when a part of the factor is approximated by a rank-deficient (also called low-rank) matrix. This operation is commonly implemented using a truncated version of an orthogonal decomposition, such as truncated SVD or rank-revealing QR [7, 8, 10, 15]. More precisely, for a given submatrix \( \mathcal{R}_{12}^{(k)} \) and a threshold \( \text{tol}_a \), this amounts to first obtaining a factorization

\[
(2.6) \quad \mathcal{R}_{12}^{(k)} = (Q_1 \quad Q_2) (U_1 \quad U_2)^T,
\]

such that \( Q = (Q_1 \quad Q_2) \) is orthogonal and

\[
\|U_2\| \leq \text{tol}_a.
\]

The approximation then corresponds to a low-rank term \( \tilde{\mathcal{R}}_{12}^{(k)} = Q_1 U_1^T \) whose rank \( r_k \) is given by the number of columns in \( Q_1 \). On the other hand, the approximation error is given by \( \mathcal{R}_{12}^{(k)} - \tilde{\mathcal{R}}_{12}^{(k)} = Q_2 U_2^T \). Hence, the condition (2.3), (2.4) follows directly from \( Q_1^T Q_2 = O \), whereas the orthogonality of columns in \( Q_2 \) further implies

\[
(2.7) \quad \| \mathcal{R}_{12}^{(k)} - \tilde{\mathcal{R}}_{12}^{(k)} \| \leq \text{tol}_a.
\]

In the case of truncated SVD the threshold may be chosen explicitly, by discarding the singular values of \( \mathcal{R}_{12}^{(k)} \) that are lower than the threshold value. This may be done for an absolute truncation threshold \( \text{tol}_a \), but also for a relative one, which amounts to \( \text{tol}_a = \text{tol}_r \| \mathcal{R}_{12}^{(k)} \| \), since \( \| \mathcal{R}_{12}^{(k)} \| \) then corresponds to the largest singular value. Regarding the rank revealing factorizations, the threshold is only available indirectly, usually through an inequality like \( \text{tol}_r < p \cdot \text{tol}_{\text{RRQR}} \), where \( \text{tol}_{\text{RRQR}} \) is the truncation threshold of the rank-revealing algorithm and \( p \) is a low-order polynomial depending on the dimensions of \( \mathcal{R}_{12}^{(k)} \). [10, 15].

Now, if only the block rows with given indices are to be modified, the decomposition (2.6) may be applied directly to those rows. Indeed, denoting with \( C \) the rows of \( \mathcal{R}_{12}^{(k)} \) to be approximated and with \( \tilde{C} \) their approximation, we note that

\[
\mathcal{R}_{12}^{(k)} - \tilde{\mathcal{R}}_{12}^{(k)} = \Pi \left( \begin{array}{c} O \\ C - \tilde{C} \end{array} \right),
\]
where $\Pi$ is a permutation which enumerates those rows last. The “local” orthogonality condition
\[
\tilde{C}^T (C - \tilde{C}) = 0
\]
is then easily shown to imply its “global” version (2.4), whereas, since $\Pi$ is orthogonal, there further holds
\[
\| R_{12}^{(k)} - \tilde{R}_{12}^{(k)} \| = \| C - \tilde{C} \|.
\]

3. Analysis.

3.1. Preliminaries. The main tool of our analysis is the sequence of auxiliary matrices $B_k$, $k = 0, \ldots, \ell$. $B_0$ is formally defined as an incomplete Cholesky preconditioner $R^T R$ produced by Algorithm 2.2 in which only the first $k$ approximations are taken into account, whereas the remaining $\ell - k$ are skipped. In particular, $B_0$ then corresponds to the original system matrix, whereas $B_{\ell}$ is the resulting preconditioner.

To make the above definition more useful we exploit the following well-known property (see, e.g., [1, 12]): if $R = (R_{ij})$ is the upper triangular (exact) Cholesky factor such that $R^T R = A$, its first $i$ block rows, corresponding to $R_{1:i,1:i}$ and $R_{1:i,i+1:n}$, are known to satisfy
\[
(A) = \begin{pmatrix} R^T_{1:i,1:i} & S_A^{(i)} \\ S_A^{(i)} & I \end{pmatrix},
\]
where
\[
S_A^{(i)} = A_{i+1:n,i+1:n} - R^T_{1:i,i+1:n} R_{1:i,i+1:n}
\]
is the Schur complement of $A$ corresponding to the bottom right $(n - i) \times (n - i)$ submatrix. In other words, an SPD matrix $A$ is uniquely determined by the first $i$ block rows of its factor and its bottom right $(n - i) \times (n - i)$ submatrix.

Consider now the incomplete Cholesky factorization implemented in Algorithm 2.2. At the end of step $k$, $k = 1, \ldots, \ell$, the first $i_k$ block rows of the factor, corresponding to $R_{11}^{(k)} = R_{1:i_k,i_k}$ and $R_{12}^{(k)} = R_{1:i_k,i+1:n}$, incorporate the first $k$ approximations. If the remaining $\ell - k$ approximations are ignored, these rows are also the final $i_k$ rows. On the other hand, the bottom right $(n - i_k) \times (n - i_k)$ submatrix has not been modified yet and equals $A_{i_k+1:n,i_k+1:n}$. The above property therefore yields
\[
B_k = \begin{pmatrix} R_{11}^{(k)} & S_B^{(k)} \\ S_B^{(k)} & I \end{pmatrix}
\]
with
\[
S_B^{(k)} = A_{i_k+1:n,i_k+1:n} - R_{12}^{(k)} T R_{12}^{(k)}
\]
denoting the corresponding Schur complement of $B_k$.

Now, the submatrix $R_{1:i_k,i+1:n}$ is precisely the one approximated during the step $k$. Hence, denoting this submatrix prior to the $k$th approximation operation by $R_{12}^{(k)}$, we have
\[
\tilde{R}_{12}^{(k)} = \text{APPROX}_k \left( R_{12}^{(k)} \right),
\]
and the relation (2.4) holds. \( R_{11}^{(k)} \) remains the same and, together with \( R_{12}^{(k)} \), it incorporates only the first \( k - 1 \) approximations; using again the above property gives therefore

\[
B_{k-1} = \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ R_{12}^{(k)} & S_B^{(k)} \end{pmatrix} \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ S_B^{(k)} & I \end{pmatrix}, \quad k = 1, \ldots, \ell,
\]

where

\[
S_B^{(k)} = A_{i_k+1:n, i_k+1:n} - R_{12}^{(k)} R_{12}^{(k)^T}.
\]

Relations (3.2), (3.3) together with (3.4), (3.5) and the orthogonality condition (2.4) form the basis of our analysis.

Before going further, we recall with the following lemma some known facts about the Schur complement. In particular, the first statement relates the Schur complement as appearing, for instance, in (3.1) to its more common form.

**Lemma 3.1.** Let \( A \) be \( N \times N \) and satisfy

\[
A = \begin{pmatrix} R_{11}^{T} & S_A \\ R_{12}^{T} & I \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ S_A & I \end{pmatrix}
\]

for some \( R_{11}, R_{12}, \) and \( S_A \) of order \( M \times M, M \times (N-M), \) and \( (N-M) \times (N-M), \) respectively.

(a) If

\[
A = \begin{pmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{pmatrix}
\]

is the \( 2 \times 2 \) block partitioning induced by the partitioning in (3.6) and if \( R_{11} \) is invertible, then

\[
S_A = A_{22} - A_{12}^T A_{11}^{-1} A_{12}.
\]

(b) \( A \) is SPD if and only if both \( R_{11} R_{11}^T \) and \( S_A \) are SPD.

(c) If \( R_{11} \) and \( S_A \) are invertible, then

\[
A^{-1} = \begin{pmatrix} * & * \\ * & S_A^{-1} \end{pmatrix}.
\]

Another important component of our analysis is the relation

\[
\widetilde{S}_B^{(k)} = S_B^{(k)} + \left( R_{12}^{(k)} - \widetilde{R}_{12}^{(k)} \right)^T \left( R_{12}^{(k)} - \widetilde{R}_{12}^{(k)} \right)
\]

between the Schur complements \( S_B^{(k)}, \widetilde{S}_B^{(k)} \) before and after the approximation operation at step \( k \), and the approximation error \( R_{12}^{(k)} - \widetilde{R}_{12}^{(k)} \). It follows directly from the definitions (3.5), (3.3) of \( S_B^{(k)} \) and \( \widetilde{S}_B^{(k)} \) together with the orthogonality condition (2.4). Its main consequence is that the orthogonality condition (2.4) ensures that all \( B_k, k = 1, \ldots, \ell \), are SPD and the Algorithm 2.2 is breakdown free, that is, it always produces a preconditioner in the factored form. This property is already observed in [16, 28] for the related variants, and we restate it in Lemma 3.2 for a general case, together with some auxiliary results.

**Lemma 3.2.** Let \( A = (A_{i,j}) \) be SPD and partitioned as in (2.1), and let \( B_k, k = 1, \ldots, \ell \), be defined as in the beginning of section 3.
Then
(a) $B_k, k = 1, \ldots, \ell$, is SPD.
(b) Algorithm 2.2 does not breakdown.
(c) There holds

(3.9)

$$\text{blockdiag}(B_k) = (A_{1,1}, \cdots, A_{i_k,i_k}, A_{i_k+1:n,i_k+1:n}), \quad k = 1, \ldots, \ell.$$  

Proof. In what follows $R_{12}^{(k)}$ and $\tilde{R}_{12}^{(k)}$, $k = 1, \ldots, \ell$, are defined as in (3.4) and (3.2); $S_B^{(k)}$ and $\tilde{S}_B^{(k)}$, $k = 1, \ldots, \ell$, are given by (3.3) and (3.5).

To prove the first statement, we note that if $B_{k-1}$ is SPD, then so are $R_{11}^{(k)} R_{11}^{(k)}$, $B$, and the Schur complement $S_B^{(k)}$, as follows from Lemma 3.1(b) applied to (3.4). It further follows from (3.8) that $\tilde{S}_B^{(k)}$ is SPD, and applying again Lemma 3.1(b) to (3.2) shows that $B_k$ is also SPD. Now, statement (a) follows from this recursive argument and the fact that $B_0 = A$.

Regarding the second statement, we note that Algorithm 2.2 may only break down at line 1b, failing to find $R_{i_k,i_k}$ such that $R_{i_k,i_k}^T R_{i_k,i_k} = T$. However, one may check that $T$ is then the leading block of the Schur complement $\tilde{S}_B^{(k-1)}$, and, since the latter is SPD, its leading block may always be factorized.

Eventually, we prove (3.9). Let $B_k = ((B_k)_{i,j})$ be the partitioning into $n \times n$ blocks induced by (2.1). Then, there holds

$$B_k = \begin{pmatrix} (B_{k-1})_{1:i_k,1:i_k} & * \\ * & A_{i_k+1:n,i_k+1:n} \end{pmatrix},$$

where the top left block follows from the comparison of (3.2) and (3.4), whereas the bottom right stems from (3.2) together with (3.3). Applying the above result for $k = 1, \ldots, \ell$ and using $B_0 = A$ finishes the proof of (3.9).

\section{3.2. Main theorem.} We are now ready to prove our main theorem. It assumes that the first $i_k$ block rows of the factor, computed during the first $k$ steps, are not modified over the following $p$ steps; this is in particular always true if $p = 0$. The extreme eigenvalues of $B_{k+p}^{-1} B_{k-1}$ are then related to those of $B_{k+p}^{-1} B_k$ with the help of an additional parameter $\gamma_k$ defined with (3.12). Note that this latter involves the Schur complement of $B_k$ and the approximation error at step $k$, and in some sense measures the approximation accuracy.

\textbf{Theorem 3.3 (main theorem).} Let $A$ be SPD and $B_k$, $k = 0, \ldots, \ell$, be defined as in section 3.1. Let $\tilde{S}_B^{(k)}$, $k = 1, \ldots, \ell$, be given by (3.3), where $R_{11}^{(k)}$, $\tilde{R}_{12}^{(k)}$ stand for, respectively, $R_{1:1:i_k-1,i_k}$ and $R_{1:i_k-1,i_k+1:n}$, as obtained at the end of step $k$ of Algorithm 2.2; let $R_{12}^{(k)}$ be given by $R_{1:i_k-1,i_k+1:n}$ as defined prior to the line 2 of the same step. Let $R_{12}^{(k)}$, $\tilde{R}_{12}^{(k)}$ satisfy the orthogonality condition (2.4) and for some $p \geq 0$ such that $k+p \leq \ell$, let $\text{approx}_s(\cdot)$, $k+1 \leq s \leq k+p$, only modify the block rows of $R_{12}^{(k)}$ with indices below $i_k$.

Then, setting $\lambda_{\text{max}}^{(k,k+p)} = \lambda_{\text{max}}(B_{k+p}^{-1} B_k)$ and $\lambda_{\text{max}}^{(k-1,k+p)} = \lambda_{\text{max}}(B_{k+p}^{-1} B_{k-1})$ and using similar notation for the minimal eigenvalues, there holds

\begin{align}
\lambda_{\text{max}}^{(k,k+p)} & \leq \lambda_{\text{max}}^{(k-1,k+p)} \leq \lambda_{\text{max}}^{(k,k+p)} + g(\lambda_{\text{max}}^{(k,k+p)}, \gamma_k), \\
\lambda_{\text{min}}^{(k,k+p)} & \geq \lambda_{\text{min}}^{(k-1,k+p)} \geq \lambda_{\text{min}}^{(k,k+p)} - g(\lambda_{\text{min}}^{(k,k+p)}, \gamma_k),
\end{align}

\section{Conclusion.}
where

(3.12) \[ \gamma_k = \left\| (\mathcal{R}^{(k)}_{12} - \tilde{\mathcal{R}}^{(k)}_{12}) \tilde{S}^{(k)}_B -1/2 \right\| < 1, \]

and

(3.13) \[ g(\lambda, \gamma) = \max_{\beta > 0} \frac{2\gamma \beta - |\lambda - 1|\beta^2}{\beta^2 + \lambda - 1}. \]

Moreover, if \( p = 0 \), right inequalities (3.10), (3.11) become equalities.

Proof. We prove inequalities only in (3.10); the proof of (3.11) follows the same lines. Considering first right inequality (3.10), we recall that \( B_{k-1}, B_k \) satisfy (3.4), (3.2), which amount to

\[
\begin{align*}
B_{k-1} &= \begin{pmatrix} \mathcal{R}^{(k)}_{11} & \mathcal{R}^{(k)}_{12} \\ \mathcal{R}^{(k)}_{12}^T & S^{(k)}_B \end{pmatrix} \begin{pmatrix} \mathcal{R}^{(k)}_{11} & \mathcal{R}^{(k)}_{12} \\ \mathcal{R}^{(k)}_{12}^T & I \end{pmatrix}, \\
B_k &= \begin{pmatrix} \mathcal{R}^{(k)}_{11} & \mathcal{R}^{(k)}_{12} \\ \mathcal{R}^{(k)}_{12}^T & \tilde{S}^{(k)}_B \end{pmatrix} \begin{pmatrix} \mathcal{R}^{(k)}_{11} & \tilde{\mathcal{R}}^{(k)}_{12} \\ \mathcal{R}^{(k)}_{12}^T & I \end{pmatrix},
\end{align*}
\]

where \( \mathcal{R}^{(k)}_{ij} \) is \( i_k \times i_k \), \( \mathcal{R}^{(k)}_{12} \) is \( i_k \times (n - i_k) \), and \( S^{(k)}_B, \tilde{S}^{(k)}_B \) are \( (n - i_k) \times (n - i_k) \), all dimensions being considered blockwise. Since the first \( i_k \) block rows are not modified during the following \( p \) steps, we also have

\[
B_{k+p} = \begin{pmatrix} \mathcal{R}^{(k)}_{11} & \mathcal{R}^{(k)}_{12} \\ \mathcal{R}^{(k)}_{12}^T & S \end{pmatrix} \begin{pmatrix} \mathcal{R}^{(k)}_{11} & \mathcal{R}^{(k)}_{12} \\ \mathcal{R}^{(k)}_{12}^T & I \end{pmatrix},
\]

for some \( S \) of order \( (n - i_k) \times (n - i_k) \). Note that both \( S \) and \( \tilde{S}^{(k)}_B \) have the same leading block, since this block is not modified during the steps \( k, \ldots, k + p \); hence

(3.14) \[ \lambda_{\max}(S^{-1} \tilde{S}^{(k)}_B) \geq 1. \]

Now, letting

\[ J = \begin{pmatrix} \mathcal{R}^{(k)}_{11}^{-1} & \mathcal{R}^{(k)}_{12}^{-1} \\ \mathcal{R}^{(k)}_{12}^T & I \end{pmatrix}, \]

the next equalities follow by direct computation:

\[
\begin{align*}
J^T B_k J &= \text{diag}(I, \tilde{S}^{(k)}_B), \\
J^T B_{k+p} J &= \text{diag}(I, S).
\end{align*}
\]

Using (3.14), this further entails

(3.16) \[ \lambda_{\max}^{(k, k+p)} = \lambda_{\max}(B_{k+p}^{-1} B_k) = \lambda_{\max} \left( (J^T B_{k+p}^{-1} J)^{-1} J^T B_k J \right) = \lambda_{\max}(S^{-1} \tilde{S}^{(k)}_B). \]

On the other hand, direct computation together with the use of (3.8) (following itself from (2.4)) for the bottom right block leads to

(3.17) \[ J^T B_{k-1} J = \begin{pmatrix} I & \mathcal{R}^{(k)}_{12} \tilde{\mathcal{R}}^{(k)}_{12} - \mathcal{R}^{(k)}_{12} \tilde{S}^{(k)}_B \\ \mathcal{R}^{(k)}_{12}^T & \tilde{\mathcal{R}}^{(k)}_{12} \tilde{S}^{(k)}_B \end{pmatrix}. \]
Hence, using this latter together with (3.15) and

$$
(3.18) \quad v_1^T \left( R_{12}^{(k)} - \tilde{R}_{12}^{(k)} \right) v_2 \leq \gamma_k \|v_1\| \sqrt{v_2^T S_B^{(k)} v_2},
$$

which follow from the definition (3.12) of $\gamma_k$, there holds

$$
\lambda_{\text{max}}^{(k-1, k+p)} = \max_{v} \frac{v^T J B_{k-1} J^T v}{v^T J B_{k+p} J^T v}
$$

(3.19)

$$
= \max_{v_1, v_2} \frac{v_1^T v_1 + v_2^T \tilde{S}_B^{(k)} v_2 + 2v_1^T \left( R_{12}^{(k)} - \tilde{R}_{12}^{(k)} \right) v_2}{v_1^T v_1 + v_2^T S v_2}
$$

(3.20)

$$
\leq \max_{\beta > 0} \frac{\beta^2 + 1 + 2\gamma_k \beta}{\beta^2 + \lambda_{\text{max}}^{-1} \left(S^{-1} \tilde{S}_B^{(k)}\right)},
$$

where we have set $\beta^2 = v_1^T v_1 \left( v_2^T \tilde{S}_B^{(k)} v_2 \right)^{-1}$. Right inequality (3.10) then follows from (3.20), (3.16) and the observation that

$$
\lambda + g(\lambda, \gamma) = \max_{\beta > 0} \frac{\beta^2 + 1 + 2\gamma \beta}{\beta^2 + \lambda^{-1}}
$$

for any $\lambda \geq 1$.

Now, left inequality (3.10) stems from (3.19) by setting $v_1 = 0$ and using (3.16). On the other hand, note that $J^T B_{k-1} J$ is SPD, and, therefore, it follows from (3.17) that the inequality (3.18) holds for some $\gamma_k < 1$.

Eventually, setting $p = 0$ we note that $S = \tilde{S}_B^{(k)}$, and, hence, the vectors $v_1, v_2$ that lead to an equality in (3.18) also allow one to reach an equality between (3.20) and (3.19). This is, however, the only approximation committed in the proof of right inequality (3.10), which therefore becomes an equality. $\square$

3.3. SSS bound. The next corollary provides via (3.21) a general condition number estimate based solely on the orthogonality assumption (2.3), (2.4). It corresponds to a repeated application of the above theorem in the case where $p = 0$. Note that since no additional assumption is made on the indices of the modified rows, the result may be applied to all of the modification patterns described in section 2.2; however, it suits the best the description of the SSS pattern, as this latter do not intend to leave some rows untouched. Now, this result may also be viewed as an extension of the Proposition 2.1 in [28] to more than one approximation step.

The corollary also introduces the parameter $\tau = \sum_{k=1}^{\ell} \gamma_k$ which, if bounded away from 1, implies via (3.22) a bounded condition number. Note that in most cases, the above condition may be relaxed by requiring that $\tau$ is bounded away from a small integer $c$. This is possible, for instance, if one may subdivide the interval $[0, \ell]$ into few, say, $c$, contiguous subintervals $[k_i, k_{i+1}]$, $i = 0, \ldots, c-1$, such that the corresponding partial sums $\sum_{k_i+1}^{k_{i+1}} \gamma_k$ are all bounded away from 1. Then, from

$$
\kappa(R^{-T} A R^{-1}) = \kappa(B_{\ell}^{-1} A) \leq \kappa(B_{\ell}^{-1} B_{k_{\ell-1}}^{-1}) \cdots \kappa(B_{k_1}^{-1} A)
$$

and the observation that each term may be bounded similarly to (3.22) it follows that the overall condition number remains bounded above.

**Corollary 3.4 (SSS bound).** Let $R$ be an upper triangular matrix obtained by applying Algorithm 2.2 to an SPD matrix $A$, with APPROX$_k(\cdot)$, $k = 1, \ldots, \ell$, such
that the orthogonality condition (2.4) is satisfied for matrices related by (2.3). Let $\gamma_k, k = 1, \ldots, \ell$, be given by (3.12) with $\tilde{S}^{(k)}_B$ defined by (3.3) and $\hat{R}^{(k)}_{12}, \tilde{R}^{(k)}_{12}$ defined as in Theorem 3.3.

Then

$$\kappa(R^{-T}AR^{-1}) \leq \prod_{k=1}^{\ell} \frac{1 + \gamma_k}{1 - \gamma_k}. \tag{3.21}$$

If $\ell = 1$, inequality (3.21) becomes an equality.

Moreover, if $\gamma := \sum_{k=1}^{\ell} \gamma_k < 1$, then

$$\kappa(R^{-T}AR^{-1}) \leq \frac{e^{\gamma \ell}}{1 - \gamma}. \tag{3.22}$$

Proof. We first prove inequality (3.21). For this, note that setting $p = 0$ in Theorem 3.3 entails $\lambda^{(k,k+p)}_{\max} = \lambda^{(k,k+p)}_{\min} = 1$, and, since $g(1, \gamma_k) = \gamma_k$, it follows from (3.10), (3.11) that

$$\lambda_{\max}(B^{-1}_kB_{k-1}) = 1 + \gamma_k, \tag{3.23}$$
$$\lambda_{\min}(B^{-1}_kB_{k-1}) = 1 - \gamma_k, \tag{3.24}$$

the equalities stemming from the last statement of the theorem. The repeated application of the above result further leads to

$$\lambda_{\max}(R^{-T}AR^{-1}) = \lambda_{\max}(B^{-1}_kB_0) \leq \prod_{k=1}^{\ell} \lambda_{\max}(B^{-1}_kB_{k-1}) = \prod_{k=1}^{\ell} (1 + \gamma_k),$$
$$\lambda_{\min}(R^{-T}AR^{-1}) = \lambda_{\min}(B^{-1}_kB_0) \geq \prod_{k=1}^{\ell} \lambda_{\min}(B^{-1}_kB_{k-1}) = \prod_{k=1}^{\ell} (1 - \gamma_k),$$

and the inequality (3.21) readily follows.

Eventually, observing that $1 + \gamma_k \leq e^{\gamma_k}$ and $(1 - \gamma_k)(1 - \gamma_s) \geq 1 - \gamma_k - \gamma_s$ for any $\gamma_k, \gamma_s > 0$, the estimate (3.22) follows directly from (3.21). □

### 3.4. One-level bound

Before we consider in more detail the case where $p > 0$, that is, the case where some block rows of the factor are not modified for several consecutive steps, we shed some light in Lemma 3.5 below on how the function $g(\lambda, \gamma)$ from our main theorem depends on $\lambda$. Considering first $\lambda \approx 1$, this lemma shows that $g(\lambda, \gamma) \approx \gamma \lambda$ (as may be concluded from the second term in the right-hand sides of (3.25), (3.26)). The estimates (3.10), (3.11) then amount to

$$\lambda^{(k-1,k+p)}_{\max} \leq (1 + \gamma_k)\lambda^{(k,k+p)}_{\max},$$
$$\lambda^{(k-1,k+p)}_{\min} \geq (1 - \gamma_k)\lambda^{(k,k+p)}_{\min},$$

where the notation for $\lambda^{(k,k+p)}_{\max} = \lambda_{\max}(B^{-1}_{k+p}B_k)$, $\lambda^{(k,k+p)}_{\min} = \lambda_{\min}(B^{-1}_{k+p}B_k)$ is the same as in Theorem 3.3. The contribution of each accuracy measure $\gamma_k$ to the bound on the condition number is then essentially the same as obtained by setting $p = 0$ (see Corollary 3.4).

On the other hand, if either $\lambda \gg 1$ ($\lambda^{(k,k+p)}_{\max}$ case) or $\lambda \ll 1$ ($\lambda^{(k,k+p)}_{\min}$ case), we observe that $g(\lambda, \gamma) \approx \gamma^2 \lambda / |\lambda - 1|$ (this follows from the first term in the right-hand sides of (3.25), (3.26)). In particular, if $\lambda = \lambda^{(k,k+p)}_{\min} \ll 1$, the lower bound (3.11) on
\( \lambda_{\text{min}}^{(k-1,k+p)} \) is essentially given by \((1 - \gamma_k^2)\lambda_{\text{min}}^{(k,k+p)} \), which compares favorably to the estimate \((1 - \gamma_k)\lambda_{\text{min}}^{(k,k+p)} \) from the above inequalities. The improvement is even more important when \( \lambda = \lambda_{\text{max}}^{(k,k+p)} > 1 \), since the upper bound (3.10) on \( \lambda_{\text{max}}^{(k-1,k,p)} \) then essentially corresponds to \( \lambda_{\text{max}}^{(k,k+p)} + \gamma_k^2 \) instead of \((1 + \gamma_k)\lambda_{\text{max}}^{(k,k+p)} \).

**Lemma 3.5.** Let \( \lambda \) and \( \gamma \leq 1 \) be real and positive, and \( g(\lambda, \gamma) \) be defined by (3.13). Then, \( g(1, \gamma) = \gamma \) and for \( \lambda \neq 1 \),

\[
(3.25) \quad g(\lambda, \gamma) \leq \min \left( \gamma^2 \frac{\lambda}{|\lambda - 1|}, \gamma \lambda \right),
\]

\[
(3.26) \quad g(\lambda, \gamma) \geq \max \left( c_1(\lambda, \gamma) \cdot \gamma^2 \frac{\lambda}{|\lambda - 1|}, c_2(\lambda, \gamma) \cdot \gamma \lambda \right),
\]

where \( c_1(\lambda, \gamma) = |\lambda - 1|^2(\lambda - 1 + \gamma^2 \lambda)^{-1} \to 1 \) for either \( \lambda \to 0 \) or \( \lambda \to \infty \) and \( c_2(\lambda, \gamma) = (2 - |\lambda - 1|/\gamma)(2 + \lambda - 1)^{-1} \to 1 \) for \( \lambda \to 1 \).

**Proof.** The proof of \( g(1, \gamma) = \gamma \) is straightforward. Now, setting \( \bar{\beta} = \gamma|\lambda - 1|^{-1} \), the first term in the maximum (3.26) follows from

\[
g(\lambda, \gamma) = \max_{\beta > 0} \frac{2\gamma\beta - |\lambda - 1|\beta^2}{\beta^2 + 1} \geq \frac{2\gamma\bar{\beta} - |\lambda - 1|\bar{\beta}^2}{\bar{\beta}^2 + 1} = \frac{\gamma^2\lambda|\lambda - 1|}{|\lambda - 1|^2 + \gamma^2 \lambda}.
\]

Using the same reasoning with \( \bar{\beta} = 1 \) instead leads to the second term.

To prove (3.25), note that \( 2\gamma\beta - |\lambda - 1|\beta^2 \leq \gamma^2|\lambda - 1|^{-1} \) holds for all \( \beta \), and the first term in the minimum (3.25) follows from

\[
g(\lambda, \gamma) = \max_{\beta > 0} \frac{2\gamma\beta - |\lambda - 1|\beta^2}{\beta^2 + 1} \leq \gamma^2 \max_{\beta > 0} \frac{|\lambda - 1|^{-1}}{\beta^2 + 1} = \gamma^2 \frac{\lambda}{|\lambda - 1|}.
\]

Noting that \( |\lambda - 1| \geq \gamma(1 - \lambda) \), the second term follows from

\[
g(\lambda, \gamma) = \max_{\beta > 0} \frac{2\gamma\beta - |\lambda - 1|\beta^2}{\beta^2 + 1} \leq \max_{\beta > 0} \frac{2\gamma\beta - \gamma(1 - \lambda)\beta^2}{\beta^2 + 1} = \frac{\lambda}{|\lambda - 1|} \geq \gamma \lambda.
\]

Now, we make the above discussion more specific by assuming that only the block row of the factor computed during the step \( k \) is modified during this step. This setting corresponds to the one-level pattern from section 2.2. Corollary 3.6 then shows (see (3.27)) that the condition number is bounded (up to a “penalization” factor \( 2 + \gamma^2 \ell \)) by a quotient of two quantities, one being essentially a sum of \( \gamma_k^2 \), the other roughly corresponding (if \( c_* \approx 1 \)) to a product of \( 1 - \gamma_k^2 \) (with, however, \( 1 - \gamma_{\text{max}}^2 \) instead of \( 1 - \gamma_k^2 \)). Note that, although this estimate is asymptotically better, it may be less accurate than (3.21) for small \( \ell \); in particular, it is always less accurate for \( \ell = 1 \).

Next, the inequality (3.28) further highlights the role played by the parameter \( \tilde{\gamma}^2 \ell = \sum_{k=1}^\ell \gamma_k^2 \); if this latter is bounded away from 1, then the condition number is also bounded. Note that this condition is less restrictive than the one required by Corollary 3.4, namely, that \( \tilde{\gamma} \ell = \sum_{k=1}^\ell \gamma_k \) should be away from 1; this comes with the additional assumption on the pattern of the modified rows. On the other hand, as with Corollary 3.4, the requirement of \( \tilde{\gamma}^2 \ell \) being bounded away from one may often be relaxed to \( \tilde{\gamma}^2 \ell < c \) for some small \( c > 1 \) by using essentially the same arguments.
Eventually, we note that although the corollary below provides some insight on how the condition number behaves in the one-level case, we still advocate the direct use of eigenvalue bounds (3.10), (3.11) if one needs to obtain an accurate estimate of the condition number; in the one-level case, this amounts to computing, for instance, \( \tilde{\lambda}_{\max}(\ell, \ell) \) using \( \tilde{\lambda}_{\max}(1, \ell) = 1 \) and \( \gamma_\ell \), and then computing \( \tilde{\lambda}_{\max}(\ell-2, \ell) \) using \( \tilde{\lambda}_{\max}(\ell-1, \ell) \) and \( \gamma_{\ell-1} \), and so on, until \( \lambda_{\max}(R^{-T}AR^{-1}) = \tilde{\lambda}_{\max}(0, \ell) \) is obtained. This procedure is used, in particular, for the numerical experiments in section 4. Note that the use of tilde emphasizes that \( \tilde{\lambda} \) is just a bound on the actual eigenvalue \( \lambda \).

**Corollary 3.6** (one-level bound). Let the assumption of Corollary 3.4 hold. In addition, let \( \ell = n - 1, \ i_k = k, \) and APPROX\(_k(\cdot) \) only modify the block row \( i_k, \ k = 1, \ldots, \ell \).

Then, setting \( \tilde{\gamma}^2 \ell := \sum_{k=1}^{\ell} \gamma_k^2 \) and \( \gamma_{\max} = \max_{1 \leq k \leq \ell} \gamma_k \), there holds

\[
\kappa(R^{-T}AR^{-1}) \leq \left(2 + \tilde{\gamma}^2 \ell \right) \cdot \frac{\left(1 + \sqrt{\tilde{\gamma}^2 \ell}\right)^2}{\left(1 - \gamma_{\max}^2\right) \prod_{k=2}^{\ell} \left(1 - \frac{\gamma_k^2}{c}\right)},}
\]

where \( c_\ast = (1 + \tilde{\gamma}^2 \ell)/(1 + \tilde{\gamma}^2 \ell + 1 - \gamma_{\max}^2) \to 1 \) when \( \tilde{\gamma}^2 \ell \to \infty \).

Moreover, if \( \tilde{\gamma}^2 \ell < 1 \), then

\[
\kappa(R^{-T}AR^{-1}) \leq \left(\frac{1 + \sqrt{\tilde{\gamma}^2 \ell}}{1 - \sqrt{\tilde{\gamma}^2 \ell}}\right)^2.
\]

**Proof.** We first show that the results follow from

\[
\lambda_{\max}(R^{-T}AR^{-1}) = \lambda_{\max}^{(0, \ell)} < \left(1 + \sqrt{\tilde{\gamma}^2 \ell}\right)^2,
\]

\[
\lambda_{\min}(R^{-T}AR^{-1}) = \lambda_{\min}^{(0, \ell)} \geq \max_{1 \geq c \geq \gamma_{\max}} \left(1 - c\right) \prod_{k=2}^{\ell} \left(1 - \frac{\gamma_k^2}{c}\right),
\]

proving these inequalities later. The estimate (3.27) follows from (3.29), (3.30) setting \( c = c_\ast \) and using

\[
1 - c_\ast \geq \frac{1 - \gamma_{\max}^2}{1 + \tilde{\gamma}^2 \ell + 1 - \gamma_{\max}^2} > \frac{1 - \gamma_{\max}^2}{2 + \tilde{\gamma}^2 \ell} > 0.
\]

This latter further shows that \( c_\ast < 1 \), whereas \( c_\ast \geq \gamma_{\max} \) follows from

\[
c_\ast - \gamma_{\max} = (1 - \gamma_{\max}) \frac{1 - \gamma_{\max} + \tilde{\gamma}^2 \ell - \gamma_{\max}^2}{2 + \tilde{\gamma}^2 \ell - \gamma_{\max}^2} > 0.
\]

On the other hand, estimate (3.28) is obtained setting \( c_\ast = (\tilde{\gamma}^2 \ell)^{1/2} > \gamma_{\max} \). Note that \( c_\ast < 1 \) by assumption, whereas (3.30) further implies

\[
\lambda_{\min}(R^{-T}AR^{-1}) \geq (1 - c_\ast) \prod_{k=1}^{\ell} \left(1 - \frac{\gamma_k^2}{c_\ast}\right) \geq (1 - c_\ast) \left(1 - \sum_{k=1}^{\ell} \frac{\gamma_k^2}{c_\ast}\right) = \left(1 - \frac{\gamma_{\max}^2}{c_\ast}\right)^2.
\]

Now, note that the assumptions on APPROX\(_k(\cdot) \) made here satisfy the requirements of Theorem 3.3 for all possible values of \( k \) and \( p \), that is, for all \( 1 \leq k \leq \ell \) and \( 0 \leq p \leq \ell - k \).
We begin with the proof of (3.29). First observe that the upper bound in (3.10) is an increasing function of \( \lambda_{\max}^{(k, k+p)} \). Hence, setting \( \tilde{\lambda}_{\max}^{(\ell, \ell)} = b > 1 \) (instead of 1) and applying the recursion (3.10) to define the remaining \( \lambda_{\max}^{(k, \ell)} \), \( k = 1, \ldots, \ell \), as

\[
\lambda_{\max}^{(k-1, \ell)} = \lambda_{\max}^{(k, \ell)} + g(\lambda_{\max}^{(k, \ell)}, \gamma_k),
\]

one concludes that \( \tilde{\lambda}_{\max}^{(0, \ell)} > \lambda_{\max}^{(0, \ell)} \). On the other hand, (3.31) also entails

\[
\tilde{\lambda}_{\max}^{(0, \ell)} \geq \cdots \geq \tilde{\lambda}_{\max}^{(k, \ell)} = b,
\]

which, together with (3.31) and \( g(\lambda, \gamma) \leq \gamma^2 \lambda |\lambda - 1|^{-1} \) (as follows from (3.25)) implies

\[
\tilde{\lambda}_{\max}^{(k-1, \ell)} \leq \tilde{\lambda}_{\max}^{(k, \ell)} + \gamma_k^2 \frac{b}{b-1},
\]

and hence

\[
\lambda_{\max}^{(0, \ell)} \leq \tilde{\lambda}_{\max}^{(0, \ell)} \leq b + \frac{b}{b-1} \sum_{k=1}^{\ell} \gamma_k^2 = b + \frac{b}{b-1} \gamma^2 \ell.
\]

Setting \( b = 1 + (\gamma^2 \ell)^{1/2} \) finishes the proof of (3.29). One may further check that this choice of \( b \) maximizes the bound in (3.32).

The proof for the lower bound (3.30) follows similar lines. First, set \( \tilde{\lambda}_{\min}^{(\ell-1, \ell)} = 1 - c \) and observe that \( \lambda_{\max}^{(\ell-1, \ell)} \geq \lambda_{\min}^{(\ell-1, \ell)} \) since \( c \geq \max_{k=1, \ldots, \ell} \gamma_k \). Further, define

\[
\tilde{\lambda}_{\min}^{(k-1, \ell)} = \lambda_{\min}^{(k, \ell)} - g(\lambda_{\max}^{(k, \ell)}, \gamma_k)
\]

with, hence, \( \lambda_{\max}^{(0, \ell)} \geq \tilde{\lambda}_{\max}^{(0, \ell)} \) and \( \tilde{\lambda}_{\min}^{(k, \ell)} \leq 1 - c, k = 1, \ldots, \ell - 1 \). Using this latter together with \( g(\lambda, \gamma) \leq \gamma^2 \lambda |\lambda - 1|^{-1} \) (see (3.25)) entails

\[
\tilde{\lambda}_{\min}^{(k-1, \ell)} \geq \tilde{\lambda}_{\min}^{(k, \ell)} \left( 1 - \frac{\gamma_k^2}{c} \right),
\]

and hence

\[
\lambda_{\max}^{(0, \ell)} \geq \tilde{\lambda}_{\max}^{(0, \ell)} \geq (1 - c) \prod_{k=2}^{\ell} \left( 1 - \frac{\gamma_k^2}{c} \right),
\]

and the inequality (3.30) follows. \( \square \)

### 3.5. HSS pattern

Regarding the HSS pattern, we note that the derivation of an analytical bound similar to (3.27) seems more involved and would perhaps provide less insight. However, by analogy to the one-level case, such a bound may be computed numerically, using again the estimates from Theorem 3.3. Since a proper use of these latter (which allows a better estimate than that given by (3.21)) is less straightforward in the HSS case, we outline the main ideas below. To begin with, observe that the indices \( \mathcal{P}_k \) of block rows approximated during the step \( k \) of Algorithm 2.2 (and associated to the node \( k \) in the corresponding HSS tree; see, e.g., Figure 2.3) are modified again only during the parent step, that is, during the step parent(\( k \)) associated to the parent of the node \( k \) in the tree. Hence, the estimates (3.10), (3.11) may be applied for this \( k \) with \( p \leq \text{parent}(k) - k - 1 \). This observation motivates the recursive procedure summarized in Algorithm 3.1, which yields an upper bound \( \tilde{\lambda}_{\max}^{(k-1, k+s)} \) on \( \lambda_{\max}^{(k-1, k+s)} \) for any \( k \geq 1 \) and \( s \geq -1 \). (The algorithm for the lower bound \( \tilde{\lambda}_{\min}^{(k-1, k+s)} \) is similar, with line 5 based on (3.11) instead of (3.10).)
Algorithm 3.1. (HSS Bound on $\lambda_{\text{max}}^{(k-1,k+s)}$): $\tilde{\lambda}_{\text{max}}^{(k-1,k+s)} = \text{cond} (k-1, k+s)$.

1. if $(s = -1)$: return 1
2. if $(s = 0)$: return $1 + \gamma_k$
3. $p = \min(\text{parent}(k) - k - 1, s)$
4. $\tilde{\lambda}_{\text{max}}^{(k,k+p)} = \text{cond} (k, k+p)$
5. $\tilde{\lambda}_{\text{max}}^{(k-1,k+p)} = \tilde{\lambda}_{\text{max}}^{(k,k+p)} + g(\tilde{\lambda}_{\text{max}}^{(k,k+p)}, \gamma_k)$
6. $\tilde{\lambda}_{\text{max}}^{(k+p,k+s)} = \text{cond} (k+p, k+s)$
7. return $\frac{\tilde{\lambda}_{\text{max}}^{(k-1,k+p)}}{\tilde{\lambda}_{\text{max}}^{(k,k+s)}}$

3.6. Block-diagonal preconditioner. We now compare the above results with the existing analysis of the preconditioners given by the block-diagonal part of $A$, as found in [1, Chapter 9]. Assuming that $A$ has a block partitioning (2.1), this latter analysis relies at step $k$ on the $2 \times 2$ partitioning of $A^{(k)} = A_{k,n,k,n}$ given by

$$A^{(k)} = \begin{pmatrix} A_{k,k} & A_{k,k+1:n} \\ A_{k,k+1:n}^T & A^{(k+1)} \end{pmatrix}.$$

The corresponding block-diagonal preconditioner is then defined by

$$D_k = \begin{pmatrix} A_{k,k} \\ A^{(k+1)} \end{pmatrix},$$

and it is known that

$$\kappa(D_k^{-1}A^{(k)}) = \frac{1 + \gamma_k^{\text{CBS}}}{1 - \gamma_k^{\text{CBS}}},$$

where

$$\gamma_k^{\text{CBS}} = \max_{v_1, v_2} \frac{v_1^T A_{k,k+1:n} v_2}{(v_1^T A_{k,k} v_1 \cdot v_2^T A^{(k+1)} v_2)^{1/2}}$$

is the CBS constant for the partitioning.

On the other hand, we observed in section 2.2 that this block-diagonal preconditioner may be obtained with Algorithm 2.2 by setting to zero the approximated part of the factor, and that it corresponds to the one-level modification pattern. Therefore, combining Lemma 3.2(c) and the fact that at step $k$ the off-diagonal part of the first $k$ block rows is set to zero yields

$$B_k = \text{blockdiag} \begin{pmatrix} A_{1,1}, \cdots, A_{k,k}, A^{(k+1)} \end{pmatrix},$$

and hence

$$\frac{1 + \gamma_k}{1 - \gamma_k} = \kappa(B_k^{-1}B_{k-1}) = \kappa(D_k^{-1}A^{(k)}) = \frac{1 + \gamma_k^{\text{CBS}}}{1 - \gamma_k^{\text{CBS}}},$$

where the first equality follows from the Corollary 3.4 in the case where $\ell = 1$. Clearly, this is only possible if $\gamma_k = \gamma_k^{\text{CBS}}$, and therefore the two parameters are identical for
all $k$. Note that the repeated use of the $2 \times 2$ bound (3.34) leads to the same estimate as (3.21). However, the asymptotically sharper bound (3.27) is also applicable here, and an even sharper estimate may be obtained by repeated application of (3.10), (3.11) with $p = \ell - k$, both estimates requiring the same CBS constants. To the best of our knowledge, these improved bounds are presented here for the first time.

### 3.7. Sharpness.

Note that one can hardly find a better estimate for the one-level case than the one based on (3.10), (3.11), and which would only involve individual accuracy measures $\gamma_k$, $k = 1, \ldots, \ell$, since this latter estimate is sharp. More precisely, Theorem 3.7 below states that for any set $\gamma_k$, $k = 1, \ldots, \ell$, of positive values smaller than 1 there is a matrix $A = A(\gamma_1, \ldots, \gamma_\ell)$ for which the incomplete Cholesky factorization algorithm performs approximations with accuracies $\gamma_k$ and produces a sequence $B_1, \ldots, B_\ell$ of auxiliary matrices such that right inequalities (3.10), (3.11) simultaneously become equalities for all $k$. In other words, for any possible bound which may be obtained using (3.10), (3.11) there is a matrix which allows to reach this bound.

Now, Theorem 3.7 is formulated in the context of block-diagonal preconditioners, that is, it assumes that Algorithm 2.2 is considered with $\text{APPROX}_k(\cdot) = O$. This shows that the sharpness property holds for this subclass of one-level preconditioners. The above assumption on the approximation operation is, however, not restrictive in practice since known approximation procedures, and in particular those based on the orthogonal decomposition as described in section 2.3, amount to all-zero approximation if the corresponding threshold is chosen small enough.

**THEOREM 3.7.** For any set of positive values $\tilde{\gamma}_k < 1$, $k = 1, \ldots, \ell$, there exist an SPD matrix $A = A(\tilde{\gamma}_1, \ldots, \tilde{\gamma}_\ell)$ partitioned as in (2.1) such that the application of Algorithm 2.2 to $A(\tilde{\gamma}_1, \ldots, \tilde{\gamma}_{\ell-1})$ with $\text{APPROX}_k(\cdot) = O$, $k = 1, \ldots, \ell$, returns an upper triangular $R$ and there holds the following:

(a) $\gamma_k = \tilde{\gamma}_k$ for $k = 1, \ldots, \ell$,

where $\gamma_k$ is given by (3.12) with $\tilde{S}_B^{(k)}$ defined by (3.3) and $\tilde{R}_{12}^{(k)}$, $\tilde{R}_{12}^{(k)}$ defined as in Theorem 3.3.

(b) Setting $\lambda_{\max}^{(k,\ell)} = \lambda_{\max}(B_1^{-1}B_k)$ and $\lambda_{\min}^{(k,\ell)} = \lambda_{\min}(B_1^{-1}B_k)$, where $B_k$, $k = 1, \ldots, \ell$, is defined as in section 3.1, there holds

$$
\lambda_{\max}^{(k-1,\ell)} = \lambda_{\max}^{(k,\ell)} + g(\lambda_{\max}^{(k,\ell)}, \tilde{\gamma}_k),
$$

$$
\lambda_{\min}^{(k-1,\ell)} = \lambda_{\min}^{(k,\ell)} - g(\lambda_{\min}^{(k,\ell)}, \tilde{\gamma}_k).
$$

**Proof.** The proof of this theorem may be found in [21].

On the other hand, the following counterexample shows that the condition number estimate obtained in the one-level case by repeated application of (3.10), (3.11) with $p = \ell - k$, $k = 1, \ldots, \ell$, may not hold assuming only the orthogonality of the dropping (2.3), (2.4), that is, the additional assumption on the indices of modified block rows (as made for the one-level case) is really necessary. The counterexample also demonstrates that a system preconditioned by the incomplete Cholesky factorization may in principle be more ill-conditioned than the original system; compare $\kappa(A) = 7$ with $\kappa(R^{-1}AR^{-1}) \approx 8.61$.

**COUNTEREXAMPLE 3.1.** Let $i_1 = 1$, $i_2 = 2$, $n = 3$, and $\ell = 2$. Set

$$
A = \begin{pmatrix}
1 & .4 & .4 \\
.4 & 1 & -.4 \\
.4 & -.4 & 1
\end{pmatrix}, \quad B_1 = \begin{pmatrix}
1 & & \\
& 1 & - .4 \\
& -.4 & 1
\end{pmatrix},
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
and $B_2 = R^T R$ with

$$R = \begin{pmatrix}
1 & -1.16 \\
-1 & -1.32 \\
\sqrt{.872} & \end{pmatrix}.$$

One may check that $\gamma_1^2 = 8/15$ and $\gamma_2^2 = 4/109$ and hence

$$\tilde{\lambda}_{\text{max}} = 1 + \gamma_2 + g(1 + \gamma_2; \gamma_1) \lesssim 1.9,$$

$$\tilde{\lambda}_{\text{min}} = 1 - \gamma_2 - g(1 - \gamma_2; \gamma_1) \gtrsim .24,$$

whereas

$$\kappa(R^{-T}AR^{-1}) = \kappa(B_2^{-1}A) \gtrsim 8.61 > 7.92 \gtrsim \frac{\tilde{\lambda}_{\text{max}}}{\tilde{\lambda}_{\text{min}}}.$$

### 3.8. $\gamma_k$ estimates

Eventually, we provide some practical upper bounds for the accuracy measure $\gamma_k$. One obvious way to obtain such a bound is to split the norm of the product of two factors in (3.12) into a product of two norms:

$$\gamma_k \leq \left\| R^{(k)}_{12} - \tilde{R}^{(k)}_{12} \right\| \left\| S^{(k)}_B^{-1/2} \right\|.$$

As noted in section 2.3, the first factor may be easily controlled by imposing a given threshold $\text{tol}_a$. However, the impact of the second factor is less obvious, since $S^{(k)}_B$ incorporates all the approximations made in the incomplete Cholesky algorithm before and during the step $k$. The next theorem is helpful in this respect, since it relates $S^{(k)}_B$ to the corresponding Schur complement $S^{(k)}_A$ of $A$, as well as the bottom right subblock of $A$.

**Theorem 3.8.** Let $A$ be SPD and partitioned as in (2.1). Let $S^{(k)}_B$ be defined by (3.3), and $R^{(k)}_{12}$, $\tilde{R}^{(k)}_{12}$ be as in Theorem 3.3.

Then

$$\mathbf{v}^T S_A^{(k)} \mathbf{v} \leq \mathbf{v}^T S^{(k)}_B \mathbf{v} \leq \mathbf{v}^T A_{i_k+1:n,i_k+1:n} \mathbf{v} \quad \forall \mathbf{v}, \quad k = 1, \ldots, \ell,$$

where $S^{(k)}_A = A_{i_k+1:n,i_k+1:n} - A_{i_k+1:n,i_k+1:n}^T A_{i_k+1:n,i_k+1:n} A_{i_k+1:n,i_k+1:n}$. Moreover,

$$\gamma_k \leq \left\| R^{(k)}_{12} - \tilde{R}^{(k)}_{12} \right\| \left\| S^{(k)}_A^{-1/2} \right\| \leq \left\| R^{(k)}_{12} - \tilde{R}^{(k)}_{12} \right\| \| A^{-1} \|^{1/2}.$$

**Proof.** We prove left inequality (3.38) by induction. Let $S^{(k)}_B$ be defined by (3.3) and note that for $k = 1$ there holds $\mathbf{v}^T S^{(1)}_B \mathbf{v} \geq \mathbf{v}^T S^{(1)}_A \mathbf{v}$, where the first inequality stems from (3.8) and the equality follows since by (3.4) $S^{(1)}_B$ is a Schur complement of $B_0 = A$.

Now, assume that (3.38) holds for a given $k$. First, note that both $S^{(k+1)}_B$ and $\tilde{S}^{(k)}_B$ are the Schur complements of $B_k$, $k = 1, \ldots, \ell - 1$, as follows from (3.4) and (3.2), respectively. Hence, according to Lemma 3.1(c), $S^{(k+1)}_B$ is a Schur complement of $S^{(k)}_B$. Similarly, $S^{(k+1)}_A$ is a Schur complement of $S^{(k)}_A$. Next, note that for two SPD matrices $A$ and $B$, $\mathbf{w}^T A \mathbf{w} \leq \mathbf{w}^T B \mathbf{w}$ for all $\mathbf{w}$ implies $\mathbf{w}^T A^{-1} \mathbf{w} \geq \mathbf{w}^T B^{-1} \mathbf{w}$ for
all \( \mathbf{v} \), and (using Lemma 3.1(c)) the same relations holds for their respective Schur complements. Hence, \( \mathbf{v}^T S^{(k+1)}_A \mathbf{v} \leq \mathbf{v}^T S^{(k+1)}_B \mathbf{v} \) follows from left inequality (3.38), verified by induction. On the other hand, (3.8) implies \( \mathbf{v}^T S^{(k+1)}_B \mathbf{v} \leq \mathbf{v}^T S^{(k+1)}_D \mathbf{v}, \) and the combination of both inequalities yields the assertion for \( k + 1 \).

Now, right inequality (3.38) follows from (3.3), whereas left inequality (3.39) stems from (3.37) together with left inequality (3.38) and \( \| S^{1/2} \| = \| S \|^{1/2} \) for any SPD \( S \). Eventually, the right inequality (3.39) follows from

\[
\| S^{(k)}_A \|^{-1} \leq \| A^{-1} \|
\]

which itself stems from Lemma 3.1(c).

4. Numerical experiments.

4.1. Model problem. We consider the linear system arising from the five-point finite difference discretization of

\[
\begin{align*}
-\Delta u + \varepsilon u &= f & \text{in } \Omega = (0,1)^2, \\
\frac{\partial u}{\partial n} &= 0 & \text{on } \partial \Omega
\end{align*}
\]

on a uniform grid \( \Omega_h \) of mesh size \( h = 1/(N - 1) \). Let the grid be partitioned into three disjoint subsets: two half-squares and a horizontal line separating them, defined as

\[
\begin{align*}
\Omega_h^{(I)} &= \{ (ih,jh) \mid 0 \leq i \leq N - 1, 0 \leq j < \lfloor N/2 \rfloor \}, \\
\Omega_h^{(II)} &= \{ (ih,jh) \mid 0 \leq i \leq N - 1, \lfloor N/2 \rfloor < j \leq N - 1 \}, \\
\Omega_h^{(Γ)} &= \{ (ih,jh) \mid 0 \leq i \leq N - 1, j = \lfloor N/2 \rfloor \}.
\end{align*}
\]

Since the subsets \( \Omega_h^{(I)} \) and \( \Omega_h^{(II)} \) are disconnected, ordering unknowns in \( \Omega_h^{(I)} \) first, those in \( \Omega_h^{(II)} \) next, and those in \( \Omega_h^{(Γ)} \) last, the \( N^2 \times N^2 \) system matrix is given by

\[
A_\Omega = \begin{pmatrix}
A_I & A_{I,Γ} \\
A_{I,Γ}^T & A_{Γ}^T
\end{pmatrix}.
\]

If \( \varepsilon > 0 \) the matrix \( A_\Omega \) is symmetric and strictly diagonally dominant; hence, it is SPD.

Often, the unknowns corresponding to \( \Omega_h^{(I)} \), \( \Omega_h^{(II)} \) are further eliminated. This happens, for instance, prior to the last stage of the (exact) Cholesky factorization method based on nested dissection \([11,19]\). The \( N \times N \) system matrix

\[
A = A_Γ - A_{Γ,Γ}^{-1} A_{I,Γ} - A_{Γ,Γ}^{-1} A_{Γ,Γ} A_{Γ}^{-1} A_{I,Γ}
\]

of the resulting reduced system then corresponds to the Schur complement of \( A_Ω \) with respect to its bottom rightmost block. It follows from Lemma 3.1(b) that \( A \) is also SPD if \( \varepsilon > 0 \).

Note that \( A \) is usually not sparse and its Cholesky factorization requires \( \mathcal{O}(N^3) \) operations. An iterative solution may therefore be an attractive alternative for large \( N \). In particular, the HSS and SSS variants of Algorithm 2.2, as observed in section 2.2, only require \( \mathcal{O}(r_{\max} N^2) \) operations to construct the preconditioner \( B_\ell = R^T R \) of \( A \) and, as we shall see below, the maximal approximation rank \( r_{\max} \) remains
nicely bounded. Hence, if the condition number of the resulting system is bounded
as well, the overall complexity is also\(^1\) \(O(r_{\max}N^2)\). Note that the preconditioner for
the original (i.e., nonreduced) system may be chosen as

\[
R_\Omega = \begin{pmatrix}
R_I & R_{\Omega I}^{-T}A_{I,\Gamma} \\
R_{\Omega II} & R_{\Omega II}^{-T}A_{II,\Gamma}
\end{pmatrix}
\]

with \(R_I, R_{\Omega II}\) being upper triangular and such that \(R_I^T R_I = A_I, R_{\Omega II}^T R_{\Omega II} = A_{\Omega II}\). In this
case

\[
\kappa(R_{\Omega II}^{-T}A_{\Omega I}R_{\Omega II}^{-1}) = \kappa(R^{-T}AR^{-1}),
\]

and, hence, our conditioning analysis for the reduced system also applies to the original
one.

4.2. Fixed threshold experiments. We now investigate how accurately the
bounds derived in section 3 reproduce the condition number for the preconditioners
described by Algorithm 2.2. More precisely, we consider the HSS, SSS, and one-level
variants of the algorithm, as well as the algorithm from [28]; \textsc{approx} operation
\(\text{corresponds in all cases to the truncated SVD decomposition with relative threshold}
\(tol_r\), as described in section 2.3.

The matrix \(A\) is subdivided into the \(n \times n\) block form (2.1) with block size 10
and, hence, with \(N = 10n\). Next, to use HSS variant with binary trees as described
in section 2.2, we set \(n = 2^t + 1\), where \(t\) is the tree depth. Here we consider \(t\) from 0
to 6, which corresponds to matrix sizes \(N\) ranging from 20 to 650; the size \(N^2\) of the
unreduced matrix \(A_{\Omega}\) is then between 400 and 422,500. For every such \(N^2\) we list
in Table 4.1 the number of approximation steps \(\ell\) performed during the factorization;
this number is approximately two times larger for HSS variant and the method in
[28] compared to SSS and one-level variants. Further, a preconditioner is especially
needed when \(A\) is strongly ill-conditioned; this is achieved by setting \(\varepsilon = 10^{-4}\) with
\(\kappa(A)\) then ranging from 1.2 \(10^6\) to 3.7 \(10^7\).

Now, we report in Figure 4.1 the exact condition number for the considered
preconditioners as well as the corresponding upper bounds; this is done for different
values of unreduced system size \(N^2\) and of relative dropping threshold \(tol_r\). More
precisely, the bound for the one-level variant is computed by repeatedly applying
\((3.10), (3.11)\) with \(p = \ell - k\), whereas inequality \((3.21)\) is used for the SSS variant
and Algorithm 3.1 (with \(k = 1\) and \(k + s = \ell\)) for the HSS one. In all cases, the
values of \(\gamma_k, k = 1, \ldots, \ell\), are computed using the definition \((3.12)\). The figure also
highlights the parameters \(\overline{\gamma}_\ell = \sum_{k=1}^{\ell} \gamma_k\) (from Corollary 3.4) and \(\overline{\gamma}_2^2\ell = \sum_{k=1}^{\ell} \gamma_k^2\)
(from Corollary 3.6) for different \(tol_r\) and \(N^2\). On the other hand, maximal and
minimal ranks for a given threshold are reported in Table 4.2.

\(^1\)This comes with the fact that one application of the preconditioner requires at most \(O(N^2)\),
even for the exact Cholesky factor. For HSS and SSS variants the complexity is linear in \(N\).

Table 4.1

<table>
<thead>
<tr>
<th>(N^2)</th>
<th>400</th>
<th>900</th>
<th>2500</th>
<th>8100</th>
<th>28900</th>
<th>108900</th>
<th>422500</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-level/SSS</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>HSS/[28]</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>15</td>
<td>31</td>
<td>63</td>
<td>127</td>
</tr>
</tbody>
</table>

Downloaded 08/02/13 to 93.74.23.116. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php
First, we note that the condition numbers of all the preconditioners remain close to each other; for the same problem size and threshold value they differ at most by a factor of 11; this factor reduces to 5 for the largest size. Remarkably, this holds for SSS/HSS/[28] factorizations, but also for a simple one-level method; in some sense this situation is similar to multigrid methods [24, 18], for which a simple two-grid scheme (analog of a one-level variant here) behaves similarly to more practical multigrid methods (alike SSS/HSS/[28] here). Note that the method from [28] leads
Table 4.2
Maximal and minimal ranks for all considered values of $N$.

<table>
<thead>
<tr>
<th>$\text{tol}_r$</th>
<th>$10^{-1}$</th>
<th>$10^{-3}$</th>
<th>$10^{-1}$</th>
<th>$10^{-3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank</td>
<td>2–2</td>
<td>3–5</td>
<td>2–4</td>
<td>3–10</td>
</tr>
</tbody>
</table>

to the rank values that are double those in the other cases; this is mainly due to the fact that the matrices approximated during each step are then also larger.

Second, the one-level estimate follows closely the corresponding condition number, even for large values of $N^2$, where the number $\ell$ of approximation steps is large. On the other hand, the estimates for the other two approaches are less accurate, especially when the corresponding condition number is away from 1. Note that the HSS bound obtained with Algorithm 3.1 is close to the SSS bound, despite the fact that the former incorporates a larger number $\ell$ of approximations; this comes with additional assumptions in HSS case on the indices of the modified rows. Eventually, pushing further the multigrid analogy, we note that the one-level condition estimate reproduces correctly the convergence behavior of the other approaches and therefore may be used to estimate their convergence rate; it is similar to the two-grid analysis in this respect. This observation is so far limited to the considered model problem and may not hold in full generality.

Now, regarding the accuracy parameters $\overline{\gamma}_\ell$ and $\widetilde{\gamma}_\ell$ we note that, as suggested by the analysis in section 3 (see the comments for Corollaries 3.4 and 3.6), the condition numbers and the related estimates of, respectively, the SSS, HSS, and one-level variants remain nicely bounded if these parameters remain below or around 1.

4.3. Adaptive threshold strategies. Another observation illustrated with Figure 4.1 is that the condition numbers, their estimates, and the corresponding accuracy parameters $\overline{\gamma}_\ell$, $\widetilde{\gamma}_\ell$ tend to increase with $N$, independently of the value $\text{tol}_r$ of the truncation threshold. In the case of accuracy parameters, this growth has two contributions: the increasing number $\ell$ of the approximation steps as well as the increase in the norm of $\widetilde{S}_B^{(k)}$ which enters the definition (3.12) of $\gamma_k$.

Let us now determine the conditions under which the condition number remains bounded independently of $N$. Clearly, if we require

$$\gamma_k \leq \frac{c}{\ell}$$

for some $c < 1$, then $\overline{\gamma}_\ell \leq c$ and the SSS estimate (3.22) will remain bounded. As mentioned in section 3, relaxing the requirement on $c$ to $c = O(1)$ should still guarantee a bounded condition number for all the considered methods. Now, to estimate $\gamma_k$ we use right inequality (3.39), where, as shown in [21, Appendix A],

$$\|A^{-1}\| \leq c_N^{-1} \cdot \frac{N - 1}{\varepsilon}$$

with $c_N \to \frac{1}{2} e^{-4\sqrt{\varepsilon}}$ for $N \to \infty$; for simplicity, we use $c_\infty$ instead of $c_N$. Hence, combining (3.39), (2.7) with the above inequalities (4.7), (4.8) and using the fact that $\ell = O(N)$ further yields

$$\text{tol}_a \leq \frac{c}{\ell} \sqrt{\frac{c_\infty \varepsilon}{N - 1}} = O(N^{-3/2}).$$

Note the use of absolute threshold here, as opposed to a more common relative threshold in the previous subsection.
On the other hand, as may be concluded from the previous subsection, the one-level bound based on the repeated application of (3.10), (3.11) with $p = \ell - k$ may provide an accurate condition number estimate for the considered methods. In this case, it is, for instance, enough to require $\tilde{\gamma}^{2\ell} \leq c$, as follows from Corollary 3.6, that is,

$$
\gamma_k \leq \sqrt{\frac{c}{\ell}}
$$

must hold for some $c = \mathcal{O}(1)$. Combining this latter with (3.39), (2.7), and (4.8) entails a less restrictive condition

$$
(4.10) \quad tol_a \leq \sqrt{\frac{c}{\ell}} \sqrt{\frac{\epsilon_{\infty}}{N-1}} = \mathcal{O}(N^{-1})
$$
Now, the results for both strategies are given for \( c = 10 \) on Figure 4.2. Note that both strategies are effective in keeping the condition number bounded above. Although the strategy based on controlling \( \tilde{\gamma}^2 \ell \) is less restrictive, it is theoretically justified only for the one-level preconditioner.

5. Concluding remarks. We have presented a conditioning analysis of incomplete Cholesky factorizations based on orthogonal dropping. The analysis covers several existing preconditioners and provides an upper bound on the condition number which only depends on the accuracy \( \gamma_k \) of individual approximations. Whereas the analysis holds regardless of any assumption on the indices of rows modified during each approximation step, such assumptions may further improve the resulting estimate.

Now, the best improvement is obtained for the preconditioners based on the one-level modification pattern. The corresponding bound is further shown to be sharp for any possible set \( \gamma_k, k = 1, \ldots, \ell \), of accuracy measures. Moreover, numerical experiments reveal that one-level bound allows an accurate estimation of the condition number for various patterns, including one-level, SSS, and HSS. This latter observation is, however, restricted to a particular model problem and may not hold in full generality.

Regarding the accuracy measure \( \gamma_k \), one may estimate its value via (3.39) by assessing the norm \( \| R_{12}^{(k)} - \tilde{R}_{12}^{(k)} \| \) of the dropped component and, for instance, the norm \( \| A^{-1} \| \) of the inverse of the system matrix. The later parameter may be obtained with few iterations of the conjugate gradient method, whereas the former is directly controlled via the threshold value \( tol_a \) of a truncated orthogonal decomposition. Hence, the analysis offers a practical way to control the condition number of the resulting preconditioners. The potentialities of such an approach are highlighted in our numerical experiments with adaptive threshold strategies. We do not pursue this discussion here, however, since it is subject to further research.

Acknowledgment. I thank Xiaoye S. Li and especially Yvan Notay for their comments on the preliminary versions of this manuscript.

REFERENCES


