FACTORIZED APPROXIMATE INVERSES WITH ADAPTIVE DROPPING*

JIŘÍ KOPAL†, MIROSLAV ROZLOŽNÍK ‡, AND MIROSLAV TŮMA §

Abstract. This paper presents a new approach to construct factorized approximate inverses for a symmetric and positive definite matrix A. The proposed strategy is based on adaptive dropping that reflects the quality of preserving the relation $UZ = I$ between the direct factor U and the inverse factor Z satisfying $A = U^T U$ and $A^{-1} = ZZ^T$. An important part of the approach is column pivoting used to minimize the growth of the condition number of leading principal submatrices of U that occurs explicitly in the dropping criterion. Numerical experiments demonstrate that the resulting approximate inverse factorization is robust as a preconditioner for solving large and sparse systems of linear equations.

Key words. Approximate inverses, incomplete factorization, Gram–Schmidt orthogonalization, preconditioned iterative methods

1. Introduction. An important source of linear systems with positive definite matrices is represented by structured problems from discretizations of partial differential equations that arise in numerous applications in science and engineering and that often lead to problems with sparse matrices. But, there exist also an increasing number of applications that provide highly unstructured systems of linear equations. Let us denote the system of linear equations by

$$Ax = b, A \in \mathbb{R}^{n\times n},\ x \in \mathbb{R}^n, b \in \mathbb{R}^n, \tag{1.1}$$

where A is the system matrix, x is the vector of unknowns and b is the right-hand-side vector. Furthermore, let us assume that A is symmetric and positive definite. In order to solve such systems, direct methods have been often the methods of choice. A basic representative of these methods is the Cholesky factorization $A = U^T U$, where U is upper triangular. Iterative Krylov subspace methods are considered as an important alternative to direct methods for the solution of large systems of linear equations. In the symmetric and positive definite case, a natural choice of the iterative method is the conjugate gradient (CG) method. In order to increase the robustness of the CG method, the system (1.1) is usually transformed by preconditioning.

Although it is generally accepted that solving linear systems requires application-based preconditioners, the need for generally reliable incomplete factorizations is still strong. The most important method of this class is the incomplete Cholesky factorization, that is the factorization $A \approx \hat{U}^T \hat{U}$, where $\hat{U}$ is upper triangular. A crucial problem is that the incomplete Cholesky factorization can break down. Namely, a computed diagonal entry at some factorization step can be nonpositive. Widespread use of generally unstructured matrices has led to an increased interest in developing

*This work was supported by the project 13-06684S of the Grant Agency of the Czech Republic and by the ERC project MORE LL1202 financed by the MŠMT of the Czech Republic.

†Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic and Technical University of Liberec, Institute of Novel Technologies and Applied Informatics, Studentská 1402/2, 461 17 Liberec 1, Czech Republic (jiri@cs.cas.cz),

‡Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic, (miro@cs.cas.cz),

§Faculty of Mathematics and Physics, Charles University in Prague, Sokolovská 83, 186 75 Praha 8 and Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic, (mirektuma@karlin.mff.cuni.cz).

breakdown-avoiding techniques. On the other hand, an algorithmic modification to get a breakdown-free incomplete factorization leads to additional inaccuracies. Formally we can see the incomplete Cholesky factorization as computing a factorization of the perturbed matrix

$$A + \Delta\hat{E} = \hat{U}^T\hat{U}, \tag{1.2}$$

where the matrix $\hat{E}$ is called the factorization error. Theoretical analysis of incomplete factorizations that takes into account the actually decomposed matrix $A + \hat{E}$ has been considered only rarely. In addition, the bounds for the factorization error are often rough and typically need additional assumptions, see the paper by Kaporin [20]. An important step to make the dropping more global is the inverse-based dropping by Bollhöfer and Saad [7, 8, 9]. See also the multilevel approach in [10] showing the power of combining pivoting and the inverse-based dropping for solving nonsymmetric systems. Another attempt to get more reliable dropping in the incomplete factorization is to evaluate simultaneously with the direct incomplete factor $\hat{U}$ also the approximate inverse factor $\hat{Z}$, see [11], [12].

The inverse factorization is a counterpart of the Cholesky factorization and it computes $A^{-1} = ZZ^T$ with Z upper triangular. Its algorithm can be seen as the Gram–Schmidt orthogonalization of standard unit vectors with respect to a nonstandard inner product induced by the positive definite matrix A. In this way we get both two factors Z and U satisfying the identities

$$ZU = UZ = I. \tag{1.3}$$

Similarly to the Cholesky factorization of A, the inverse factorization can be computed approximatively. In such case we talk about the approximate inverse factorization $A^{-1} \approx \hat{Z}\hat{Z}^T$. We will consider the approximate factorizations $\hat{Z}\hat{U} \approx I$ and $\hat{U}\hat{Z} \approx I$ and introduce the left residual $\Delta\hat{F}$ and the right residual $\Delta\hat{G}$ defined by the relations $\hat{Z}\hat{U} = I+\Delta\hat{F}$ and $\hat{U}\hat{Z} = I+\Delta\hat{G}$, respectively. The history of the inverse factorizations goes back to papers by Morris [26], Purcell [27], Fox, Huskey and Wilkinson [18] as well as to the papers on the approximate inverse factorizations FSAI [21] and AINV [3], [2]. The combined use of the incomplete Cholesky factorization and the approximate inverse decompositions is also not new. In particular, it influenced dropping strategies for the incomplete factorizations that are rather sophisticated, see, e.g., [7], [25].

If we use the computed approximate factorized inverse as a preconditioner of some Krylov space method then the transformed system is

$$\hat{Z}^T A\hat{Z}y = \hat{Z}^T b, \qquad x = \hat{Z}y. \tag{1.4}$$

The quality of the approximation is then determined by the loss of orthogonality between the column vectors of $\hat{Z}$ defined as $\Delta\hat{H} = \hat{Z}^T A\hat{Z} - I$. This quantity is an analogue of the expression $\hat{U}^{-T}A\hat{U}^{-1} - I$ introduced by Chow and Saad [13] as a measure of stability. It is clear that a small right residual $\Delta\hat{G}$ together with a small factorization error $\Delta\hat{E}$ imply a small loss of orthogonality $\Delta\hat{H}$. Indeed, we have

$$\hat{Z}^T A\hat{Z} - I = \hat{Z}^T(\hat{U}^T\hat{U} - \Delta\hat{E})\hat{Z} - I = (I + \Delta\hat{G})^T(I + \Delta\hat{G}) - I - \hat{Z}^T\Delta\hat{E}\hat{Z}. \tag{1.5}$$

In this paper we introduce a new algorithmic strategy for dropping nonzero entries in the approximate inverse factorization that uses Gram–Schmidt process with

respect to a nonstandard inner product [3]. This work represents a continuation and refinement of the research published in [23]. Assumptions given there are supported by the theoretical results presented here. We also show that the new approximate inverse preconditioner efficiently solves large problems and it is rather robust when compared with the standard non-adaptive preconditioners. The paper is organized as follows. In Section 2 we recall the approximate inverse factorization based on Gram–Schmidt process with column pivoting. In Section 3 we analyze its behavior in finite precision arithmetic and give bounds for the right residual $\bar{U}\bar{Z} - I$ of the computed factors $\bar{Z}$ and $\bar{U}$. The new dropping strategy is introduced in Section 4. In Section 5 we present results of some numerical experiments. Concluding remarks are given in Section 6.

2. Gram–Schmidt process with column pivoting. In the following we consider the Gram–Schmidt process for orthogonalization of the standard unit vectors $e_1, \ldots, e_n$ with respect to the inner product $\langle \cdot, \cdot \rangle_A$ induced by the matrix A. We assume that the unit vectors are permuted so that they represent column vectors of the permutation matrix P. In this case, the Gram–Schmidt process applied to the columns of P leads to the factors Z and U satisfying

$$ZU = P, \tag{2.1}$$

where the columns of Z are A-orthonormal with $Z^T AZ = I$ and U is the upper triangular Cholesky factor of the matrix $P^T AP = U^T U$. It is clear that Z is the inverse factor satisying $A^{-1} = ZZ^T$. The Gram–Schmidt process is summarized in Algorithm 1, where $Z = [z_1,\ z_2,\ \ldots\ , z_n]$ are the resulting A-orthonormal vectors and $U = [\alpha_{j,k}]$ contains the orthogonalization coefficients. Here we consider the modified version of the Gram–Schmidt process [19] that is equivalent to the SAINV algorithm [2] as explained in [28].

Algorithm 1 Modified version of the Gram–Schmidt process with column permutation and with respect to the inner product $\langle \cdot, \cdot \rangle_A$

for $k := 1 \to n$ **do**
  $z_k^{(0)} := Pe_k$
  for $j := 1 \to k-1$ **do**
    $\alpha_{j,k} := \langle z_k^{(j-1)}, z_j \rangle_A$
    $z_k^{(j)} := z_k^{(j-1)} - \alpha_{j,k} z_j$
  end for
  $\alpha_{k,k} := \| z_k^{(k-1)} \|_A$
  $z_k := z_k^{(k-1)} / \alpha_{k,k}$
end for

Algorithm 1 computes for each k a column z_k of the factor Z using the vector Pe_k that is A-orthogonalized against previously computed vectors $z_1, \ldots, z_{k-1}$. This organization of the computation is known as a left-looking approach. The permutation matrix P is a priori unknown and has to be computed on-the-fly. Therefore, the left-looking Algorithm 1 requires additional precomputation of orthogonalization coefficients using the classical version of the Gram–Schmidt process [19]. Indeed, for each k and $j = k, \ldots, n$ we update the A-norms of the vectors $z_j^{(k-1)}$ as follows

$$\|z_j^{(k-1)}\|_A^2 = \|z_j^{(k-2)}\|_A^2 - \langle z_j^{(0)}, z_{k-1}\rangle_A^2. \tag{2.2}$$

The new k-th column vector $Pe_k \equiv e_i$ is chosen such that

$$\|z_i^{(k-1)}\|_A = \max_{k\le j\le n} \|z_j^{(k-1)}\|_A. \tag{2.3}$$

The permutation P is then obtained implicitly by the application of column pivoting with the criterion (2.3). It is clear that the orthogonalization coefficients $\alpha_{j,k}$ stored in the factor U satisfy the inequalities

$$\alpha_{1,1} \ge \alpha_{2,2} \ge \ldots \ge \alpha_{n,n} > 0, \tag{2.4}$$

$$\alpha_{k,k} > |\alpha_{k,j}|, \quad k = 1,\ldots,n, \quad j = k+1,\ldots,n. \tag{2.5}$$

In addition, (2.2) together with (2.4) and (2.5) also implies

$$\alpha_{k,k}^2 \ge \sum_{i=k}^{j} \alpha_{i,j}^2, \quad j = k+1,\ldots,n. \tag{2.6}$$

3. Gram–Schmidt process in finite precision arithmetic. Consider the computation of factors Z and U by Algorithm 1 in finite precision arithmetic. Due to rounding errors, we will distinguish between the exact quantities and the actually computed quantities. The quantities computed in finite precision arithmetic will be denoted by bars, e.g., $\bar{Z}, \bar{U}$. The bounds for the norms of $\Delta\bar{F} = \bar{Z}\bar{U} - I$, $\Delta\bar{H} = \bar{Z}^T A\bar{Z} - I$ and $\Delta\bar{E} = \bar{U}^T\bar{U} - A$ for the main versions of the Gram–Schmidt process with respect to the A-inner product have been given in [28]. They do not depend on any specific order of the entries in $\bar{U}$. In the following we will consider Algorithm 1 with column pivoting (2.3). For simplicity of our presentation, we assume that A has been already symmetrically reordered so that Algorithm 1 in finite precision arithmetic computes the factor $\bar{U} = [\bar{\alpha}_{j,k}]$ such that its entries $\bar{\alpha}_{j,k}$ satisfy inequalities analogous to (2.4), (2.5) and (2.6) for the exact arithmetic entries $\alpha_{j,k}$. The principal leading submatrices of $\bar{Z}$, $\bar{U}\ldots$ will be denoted by an additional subscript as $\bar{Z}_k$, $\bar{U}_k, \ldots$ for $k = 1,\ldots,n-1$. Entries of the matrices, e.g., for the matrix $\bar{U}_k$, we denote using $[\bar{U}_k]_{j,i} \equiv e_j^T\bar{U}_k e_i$. We will use standard notation to denote matrices with the absolute values of their entries, e.g., $|\bar{Z}_k|$, $|\bar{U}_k|, \ldots$ We assume the standard IEEE 754 model of floating-point computations [19]. The term $\mathcal{O}(\mathbf{u})$ represents a low degree polynomial in the problem dimension k multiplied by the unit roundoff $\mathbf{u}$. For simplicity, we do not evaluate the terms proportional to higher powers of $\mathbf{u}$ and skip also some technical details.

Let us focus on the right residual of computed factors defined as

$$\Delta\bar{G} = \bar{U}\bar{Z} - I. \tag{3.1}$$

Theorem 3.1 bounds the right residual (3.1) for the factors $\bar{Z}$ and $\bar{U}$ computed by Algorithm 1 in finite precision arithmetic.

THEOREM 3.1. *Let A be a symmetric and positive definite matrix. Let $\bar{Z}_k$ and $\bar{U}_k$ be computed by Algorithm 1 in finite precision arithmetic and denote by $\bar{D}_k$ the diagonal matrix defined as $\bar{D}_k \equiv \mathrm{diag}(\bar{U}_k)$. Then the entries of the right residuals $\Delta\bar{G}_k = \bar{U}_k\bar{Z}_k - I_k$ are bounded by*

$$|\Delta\bar{G}_k| \le \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_k||\bar{Z}_k||\bar{U}_k|\bar{D}_k^{-1}, \tag{3.2}$$

where γ_k represents the growth factor given recursively as

$$\gamma_{k-1}|\bar{D}_{k-1}^{-1}\bar{U}_{k-1}||\bar{D}_{k-1}^{-1}\bar{u}_k| \le \gamma_k|\bar{D}_{k-1}^{-1}\bar{u}_k|, \qquad \gamma_1 = 1. \tag{3.3}$$

Proof. The proof uses the bordering scheme with the partitioned factors

$$\bar{U}_k = \begin{pmatrix} \bar{U}_{k-1} & \bar{u}_k \\ 0 & \bar{\alpha}_{k,k} \end{pmatrix}, \quad \bar{Z}_k = \begin{pmatrix} \bar{Z}_{k-1} & \bar{w}_k \\ 0 & \bar{\beta}_{k,k} \end{pmatrix}. \tag{3.4}$$

The diagonal entries $\bar{\beta}_{k,k}$ in the factor $\bar{Z}_k$ satisfy

$$\bar{\beta}_{k,k} = \frac{1}{\bar{\alpha}_{k,k}} + \Delta\bar{\beta}_{k,k}, \qquad |\Delta\bar{\beta}_{k,k}| \le \frac{\mathbf{u}}{\bar{\alpha}_{k,k}}$$

and for each $k = 1, \ldots, n$ we then get

$$|\bar{\beta}_{k,k}\bar{\alpha}_{k,k} - 1| = \left|\frac{\bar{\alpha}_{k,k}(1 + \Delta\beta_{k,k})}{\bar{\alpha}_{k,k}} - 1\right| \le \mathbf{u}. \tag{3.5}$$

The last column vector in $\bar{Z}_k$ computed using the already computed column vectors of $\bar{Z}_{k-1}$ and the coefficients in the last column vector of $\bar{U}_k$ satisfy

$$\bar{w}_k = -\bar{Z}_{k-1}\bar{u}_k\bar{\beta}_{k,k} + \Delta w_k, \qquad |\Delta w_k| \le \mathcal{O}(\mathbf{u})|\bar{Z}_{k-1}||\bar{u}_k|\bar{\beta}_{k,k}. \tag{3.6}$$

The proof is by induction on k. Assume that (3.2) is true for matrices of the order $k-1$. Considering (3.4) for $k = 2, \ldots, n$ we get

$$\bar{U}_k\bar{Z}_k - I_k = \begin{pmatrix} \bar{U}_{k-1}\bar{Z}_{k-1} - I_{k-1} & \bar{U}_{k-1}\bar{w}_k + \bar{u}_k\bar{\beta}_{k,k} \\ 0 & \bar{\alpha}_{k,k}\bar{\beta}_{k,k} - 1 \end{pmatrix}. \tag{3.7}$$

Taking into account (2.5), we have

$$[|\bar{D}_{k-1}^{-1}\bar{U}_{k-1}|]_{j,i} \le 1, \quad j = 1, \ldots, k-1, \quad i = 1, \ldots, k-1 \tag{3.8}$$

and

$$[|\bar{D}_{k-1}^{-1}\bar{u}_k|]_j < 1, \quad j = 1, \ldots, k-1, \tag{3.9}$$

then after some manipulations with the off-diagonal block of (3.7), using (3.6) as well as the definition of the growth factor γ_k given in (3.3) we obtain

$$\begin{aligned}
\bar{U}_{k-1}\bar{w}_k + \bar{u}_k\bar{\beta}_{k,k} &= \bar{U}_{k-1}(-\bar{Z}_{k-1}\bar{u}_k\bar{\beta}_{k,k} + \Delta w_k) + \bar{u}_k\bar{\beta}_{k,k} \\
&= (I_{k-1} - \bar{U}_{k-1}\bar{Z}_{k-1})\bar{u}_k\bar{\beta}_{k,k} + \bar{U}_{k-1}\Delta w_k \\
&\le |I_{k-1} - \bar{U}_{k-1}\bar{Z}_{k-1}||\bar{u}_k|\bar{\beta}_{k,k} + \mathcal{O}(\mathbf{u})|\bar{U}_{k-1}||\bar{Z}_{k-1}||\bar{u}_k|\bar{\beta}_{k,k} \\
&\le \mathcal{O}(\mathbf{u})\gamma_{k-1}\left(|\bar{U}_{k-1}||\bar{Z}_{k-1}\bar{D}_k||\bar{D}_k^{-1}\bar{U}_{k-1}|\bar{D}_{k-1}^{-1}|\bar{u}_k| + |\bar{U}_{k-1}||\bar{Z}_{k-1}||\bar{u}_k|\right)\bar{\beta}_{k,k} \\
&\le \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_{k-1}|(|\bar{Z}_{k-1}\bar{D}_{k-1}||\bar{D}_{k-1}^{-1}\bar{u}_k| + |\bar{Z}_{k-1}||\bar{u}_k|)\bar{\beta}_{k,k} \\
&= \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_{k-1}|(|\bar{Z}_{k-1}||\bar{u}_k| + |\bar{Z}_{k-1}||\bar{u}_k|)\bar{\beta}_{k,k} \\
&= \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_{k-1}||\bar{Z}_{k-1}||\bar{u}_k|\bar{\beta}_{k,k}.
\end{aligned} \tag{3.10}$$

From the inequalities (3.10) and (3.5) we get then the statement of our Theorem. ∎

A slightly different bound on the right residual (3.2) is given in the following corollary.

COROLLARY 3.1. *The right residual of the computed factors $\bar{Z}$ and $\bar{U}$ in Algorithm 1 with the column pivoting (2.3) satisfy*

$$\|\Delta\bar{G}_k\| \leq \mathcal{O}(\mathbf{u})\gamma_k\|\bar{U}_k\|\|\bar{Z}_k\bar{D}_k\|\|\bar{D}_k^{-1}\| \leq \frac{\mathcal{O}(\mathbf{u})\gamma_k\kappa(\bar{U}_k)\kappa(\bar{D}_k)}{1-\mathcal{O}(\mathbf{u})\gamma_k\kappa(\bar{U}_k)\kappa(\bar{D}_k)}. \tag{3.11}$$

Proof. The bound (3.2) can be rewritten as follows

$$|\Delta\bar{G}_k| \leq \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_k||\bar{Z}_k\bar{D}_k||\bar{D}_k^{-1}\bar{U}_k|\bar{D}_k^{-1}. \tag{3.12}$$

Taking into account (3.8) and using appropriate bounds for the norms of the matrices $|\bar{U}_k|, |\bar{Z}_k\bar{D}_k|$ and $|\bar{D}_k^{-1}\bar{U}_k|$ we get (3.11). ∎

The bound (3.11) can be significantly improved for M-matrices even without pivoting as we show in Theorem 3.2 below. Note that a nonsingular real square matrix A with non-positive off-diagonal entries is called an M-matrix if all the entries of its inverse are non-negative, see, e.g., [6], [17].

THEOREM 3.2. *Let A be a symmetric and positive definite M-matrix. Let $\bar{Z}_k$ and $\bar{U}_k$ be computed by Algorithm 1 in finite precision arithmetic, where we set the coefficients $\bar{\alpha}_{j,k}$ equal to zero whenever its computed counterpart eventually becomes positive. Then the right residuals $\Delta\bar{G}_k$ are bounded by*

$$|\Delta\bar{G}_k| \leq \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_k||\bar{Z}_k|. \tag{3.13}$$

Proof. Assume A_k is an M-matrix. Then its Cholesky factorization has the form $A_k = U_k^T U_k$ such that $\mathrm{striu}(U_k) \leq 0$ and $D_k \equiv \mathrm{diag}(U_k) > 0$ [17]. It is easy to see that $z_1 \geq 0$. Due to $Z_k D_k = I_k - Z_k\,\mathrm{striu}(U_k) = I_k + Z_k\,|\mathrm{striu}(U_k)|$ we have $z_k^{(j)} \geq z_k^{(j-i)} \geq 0, j = 1,\ldots,k-1, k = 2,\ldots,n$ and thus $Z = |Z|$. In [28] it has been shown that upper triangular factor $\bar{U}_k$ computed in finite precision arithmetic by the modified Gram–Schmidt orthogonalization process with respect inner product induced by the matrix A_k satisfies the relation $A_k + \Delta\bar{E}_k = \bar{U}_k^T\bar{U}_k$ where $\|\Delta\bar{E}_k\| \leq \mathcal{O}(\mathbf{u})\kappa(A_k)\|A_k\|$. Since the off-diagonal entries of $\bar{U}_k$ are non-positive then $A_k + \Delta\bar{E}_k$ is an M-matrix. In addition, if $\mathcal{O}(\mathbf{u})\kappa^2(A) < 1$ then $A_k + \Delta\bar{E}_k$ is symmetric positive definite matrix and the modified Gram–Schmidt process runs to completion without breakdown. This implies $\bar{Z}_k = |\bar{Z}_k|$ and $|\bar{U}_k| = -\bar{U}_k + 2\bar{D}_k$. For the term $|\bar{Z}_k||\bar{U}_k|\bar{D}_k^{-1}$ it follows

$$|\bar{Z}_k||\bar{U}_k|\bar{D}_k^{-1} = \bar{Z}_k(-\bar{U}_k + 2\bar{D}_k)\bar{D}_k^{-1} = -\Delta\bar{F}_k\bar{D}_k^{-1} + 2\bar{Z}_k - \bar{D}_k^{-1}. \tag{3.14}$$

It has been shown in [28] that the matrices $\bar{Z}_k$ and $\bar{U}_k$ computed by the modified Gram–Schmidt process with respect to inner product induced by the matrix A satisfy the recurrence $\bar{Z}_k\bar{U}_k = I_k + \Delta\bar{F}_k$ with $|\Delta\bar{F}_k| \leq \mathcal{O}(\mathbf{u})|\bar{Z}_k||\bar{U}_k|$. Therefore

$$|\bar{Z}_k||\bar{U}_k|\bar{D}_k^{-1} \leq \mathcal{O}(\mathbf{u})|\bar{Z}_k||\bar{U}_k|\bar{D}_k^{-1} + |2\bar{Z}_k - \bar{D}_k^{-1}|. \tag{3.15}$$

leading to the bound $|\bar{Z}_k||\bar{U}_k|\bar{D}_k^{-1} \leq (2 + \mathcal{O}(\mathbf{u}))|\bar{Z}_k|$. Using it in (3.2) we get

$$|\Delta\bar{G}_k| \leq \mathcal{O}(\mathbf{u})\gamma_k|\bar{U}_k||\bar{Z}_k|. \tag{3.16}$$

∎

In the following we will introduce a dropping strategy that is based on the monitoring of the right residuals and keeping their sizes on the level given by bounds similar to (3.11) and (3.13).

4. Approximate inverse factorization with adaptive dropping. In this Section we propose a specific Gram–Schmidt process with dropping that attempts to compute approximate factors with uniformly bounded right residuals. The approximate quantities will be denoted by tildes, e.g., $\tilde{Z}, \tilde{U}, \ldots$, and $\tilde{Z}_k, \tilde{U}_k, \ldots$ for their principal submatrices of dimension $k = 1, \ldots, n$. The crucial idea of our approach is to drop the entries in the factor $\tilde{Z}$ on a level related to the size of the right residual

$$\Delta\tilde{G} = \tilde{U}\tilde{Z} - I. \tag{4.1}$$

In particular, we will require that the norm of each column of the right residual (4.1) is uniformly bounded by a drop tolerance $\boldsymbol{\tau}$. This bound can be written as

$$\|\Delta\tilde{G}_k e_k\| \leq \boldsymbol{\tau}, \tag{4.2}$$

where $\Delta\tilde{G}_k = \tilde{U}_k\tilde{Z}_k - I_k$. Note that due to the interlacing property of the singular values [29] the norms of the right residuals $\Delta\tilde{G}_k$ satisfy

$$\|\Delta\tilde{G}_k\| \leq \|\Delta\tilde{G}_{k+1}\|, \; k = 1, \ldots, n-1. \tag{4.3}$$

Clearly, if we want to satisfy the bound (4.2) for all columns, the dropping must be adaptive.

The critical step in Algorithm 1 is the computation of potentially dense vectors $\tilde{z}_k^{(k-1)}$ for $k = 1, \ldots, n$. Given the initial vector $\tilde{z}_k^{(0)} = \tilde{P}e_k$, let us consider $\tilde{z}_k^{(k-1)}$ computed by the recurrences

$$\tilde{z}_k^{(j)} = \tilde{z}_k^{(j-1)} - \tilde{\alpha}_{j,k}\tilde{z}_j, \quad \tilde{\alpha}_{j,k} = \langle \tilde{z}_k^{(j-1)}, \tilde{z}_j \rangle_A, \quad j = 1, \ldots, k-1, \tag{4.4}$$

where $\tilde{\tilde{z}}_k = \tilde{z}_k^{(k-1)}/\|\tilde{z}_k^{(k-1)}\|_A$ is generally dense. In order to keep the factor $\tilde{Z}$ sparse, some of its entries should be dropped. Let $s_k \in \{0,1\}^n$ be a vector that specifies the chosen dropping and assume that some entries of $\tilde{z}_k^{(k-1)}$ have been dropped. The resulting sparse vector is given as $\tilde{z}_k^{(k-1)} \circ s_k$ and it is normalized to get the new column of $\tilde{Z}$ in the form

$$\tilde{z}_k \equiv \frac{\tilde{z}_k^{(k-1)} \circ s_k}{\|\tilde{z}_k^{(k-1)} \circ s_k\|_A}. \tag{4.5}$$

Here the operator "$\circ$" denotes the Hadamard (entry-wise) product. Let us define the correction vector $\Delta\tilde{z}_k$ as $\Delta\tilde{z}_k = \tilde{z}_k - \bar{\tilde{z}}_k$, where $\bar{\tilde{z}}_k$ is the vector computed by the recurrence (4.4) in finite precision arithmetic. We can then write

$$\Delta\tilde{G}_k e_k = \Delta\bar{\tilde{G}}_k e_k + \tilde{U}_k \Delta\tilde{z}_k, \tag{4.6}$$

where the right residual $\Delta\bar{\tilde{G}}_k e_k = \tilde{U}_k\bar{\tilde{z}}_k - e_k$ satisfies the bound (3.12) in the general case and the bound (3.16) if A is an M-matrix. Let us consider dropping at the step k such that

$$\frac{\|\Delta\tilde{z}_k\|}{\|\bar{\tilde{z}}_k\|_\infty} \leq \boldsymbol{\tau_k} \tag{4.7}$$

for some step-dependent parameter $\boldsymbol{\tau_k}$ satisfying the inequalities

$$\boldsymbol{\tau_k}\, \|\tilde{U}_k\|\|\bar{\tilde{z}}_k\|_\infty \leq \boldsymbol{\tau_k}\, \|\tilde{U}_k\|\|\bar{\tilde{z}}_k\| \lessapprox \boldsymbol{\tau_k}\, \|\tilde{U}_k\|\|\tilde{U}_k^{-1}\|\|e_k\| < \boldsymbol{\tau}.$$

Then the right residual $\Delta\tilde{G}_k e_k$ that appears in (4.6) can be up to the terms proportional to the product of the unit roundoff and the growth factor bounded by the parameter $\boldsymbol{\tau}$ and thus it satisfies the required bound (4.2). Consequently, $\boldsymbol{\tau_k}$ satisfies the following inequality

$$\boldsymbol{\tau_k} \leq \frac{\boldsymbol{\tau}}{\kappa(\tilde{U}_k)}. \tag{4.8}$$

This new dropping technique is thus based on monitoring the condition number of $\tilde{U}_k$. The interlacing property for singular values [29] implies that $\kappa(\tilde{U}_1) \leq \cdots \leq \kappa(\tilde{U}_n)$ and due to (4.8) the sequence of drop tolerances $\boldsymbol{\tau_k}$ is non-increasing as $\boldsymbol{\tau_k}$ decreases when $\kappa(\tilde{U}_k)$ increases. Since the proposed dropping strategy depends on the condition numbers $\kappa(\tilde{U}_k)$, a natural strategy is to keep the increase in the sequence of the condition numbers $\kappa(\tilde{U}_k)$ as low as possible and this can be achieved by the pivoting. The resulting procedure is summarized in Algorithm 2.

Algorithm 2 Modified version of the Gram–Schmidt process with column permutation and with adaptive dropping

for $k := 1 \to n$ **do**
  $\tilde{z}_k^{(0)} := \tilde{P} e_k$
  for $j := 1 \to k-1$ **do**
    $\tilde{\alpha}_{j,k} := \langle \tilde{z}_k^{(j-1)}, \tilde{z}_j \rangle_A$
    $\tilde{z}_k^{(j)} := \tilde{z}_k^{(j-1)} - \tilde{\alpha}_{j,k} \tilde{z}_j$
  end for
  $\tilde{\tilde{\alpha}}_{k,k} := \|\tilde{z}_k^{(k-1)}\|_A$
  $\tilde{\tilde{z}}_k := \tilde{z}_k^{(k-1)} / \tilde{\tilde{\alpha}}_{k,k}$
  for $i := 1 \to n$ **do**
    if $|e_i^T \tilde{\tilde{z}}_k| > \boldsymbol{\tau}\ \|\tilde{\tilde{z}}_k\|_\infty / \kappa(\tilde{U}_k)$ **then**
      $e_i^T s_k := 1$
    else
      $e_i^T s_k := 0$
    end if
  end for
  $(\tilde{P} e_k)^T s_k := 1$
  $\tilde{\alpha}_{k,k} := \|\tilde{\tilde{z}}_k \circ s_k\|_A$
  $\tilde{z}_k := (\tilde{\tilde{z}}_k \circ s_k) / \tilde{\alpha}_{k,k}$
end for

5. Numerical experiments. In this section we present results of our numerical experiments. We will show that the new dropping strategy can lead to reliable approximate inverse factorization. All codes were written in FORTRAN 95, and have been compiled with Intel Fortran Composer XE 2013. We have used one processor of Intel Core2 Q6700 (2.66GHz, 16GB RAM). The figures have been prepared in Matlab^{TM}. Table 5.1 summarizes test matrices taken from the Tim Davis collection of sparse matrices [15] including the test one used for detailed demonstrations. The table presents their dimensions and nonzero counts (sizes) together with their short descriptions. Note that a significant proportion of matrices comes from structural mechanics, where approximate inverse preconditioners represent a method of choice, see [2].

TABLE 5.1
Test problems

Matrix	n	nnz	Application
BCSSTK19	817	6,853	stiffness matrix - part of a suspension bridge
FV1	9,604	47,434	FEM - 2D Laplace equation
FV3	9,801	48,413	FEM - 2D Laplace equation
MSC10848	10,848	1,229,776	MSC Nastran matrix
BCSSTK25	15,439	133,840	stiffness matrix - skyscraper
OLAFU	16,146	515,651	NASA matrix - accuracy problem on Y-MP
BODYY4	17,546	69,742	NASA matrix collected by Alex Pothen
BODYY5	18,589	73,935	NASA matrix collected by Alex Pothen
RAEFSKY4	19,779	674,195	buckling problem for container
MSC23052	23,052	1,142,686	MSC Nastran matrix
BCSSTK36	23,052	1,143,140	stiffness matrix - car shock absorber
VANBODY	47,072	1,191,985	stiffness matrix - GHS collection
NASASRB	54,870	2,677,324	NASA matrix - shuttle rocket booster
OILPAN	73,752	2,148,558	INPRO test matrix
S3DKQ4M2	90,449	2,455,670	FEM on cylindrical shells
AUTOSHELL8	504,855	17,579,155	sheet metal forming matrix
LDOOR	952,203	42,493,817	INDEED test matrix - GHS collection

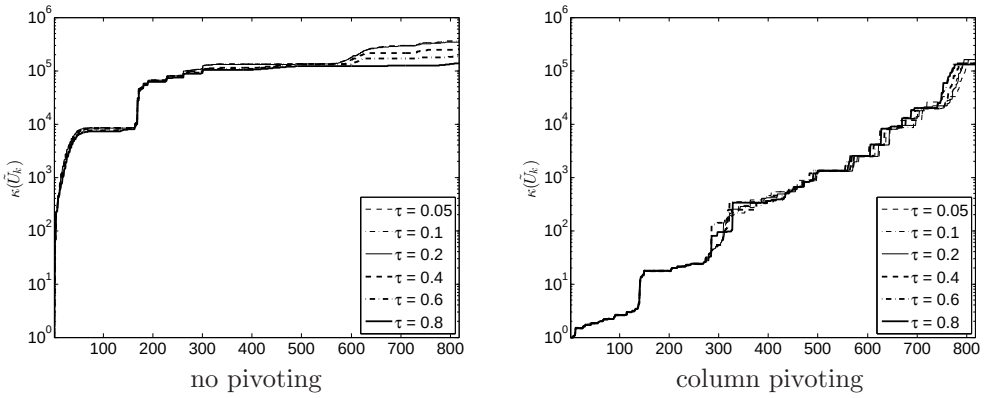

FIG. 5.1. *Conditioning of the leading principal submatrices $\tilde{U}_k$ of the matrix BCSSTK19 without pivoting (on the left) and with pivoting (on the right).*

The importance of column pivoting is indicated in Figure 5.1 for the matrix BCSSTK19. The plot on the left shows dependence of the condition number $\kappa(\tilde{U}_k)$ on the major loop index in Algorithm 2 with suppressed pivoting $\tilde{P} = I$. The plot on the right corresponds to Algorithm 2 with adaptive dropping and column pivoting. It is clear that the growth of $\kappa(U_k)$ is much more moderate with pivoting than without pivoting. This implies that one can safely drop more entries also in the later steps of the factorization. Figure 5.2 showing the sparsity pattern of the factor $\tilde{Z}$ confirms the fact that we can drop significantly more nonzeros with pivoting. As above, the plot on the left corresponds to the factorization with suppressed pivoting and the plot on the right corresponds to Algorithm 2 with column pivoting and adaptive dropping. Similarly to [24] the actual implementation of pivoting is based on the concept of heaps [14] and it is very fast. In the subsequent experiments with larger matrices the condition number $\kappa(\tilde{U}_k)$ in the dropping criterion (4.8) was cheaply approximated

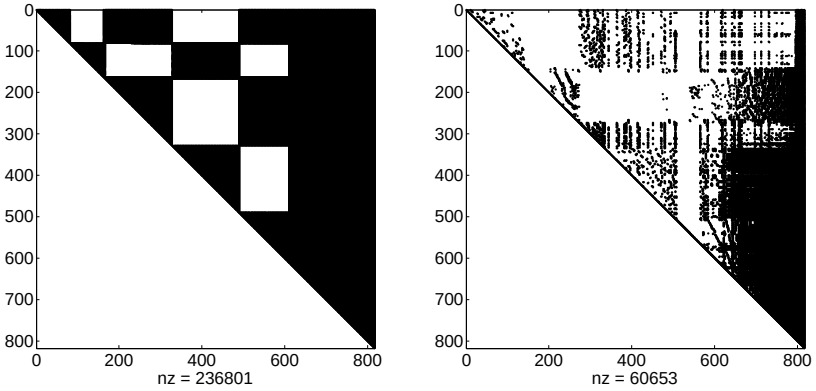

FIG. 5.2. *Sparsity structure of the factor $\tilde{Z}$ of the matrix BCSSTK19 without pivoting (on the left) and with pivoting (on the right) with $\tau = 0.2$.*

by the ratio of the largest and the smallest diagonal entry of $\tilde{U}_k$. Let us note here that there exists a different and important way to relax the dropping. It replaces $\kappa(\tilde{U}_k)$ by an approximate norm $\tilde{U}_k^{-1}$. Namely, one could argue that the norm of $\tilde{U}_k$ stays moderate during the decomposition and the inverse $\tilde{U}_k^{-1}$ actually drives the factorization, for details see [7, 8, 9]. Let us also note that an example in [23] (based on more experiments) shows that the simple estimation of $\kappa(\tilde{U}_k)$ based on diagonal entries is sufficient when compared with the expensive condition number computation, but it does not need to be the case. This is the reason that we would like to explore more sophisticated strategies for pivoting in the future.

Table 5.2 compares results for Algorithm 2 that uses column pivoting and adaptive dropping with the tolerance τ (we refer to it as adaptive SAINV) and the standard SAINV algorithm with absolute dropping with a drop tolerance τ and no pivoting. (this algorithm is denoted as standard SAINV). For relations of standard SAINV to other algebraic preconditioning strategies see, for example, [4]. For all test matrices we have computed two sparse instances of the factor $\tilde{Z}$. The dropping tolerances in corresponding standard SAINV and adaptive SAINV were experimentally chosen so that the factors are comparable in size and thus allow a fair comparison. These two factorizations were used for preconditioning of the conjugate gradient method (CG). The stopping criterion used for CG is based on the backward error. The iterations were terminated as soon as the backward error reached 10^{-6}. The initial guess was set to zero vector and the right-hand side vector was chosen so that the solution is the vector of all 1's.

Let discuss the results in Table 5.2. Despite a fast implementation of pivoting based on heaps, adaptive SAINV factorization is typically slower than standard SAINV factorization. It is not always the case; pivoting induces a different reordering and this can strongly influence the factorization [16], [5] or [1]. On the other hand, Table 5.2 shows that the adaptive approach is more reliable. While adaptive SAINV was able to solve all problems, it was not the case for standard SAINV. In order to judge on the method robustness, an important source of information is the behavior of the preconditioned iterative method with respect to different sizes of the factor $\tilde{Z}$. Figures 5.3 and 5.4 describe dependence of the iteration counts on the preconditioner size for a chosen set of four matrices. As one could expect, the figures confirm that the

TABLE 5.2
Size (nnz), iteration count (its), factorization time in seconds (t_f) and time for the preconditioned iterations (t_{it}) for standard SAINV and adaptive SAINV.

Matrix	Standard SAINV					Adaptive SAINV				
	nnz	τ	*its*	t_f	t_{it}	*nnz*	τ	*its*	t_f	t_{it}
FV1	19,110	0.08	17	0.08	0.02	18,174	0.03	18	0.12	0.02
FV3	30,848	0.06	115	0.09	0.08	33,447	0.004	104	0.11	0.08
MSC10848	45,162	0.003	76	1.22	0.33	45,030	0.09	92	1.08	0.39
BCSSTK25	86,004	0.004	15	0.34	0.03	79,927	0.006	8	0.33	0.03
OLAFU	18,551	0.4	18	0.56	0.08	54,291	0.4	18	0.89	0.08
BODYY4	47,196	0.004	36	0.17	0.06	48,375	0.009	42	0.21	0.06
BODYY5	73,552	0.002	36	0.19	0.06	74,356	0.0002	58	0.24	0.09
RAEFSKY4	124,654	0.06	7	1.19	0.05	130,072	0.06	7	1.41	0.05
MSC23052	255,457	0.002	‡	3.36	‡	149,101	0.002	289	1.59	1.56
BCSSTK36	115,452	0.02	795	1.43	4.09	123,459	0.06	286	1.06	1.41
VANBODY	231,839	0.006	5	1.97	0.06	418,134	0.006	8	3.16	0.09
NASASRB	87,594	0.1	611	1.36	5.59	190,171	0.1	489	1.97	5.06
OILPAN	106,154	0.02	406	1.88	5.14	116,780	0.4	403	2.15	5.14
S3DKQ4M2	195,223	0.02	302	3.24	5.78	313,447	0.02	285	3.61	5.24
AF_SHELL8	1,085,444	0.02	194	9.91	12.5	1,059,786	0.02	164	15.1	12.2
LDOOR	2,484,854	0.02	454	28.4	77.4	2,010,227	0.4	456	39.6	89.2

new approach is less prone to instabilities. We can also see than from Table 5.2 that although adaptive SAINV is often better than standard SAINV, it is not always the case. This is demonstrated for the matrix OILPAN where the standard SAINV seems to be slightly better. Also let us note that in the case of the matrix AF_SHELL8 we can see that the preconditioner is not very efficient since the number of its iterations generally increases with the size. The computation of a very dense preconditioner would be extremely expensive and out of scope of our computational resources.

It follows from Figures 5.1-5.4 that standard SAINV often leads to denser factors than adaptive SAINV. This may be an effect of instabilities that sometimes lead to large growth of entries in the factorization. Note that this is in agreement with a previous observation that SAINV often generates much smaller factors than other less stable variants [2]. In many cases, we were not able to generate a very large fill-in in the adaptive SAINV within the same interval of given τ that we call here the drop tolerance. A related question is how robust is the choice of the drop tolerance so that we could assume that it may lead to computation of a useful preconditioner, see, for example a recent study [22]. Note that in practice of algebraic preconditioning we are often interested in finding a reliable value of drop tolerance, and the adaptiveness was introduced here partially with this intention. Let us demonstrate this effect for the matrix AF_SHELL8 from Figure 5.4. Changing the drop tolerance from 0.1 to 1.0d-4 the preconditioner size for the adaptive SAINV changes from 788,062 to 2,788,921 that could be considered as a modest change. At the same time, standard SAINV for the drop tolerance 0.1 provides preconditioner of the size 511,108 and for 0.6d-3 the preconditioner has the size 2,820,445. This robustness with respect to the choice of the drop tolerance can be possibly explained by a smoother growth of the condition number demonstrated for a small problem on Figure 5.1 and by growth of entries if pivoting is not used.

Let us look at the relation between the column pivoting on one side and the adap-

tive dropping on the other side. It is true that pivoting does the job of avoiding small pivots and therefore it limits the growth of entries in Z. The role of the adaptive dropping is more subtle, and it contributes to the robustness with respect to the drop tolerance. If we consider τ as an user input, a practical goal is to find a preconditioner that works even when τ is not the best one. For the same drop tolerance τ, the adaptive dropping typically provides a larger factor than the non-adaptive dropping and it may represent a safer preconditioner. We show this on the example in Table 5.3 with a simple Laplacian from a two-dimensional grid 60×60 discretized by five point finite differences. Here we present the iteration counts and preconditioner sizes for both adaptive and non-adaptive SAINV with the column pivoting. We can see that in this simple but important case the adaptiveness implies slightly denser factors than the standard approach for the same drop tolerance since it moves the preconditioner to satisfy the uniform bound (4.2) by allowing more fill-in. But, as shown in figures, standard SAINV may suffer from instabilities and then the adaptiveness may contribute to limit the growth of entries in the preconditioner although the influence of pivoting of the adaptive SAINV seems to be profound. Studying other possible effects of such uniform bounds as (4.2) also for different preconditioners and interplay between pivoting and adaptiveness are a part of our future research plans. Note that another experiment with adaptive dropping without the column pivoting is presented in [23].

TABLE 5.3

A comparison of adaptive SAINV and non-adaptive SAINV (both algorithms use column pivoting) for the 2D Laplacian matrix. Drop tolerance is denoted by τ, its denotes the number of iterations of PCG and size denotes the preconditioner size.

τ	*its*		*size*	
	adaptive SAINV	non-adaptive SAINV	adaptive SAINV	non-adaptive SAINV
0.250	79	87	11,589	10,680
0.225	69	87	12,880	10,715
0.203	54	84	15,754	11,208
0.164	47	57	18,176	15,441
0.133	41	47	21,603	17,698
0.108	38	43	24,417	20,765
0.087	32	40	30,565	23,269
0.071	29	34	36,178	29,266

Summarizing our findings, adaptive SAINV represents a significant enhancement of the standard SAINV scheme since it clearly increases the predictability of the behavior of the preconditioned conjugate gradient method. This approach can be combined with standard preprocessing techniques.

6. Conclusions. In this paper we considered Gram–Schmidt process with column pivoting. We have analyzed its behavior in the finite precision arithmetic focusing on the bound for the right residual of the computed factors. Based on that we have introduced a new adaptive dropping that is monitoring the right residuals and trying to keep their size uniformly bounded by a given drop tolerance. Numerical results indicate that this approach leads to a robust preconditioner for the conjugate gradient method.

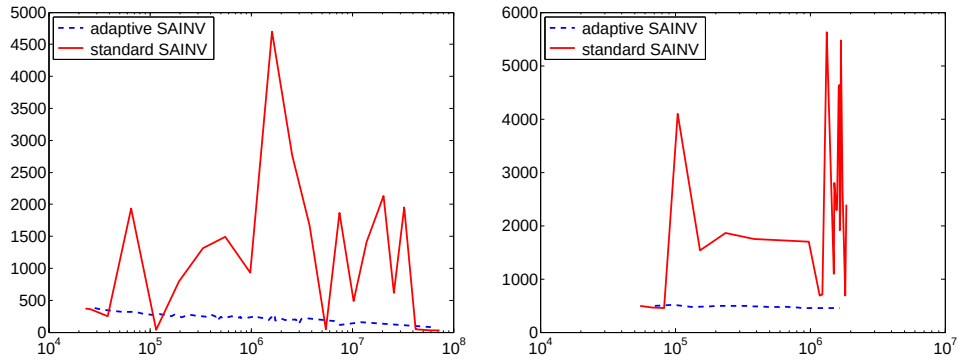

FIG. 5.3. *Preconditioner size versus iteration counts for the CG method for the matrices BCSSTK36 (left) and NASASRB (right)*

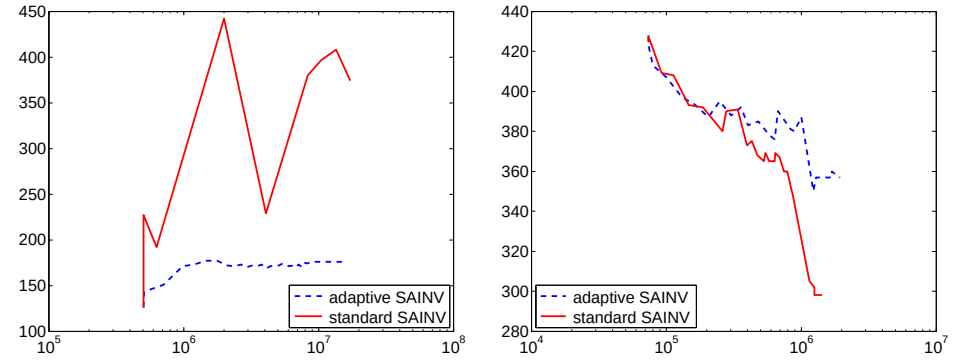

FIG. 5.4. *Preconditioner size versus iteration counts for the CG method for the matrices AF_SHELL8 (left) and OILPAN (right)*

7. Acknowledgements. We would like to thank the anonymous reviewers for carefully reading our manuscript and making significant constructive suggestions for its improvement.

REFERENCES

[1] M. Benzi. Preconditioning techniques for large linear systems: a survey. *J. of Computational Physics*, 182(2):418–477, 2002.

[2] M. Benzi, J. K. Cullum, and M. Tůma. Robust approximate inverse preconditioning for the conjugate gradient method. *SIAM J. on Scientific Computing*, 22(4):1318–1332, 2000.

[3] M. Benzi, C. D. Meyer, and M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. on Scientific Computing*, 17(5):1135–1149, 1996.

[4] M. Benzi and M. Tůma. A comparative study of sparse approximate inverse preconditioners. *Applied Numerical Mathematics*, 30(2-3):305–340, 1999.

[5] M. Benzi and M. Tůma. Orderings for factorized sparse approximate inverse preconditioners. *SIAM J. on Scientific Computing*, 21(5):1851–1868, 2000.

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

[7] M. Bollhöfer. A robust ILU with pivoting based on monitoring the growth of the inverse factors. *Linear Algebra and its Applications*, 338:201–218, 2001.

[8] M. Bollhöfer. A robust and efficient ILU that incorporates the growth of the inverse triangular factors. *SIAM J. on Scientific Computing*, 25(1):86–103, 2003.

[9] M. Bollhöfer and Y. Saad. On the relations between ILUs and factored approximate inverses. *SIAM J. on Matrix Analysis and Applications*, 24(1):219–237, 2002.

[10] M. Bollhöfer and Y. Saad. Multilevel preconditioners constructed from inverse-based ILUs. *SIAM J. on Scientific Computing*, 27(5):1627–1650, 2006.

[11] R. Bru, J. Marín, J. Mas, and M. Tůma. Balanced incomplete factorization. *SIAM J. on Scientific Computing*, 30(5):2302–2318, 2008.

[12] R. Bru, J. Marín, J. Mas, and M. Tůma. Improved balanced incomplete factorization. *SIAM J. on Matrix Analysis and Applications*, 31(5):2431–2452, 2010.

[13] E. Chow and Y. Saad. Experimental study of ILU preconditioners for indefinite matrices. *J. of Computational and Applied Mathematics*, 86(2):387–414, 1997.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009.

[15] T. A. Davis. *University of Florida Sparse Matrix Collection*. available online at http://www.cise.ufl.edu/research/sparse/matrices/, 1994.

[16] I. S. Duff and G. A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT Numerical Mathematics*, 29:635–657, 1989.

[17] M. Fiedler and V. Pták. On matrices with non-positive off-diagonal elements and positive principal minors. *Czechoslovak Mathematical Journal*, 12 (87):382–400, 1962.

[18] L. Fox, H. D. Huskey, and J. H. Wilkinson. Notes on the solution of algebraic linear simultaneous equations. *Quart. J. Mech. and Appl. Math.*, 1:149–173, 1948.

[19] N. J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2002.

[20] I. E. Kaporin. High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$ decomposition. *Numerical Linear Algebra with Applications*, 5:483–509, 1998.

[21] L. Y. Kolotilina and A. Y. Yeremin. Factorized sparse approximate inverse preconditionings. I. Theory. *SIAM J. on Matrix Analysis and Applications*, 14(1):45–58, 1993.

[22] I. N. Konshin, M. A. Olshanskii, and Y. V. Vassilevski. ILU preconditioners for non-symmetric saddle-point matrices with application to the incompressible Navier-Stokes equations. Preprint, 2015.

[23] J. Kopal, M. Rozložník, and M. Tůma. Approximate inverse preconditioners with adaptive dropping. *Advances in Engineering Software*, 84:13–20, 2015.

[24] N. Li and Y. Saad. Crout versions of ILU factorization with pivoting for sparse symmetric matrices. *Electronic Transactions on Numerical Analysis*, 20:75–85, 2005.

[25] S. MacLachlan, D. Osei-Kuffuor, and Y. Saad. Modification and compensation strategies for threshold-based incomplete factorizations. *SIAM J. on Scientific Computing*, 34(1):A48–A75, 2012.

[26] J. Morris. An escalator process for the solution of linear simultaneous equations. *Philos. Mag.*, 37:106–120, 1946.

[27] E. W. Purcell. The vector method of solving simultaneous linear equations. *J. Math. Phys.*, 32:150–153, 1953.

[28] M. Rozložník, M. Tůma, A. Smoktunowicz, and J. Kopal. Numerical stability of orthogonalization methods with a non-standard inner product. *BIT*, 52(4):1035–1058, 2012.

[29] R. C. Thompson. Principal submatrices. IX. Interlacing inequalities for singular values of submatrices. *Linear Algebra and Appl.*, 5:1–12, 1972.