Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com

An adaptive multilevel factorized sparse approximate inverse preconditioning[☆]

Jiří Kopal [*], [a], [b], Miroslav Rozložník [a], Miroslav Tůma [a], [c]

[a] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, CZ-182 07 Prague 8, Czech Republic
[b] Institute of Novel Technologies, Technical University of Liberec, Studentská 1402/2, CZ-461 17 Liberec 1, Czech Republic
[c] Faculty of Mathematics and Physics, Charles University in Prague,Czech Republic

ARTICLE INFO

Article history:
Received 29 February 2016
Received in revised form 10 August 2016
Accepted 3 October 2016
Available online xxx

Keywords:
Approximate inverse
Gram–Schmidt orthogonalization
Incomplete factorization
Multilevel methods
Preconditioned conjugate gradient method

2010 MSC:
65F08
65F25
65F50

ABSTRACT

This paper deals with adaptively preconditioned iterative methods for solving large and sparse systems of linear equations. In particular, the paper discusses preconditioning where adaptive dropping reflects the quality of preserving the relation $UZ = I$ between the direct factor U and the inverse factor Z that satisfy $A = U^TU$ and $A^{-1} = ZZ^T$. The proposed strategy significantly extends and refines the approach from [1], see also [2], by using a specific multilevel framework. Numerical experiments with two levels demonstrate that the new preconditioning strategy is very promising. Namely, we show a surprising fact that in our approach the Schur complement is better to form in a more sophisticated way than by a standard sparse matrix-matrix multiplication.

1. Introduction

An important source of linear systems with positive definite matrices is represented by discretization of partial differential equations. Such equations arise in numerous applications in science and engineering and often lead to problems with sparse matrices. Let us consider the system of linear equations

$$Ax = b, \;\; A \in \mathbb{R}^{n\times n}, \;\; x \in \mathbb{R}^n, b \in \mathbb{R}^n, \tag{1}$$

where A is the system matrix, x is the vector of unknowns and b is the right-hand side vector. Here we will assume that the system matrix A is symmetric and positive definite.

Direct methods including the sparse Cholesky method as their standard representative are often considered as a method of choice. This approach is usually based on the factorization $A = U^TU$, where U is upper triangular. An important alternative to direct solvers are iterative Krylov space methods. In the symmetric and positive definite case, a natural choice in this class is the conjugate gradient (CG) method. In order to increase the robustness of iterative methods, the system (1) needs to be transformed, and this transformation is called preconditioning.

While in some cases preconditioning can be application-based, in other situations we have to rely on general algebraic approaches. Consequently, the need for generally reliable incomplete factorizations is strong. An important preconditioning strategy is based on incomplete Cholesky factorizations, that is on factorization $A \approx \hat{U}^T\hat{U}$, where $\hat{U}$ is upper triangular. There are a lot of possibilities to determine the way to approximate the exact Cholesky factorization.

But there are a few potential problems connected to the incomplete Cholesky factorization. First, it can break down. This means that a diagonal entry computed at some factorization step is zero or negative. Such a situation can be cured by various strategies that modify the original matrix introducing in this way an additional error. Formally the incomplete Cholesky factorization can be described as an exact factorization of a perturbed matrix

$$A + \Delta\hat{E} = \hat{U}^T\hat{U}, \tag{2}$$

where the matrix $\hat{E}$ is called the factorization error. Theoretical analysis of the incomplete Cholesky factorization that takes into account

[☆] Supported by the project GA13-06684S of the Grant Agency of the Czech Republic and by the ERC project MORE LL1202 financed by the MŠMT of the Czech Republic.

[*] Corresponding author.

Email addresses: jiri@cs.cas.cz, jiri.kopal@tul.cz (J. Kopal); miro@cs.cas.cz (M. Rozložník); tuma@cs.cas.cz, mirektuma@karlin.mff.cuni.cz (M. Tůma)

http://dx.doi.org/10.1016/j.advengsoft.2016.10.005
0965-9978/© 2016 Published by Elsevier Ltd.

general matrix perturbations used in practice is very difficult and successful only in special cases, see, e.g., [3]. In addition, the bounds for the factorization error are often rough, they typically need additional assumptions and are difficult to apply if the factorization should be kept reasonably sparse. An important step to make factorization more robust is based on incomplete factors computed with the inverse-based dropping proposed by Bollhöfer and Saad [4–6]. Another attempt to get more reliable factorization is to evaluate simultaneously the direct approximate factor $\widehat{U}$ with the approximate inverse factor $\widehat{Z}$, see [7,8].

A counterpart of the Cholesky factorization is the inverse factorization introduced in [9]. It computes $A^{-1} = ZZ^T$ with Z upper triangular. Its algorithm is nothing more than the Gram–Schmidt orthogonalization of standard unit vectors with respect to a non-standard inner product induced by the matrix A. In this way we get two factors Z and U that satisfy the identities $ZU = UZ = I$. Similarly to the Cholesky factorization of A, the inverse factorization can be computed incompletely. In general, even more effort must be done to keep reasonable sparsity in the approximate inverse factorization $A^{-1} \approx \widehat{Z}\widehat{Z}^T$ computed by the approximate Gram–Schmidt orthogonalization, but such approximate inverse construction has also strong advantages. For example, we can get an incomplete Cholesky factor U without a breakdown and construct rather robust preconditioners for some classes of problems [10,11].

There is a rich history of incomplete factorizations that are based on multilevel reorderings or that explicitly use more levels in the factorization. Let us consider first the goal to achieve the multilevel effect by reorderings including also approaches for more general systems than symmetric and positive definite. In case of matrices from structured PDE discretizations we have interesting examples of recursive red-black reorderings in [12] where the author looks at conditioning of the final system matrix, studies the nested recursive factorization with two levels for nine-point difference matrices [13] and use it as a smoother in multigrid. Another combination of multigrid with incomplete factorization is described in [14]. A renumbering strategy with more levels that works also on unstructured grids was presented in [15], see also [16]. Let us also mention general matrix reorderings in [17], solving eigenvalue problems in [18] and also the use of more levels in incomplete factorizations in various applications. Such factorizations are typically used either directly, for example, for solving saddle-point problems, or as preconditioners, see, e.g., in [19–21], and also [22]. Related approximate inverse factorizations considered here is the line of research in [23–25]. We believe that the construction of the Schur complement proposed here may be combined with the framework and strategies in [26–28], but see also recent papers on other multilevel approaches from domain decomposition and nonsymmetric multilevel approximate inverse technique based on a block independent set reordering scheme and using factorized inverses as [29].

This paper presents a multilevel approach for computing the above mentioned approximate inverse factorization. The factorization uses the adaptive dropping introduced in [30], see also [1]. Here we propose the approximate inverse multilevel factorization as well as a new way to perform data transfer between levels in order to minimize the errors caused by related incomplete orthogonalization process.

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

$$\widehat{Z}^T A\widehat{Z}y = \widehat{Z}^T b, \quad x = \widehat{Z}y. \tag{3}$$

The quality of the approximation is determined by the loss of orthogonality between the column vectors of $\widehat{Z}$ defined as $\Delta\widehat{H} = \widehat{Z}^T A\widehat{Z} - I$. This quantity is an analogue of the expression $\widehat{U}^{-T} A\widehat{U}^{-1} - I$ introduced by Chow and Saad [31] as a measure of stability. It is clear that a small right residual $\Delta\widehat{G} = I - \widehat{U}\widehat{Z}$ together with a small error in Cholesky factorization (2) imply a small loss of orthogonality $\Delta\widehat{H}$. Indeed, we have

$$\begin{aligned}\widehat{Z}^T A\widehat{Z} - I &= \widehat{Z}^T \left(\widehat{U}^T\widehat{U} - \Delta\widehat{E}\right)\widehat{Z} - I \\ &= \Delta\widehat{G}^T + \Delta\widehat{G} + \Delta\widehat{G}^T\Delta\widehat{G} - \widehat{Z}^T\Delta\widehat{E}\widehat{Z}.\end{aligned} \tag{4}$$

This relation is a theoretical basis of the adaptive dropping that we will use here.

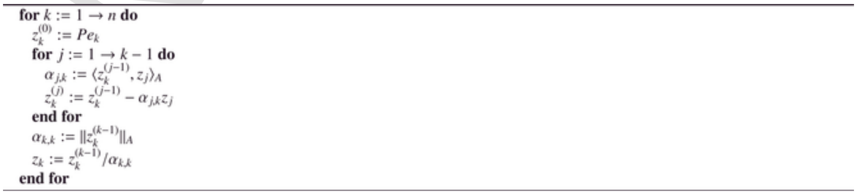
The paper is organized as follows. Basics of the underlying theory are summarized in Section 2. The multilevel scheme is described in Section 3. Experimental results showing the qualitative improvements of the new approach are shown in Section 4 and the paper is finalized by conclusions and description of future work.

2. Gram–Schmidt based approximate inverse preconditioners

Let us consider the Gram–Schmidt orthogonalization of the standard unit vectors $e_1, \dots, e_n$ 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

$$ZU = P, \tag{5}$$

where the columns of Z are A-orthonormal with $Z^TAZ = I$ and U is the upper triangular Cholesky factor of the matrix $P^TAP = U^TU$. It is clear that Z is the inverse factor satisfying $A^{-1} = ZZ^T$. The Gram–Schmidt process is summarized in Algorithm 1, where $Z = [z_1, \cdots, z_n]$ are the resulting A-orthonormal vectors and $U = [\alpha_{j,k}]$ contains the orthogonalization and normalization coefficients. Here we consider the modified version of the Gram–Schmidt process [32] that is equivalent to the SAINV algorithm [33] as explained in [2].



```
for $k := 1 \rightarrow n$ do
  $z_k^{(0)} := Pe_k$
  for $j := 1 \rightarrow k-1$ 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. Modified version of the Gram–Schmidt process with column permutation and with respect to the inner product $\langle\cdot,\cdot\rangle_A$.

Algorithm 1 computes for each k a column z_k of the factor Z using the vector Pe_k that is A-orthogonalized against the previously computed vectors $z_1, \ldots, z_{k-1}$. This organization of the computation is known as the left-looking approach. Our goal is to obtain factor U such that its entries satisfy inequalities

$$\alpha_{1,1} \geq \alpha_{2,2} \geq \ldots \geq \alpha_{n,n} > 0 \tag{6}$$

$$\alpha_{i,i}^2 \geq \sum_{j=i}^{k} \alpha_{j,k}^2, \quad k = i+1, \cdots, n. \tag{7}$$

Note that, (6) and (7) also imply

$$\alpha_{j,j} > \left|\alpha_{j,k}\right|, \quad j = 1, \ldots, n, \quad k = j+1, \cdots, n. \tag{8}$$

The permutation P that leads to U in the above mentioned form is not a priori known and has to be computed on-the-fly. In addition, Algorithm 1 requires additional precomputation of orthogonalization coefficients using the classical variant of the Gram–Schmidt process [32]. For each k and $j = k, \ldots, n$ we update the A-norms of the vectors $z_j^{(k-1)}$ as follows

$$\| z_j^{(k-1)} \|_A^2 = \| z_j^{(k-2)} \|_A^2 - 1 \left\langle z_j^{(0)}, z_{k-1} \right\rangle_A^2. \tag{9}$$

The new kth column vector $Pe_k \equiv e_i$ is chosen such that

$$\| z_i^{(k-1)} \|_A = \max_{k \leq j \leq n} \| z_j^{(k-1)} \|_A. \tag{10}$$

Permutation P is thus obtained implicitly by the application of column pivoting with the criterion (10).

It has been shown in [1,34] that the generalized Gram–Schmidt process with pivoting combined with a specific adaptive dropping may lead to successful preconditioners from the point of view sparsity and convergence of the preconditioned conjugate gradient method at the same time. Let us remind, that the generalized Gram–Schmidt process computes the same factors as the Cholesky factorization algorithm only in exact arithmetic. Its numerical properties in the finite precision arithmetic for various implementations are summarized in [35]. For a comparison with another approach [36] on similar type of engineering problems, see [34].

Without going into the details, an incomplete factorization based on dropping entries in $\widetilde{z}_k$ with respect to local thresholds

$$\tau_k = \frac{\tau}{\kappa\left(\widetilde{U}_k\right)} \tag{11}$$

has been proposed in [34]. An input parameter τ determines accuracy of the factorization (preconditioner), $\kappa(\cdot)$ denotes the condition number. Note that an extra lower subscript in the matrix notation, e.g., $A_k \in \mathbb{R}^{k \times k}$, is used to denote leading principal submatrices. In addition, we introduce an extra upper tilde notation for approximate quantities computed by the dropping technique as described in [34]. This dropping is performed by comparing components of the vector $\left|\widetilde{z}_k^{(k-1)}\right| / \| \widetilde{z}_k^{(k-1)} \|_\infty$ and corresponding threshold τ_k. Nonzero entries (fill) in the incomplete factor are allowed only at positions where elements of the vector $\left|\widetilde{z}_k^{(k-1)}\right| / \| \widetilde{z}_k^{(k-1)} \|_\infty$ are larger than τ_k.

Taking into account $\kappa\left(\widetilde{U}_1\right) \leq \kappa\left(\widetilde{U}_2\right) \leq \ldots \leq \kappa\left(\widetilde{U}\right)$ together with (11), it is clear that dropping based on quantity τ_k may tend to produce more fill-in in vectors $\widetilde{z}_k$ for increasing k. Our goal here is to store the inverse triangular factor implicitly, i.e., in a more compact way. One possible way how to improve its sparsity may be the multilevel approach introduced in next section.

3. Multilevel approximate inverse preconditioning with adaptive dropping

In this section we propose a multilevel scheme for SAINV with l_{max} levels. Assume $\widetilde{P}^{(\ell)}$, $\ell = 1, \ldots, \ell_{max}$ are permutation matrices, such that

$$\left(\widetilde{P}^{(\ell)}\right)^T A^{(\ell)} \widetilde{P}^{(\ell)} = \begin{pmatrix} B^{(\ell)} & E^{(\ell)} \\ \left(E^{(\ell)}\right)^T & F^{(\ell)} \end{pmatrix}, \quad A^{(1)} = A, \tag{12}$$

where the entries of the Cholesky factorization of $B^{(\ell)} \approx \left(\widetilde{U}^{(\ell)}\right)^T \widetilde{U}^{(\ell)}$ satisfy inequalities (6) and (7) at each level. Let the inverse of $B^{(l)}$ be approximated by $\widetilde{Z}^{(\ell)}\left(\widetilde{Z}^{(\ell)}\right)^T$ (incomplete inverse triangular factorization). Then we can write

$$\begin{pmatrix} \left(\widetilde{Z}^{(\ell)}\right)^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} B^{(\ell)} & E^{(\ell)} \\ \left(E^{(\ell)}\right)^T & F^{(\ell)} \end{pmatrix} \begin{pmatrix} \widetilde{Z}^{(\ell)} & 0 \\ 0 & I \end{pmatrix} \approx \begin{pmatrix} I & \left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)} \\ \left(\left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)}\right)^T & F^{(\ell)} \end{pmatrix}. \tag{13}$$

Eliminating the off-diagonal blocks in (13) we obtain

$$\begin{pmatrix} I & 0 \\ -\left(E^{(\ell)}\right)^T \widetilde{Z}^{(\ell)} & I \end{pmatrix} \begin{pmatrix} I & \left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)} \\ \left(E^{(\ell)}\right)^T \widetilde{Z}^{(\ell)} & F^{(\ell)} \end{pmatrix} \begin{pmatrix} I & -\left(\widetilde{Z}^{(\ell)}\right)^T \\ 0 & I \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & A^{(\ell+1)} \end{pmatrix},$$

where $A^{(\ell+1)}$ is the Schur complement of the principal leading block with respect to the matrix on the right-hand side of (13). This can be written in a standard way via matrix multiplications as

$$A^{(\ell+1)} = F^{(\ell)} - \left(\left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)}\right)^T \left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)}. \tag{15}$$

But, as we will see below, dealing with a factorization based on incomplete orthogonalization, we have more possibilities to define the system matrix for the next level. Note that the block $\left(E^{(\ell)}\right)^T \widetilde{Z}^{(\ell)}$ represents a block of the Cholesky factor of the partially decomposed matrix

$$\left(\widetilde{P}^{(\ell)}\right)^T A^{(\ell)} \widetilde{P}^{(\ell)} \approx \begin{pmatrix} \left(\widetilde{U}^{(\ell)}\right)^T & 0 \\ \left(\left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)}\right)^T & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & A^{(\ell+1)} \end{pmatrix} \times \begin{pmatrix} \widetilde{U}^{(\ell)} & \left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)} \\ 0 & I \end{pmatrix}. \qquad (16)$$

The steps described by formulas (12)–(14) are then recursively applied at all levels. In the last step $\ell = \ell_{max}$ we get $\left(\widetilde{P}^{(\ell_{max})}\right)^T A^{(\ell_{max})} \widetilde{P}^{(\ell_{max})}$ equal to $B^{(\ell_{max})}$. There are various possibilities how the levels can be defined and used, for example, by finding strong connections as in the algebraic multigrid or looking for independent sets, see, e.g., [24]. We propose a different criterion to start a new level that is based mainly on the size of $\widetilde{Z}^{(\ell)}$ (nnz $\left(\widetilde{Z}^{(\ell)}\right)$), see (19). In addition, we use reordering based on the above mentioned pivoting with respect to the diagonal entries in $\widetilde{U}$, see [1,34]. Note that the block $\left(\widetilde{Z}^{(\ell)}\right)^T E^{(\ell)}$ in the solve steps is applied in a factored form, i.e., storing $\left(\widetilde{Z}^{(\ell)}\right)^T$ and $E^{(1)}$ separately.

Having a partial factorization for some level $1 \leq 1 < 1_{max}$, let us consider construction of the system matrix for the next level, that is typically formed as the Schur complement by matrix multiplication. It is easy to see that this construction of the Schur complement (15) represents the classical Gram–Schmidt (CGS) process, i.e.,

$$\left[A^{(\ell+1)}\right]_{j,i} = \left[F^{(\ell)}\right]_{j,i} - \sum_{k=1}^{\min(j,i)} \langle z_k, z_j^{(0)} \rangle_{A^{(\ell)}} \langle z_i^{(0)}, z_k \rangle_{A^{(\ell)}}. \qquad (17)$$

An alternative way to get the next system matrix is to employ the right-looking version of the modified Gram–Schmidt (MGS) process and compute the Schur complement as

$$\left[A^{(\ell+1)}\right]_{j,i} = \left[F^{(\ell)}\right]_{j,i} - \sum_{k=1}^{\min(j,i)} \langle z_k, z_j^{(k-1)} \rangle_{A^{(\ell)}} \langle z_i^{(k-1)}, z_k \rangle_{A^{(\ell)}}, \qquad (18)$$

where the vectors $z_j^{(k-1)}$ and $z_i^{(k-1)}$ are computed using the scalar product from Algorithm 1. The practical difference between these two approaches may be enormous and we will discuss it in Section 4.

4. Numerical experiments

In this section we show results of our numerical experiments. All the experiments here were performed using Matlab$^{\text{TM}}$ with $u \approx 1.1 \cdot 10^{-16}$. In particular, we chose the matrix *bcsstk08* [37] with $\kappa(A) \approx 4.7 \cdot 10^7$, $\text{nnz}(A) = 12960$, and dimension equal to 1074 but our tables cover also other matrices from the Harwell-Boeing set. Similarly as in [1] we use a cheap evaluation of τ_k based on approximation of $\kappa\left(\widetilde{U}_k\right)$ by the ratios of extremal diagonal entries of $\widetilde{U}_k$.

In order to present advantages of multilevel framework clearly, we first show for comparison purposes results without using the multilevel framework as depicted in Fig. 1.

The condition under which we start a new level in this framework will be considered in the form

$$\text{nnz}\left(\widetilde{Z}^{(\ell)}\right) > \text{nnz}\left(A^{(\ell)}\right) \quad \text{and} \quad k > \frac{n_\ell}{2} \quad \text{and} \quad k > \text{min\_size} \quad \text{and} \quad -k > \text{min\_size},$$

where $\text{min\_size} = 100$ and n_1 denotes dimension of $A^{(1)}$. This condition has led for all considered values of τ to at most two level scheme, i.e., $\ell_{max} = 2$. Numerical results for the case when the Schur complement is formed by the coefficients of the classical Gram–Schmidt process (17), are shown in Fig. 2. It is easy to see that convergence of CG method is significantly worse. Moreover, it is necessary to note that the Schur complement $A^{(2)}$ is indefinite for all considered τ, and some remedy is necessary. Correction of the Schur complement has been done in an iterative fashion. When a breakdown occurs, we correct the Schur complement by an extra diagonal shift equal to the identity matrix multiplied by $\tau_k \cdot 2^p$ for $p = 0, 1, \ldots$ until a breakdown does not occur, see [3]. Fig. 3 depicts numerical results for the case when the Schur complement is formed by the coefficients of the modified Gram–Schmidt process (18). More accurate computation of the Schur complement delivers significantly better convergence of CG method with respect previous case. Apart from this fact, the Schur complement is positive definite for all τ, therefore corrections were not necessary. In addition, similarities between convergence without employing the multilevel framework and with multilevel framework

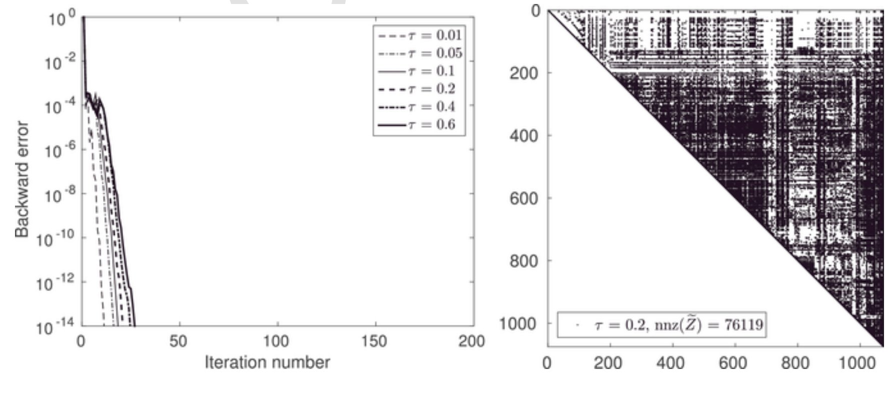


Fig. 1. One level approach: convergence of the CG method (left) and sparsity pattern of $\widetilde{Z}$ for $\tau = 0.2$ (right).

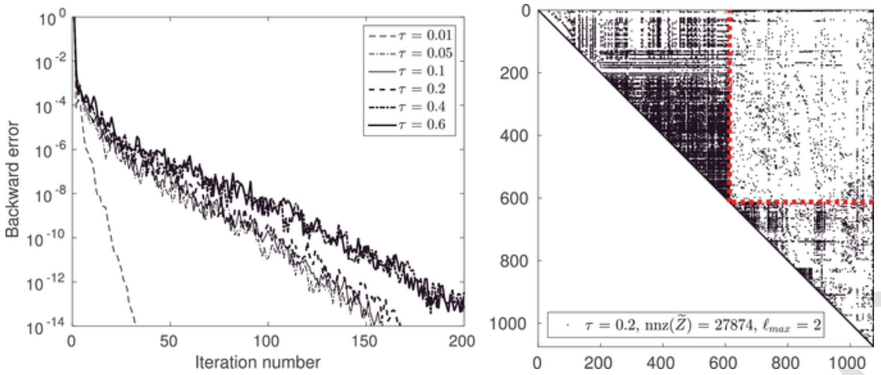


Fig. 2. Multilevel approach with CGS form of Schur complement: convergence of the CG method (left) and sparsity pattern of $\widetilde{Z}$ for $\tau = 0.2$ (right).

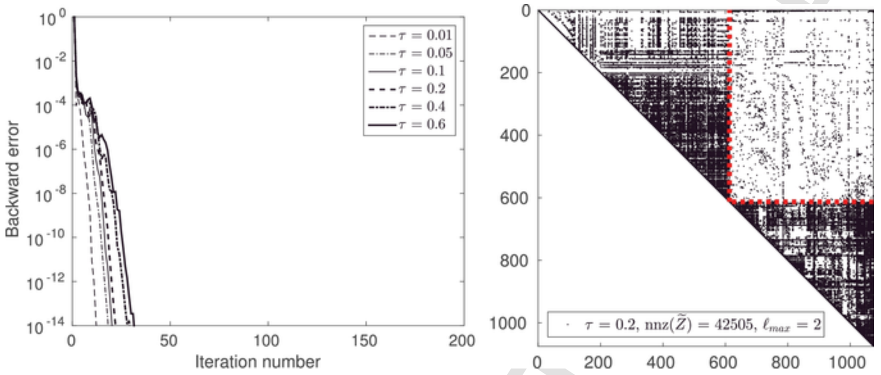


Fig. 3. Multilevel approach with MGS form of Schur complement: convergence of the CG method (left) and sparsity pattern of $\widetilde{Z}$ for $\tau = 0.2$ (right).

when the Schur complement is formed by the modified Gram–Schmidt process are very surprising. Let us also note that computation of such preconditioning is only a slightly more demanding. Number of nonzero entries of the preconditioner as a function of τ for all three approaches is depicted in Fig. 4.

For the set of the chosen matrices from [37] we summarize results of the algorithms in Tables 1–3. For these experiments we do not consider any diagonal shift. The case when a breakdown occur due to the indefinite Schur complement (its approximation) is denoted by symbol †. By the same symbol is denoted when CG does not converge within 500 iterations. Otherwise, we consider stopping criterion based on backward error less than 10^{-8}.

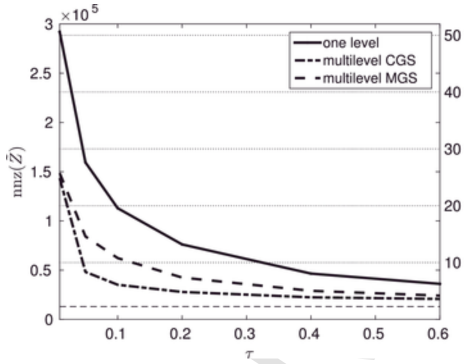


Fig. 4. Number of nonzeros of the preconditioner (for one level approach denoted as “one level”, multilevel approach with CGS form of the Schur complement (17) denoted as “multilevel CGS”, and multilevel approach with MGS form of the Schur complement (18) denoted as “multilevel MGS”) as a function of τ (tiny dotted horizontal lines denote relative fill-in with respect dense triangular factor; tiny dashed horizontal line denotes nnz(A)).

5. Conclusion and open questions

In this paper we have proposed a multilevel scheme for the factorized approximate inverse preconditioning based on the Gram–Schmidt process with the inner-product induced by a symmetric and positive definite matrix. We have shown that the multilevel framework may significantly reduce preconditioner sizes. Moreover, the convergence of the conjugate gradient method stays in the case of the new Schur complement construction based on the modified Gram–Schmidt process essentially the same as without employing the multilevel framework. While here we were interested in presenting practical algorithmic features of the approach, full theoretical analysis and construction of preconditioners based on $\widetilde{U}$ are open questions for the future.

Acknowledgments

We would like to thank the anonymous reviewers for their feedback.

Table 1
One-level algorithm.

matrix		τ					
		0.01	0.05	0.1	0.2	0.4	0.6
bcsstk08	$\mathrm{nnz}\left(\widetilde{Z}\right)$	292,051	159,425	112,955	76,119	46,520	36
	iters.	7	11	13	14	16	17
bcsstk09	$\mathrm{nnz}\left(\widetilde{Z}\right)$	313,018	60,145	28,778	10,880	2280	19
	iters.	18	53	78	127	202	21
bcsstk10	$\mathrm{nnz}\left(\widetilde{Z}\right)$	89,218	60,755	50,150	38,929	29,616	23
	iters.	5	10	16	28	40	71
bcsstk19	$\mathrm{nnz}\left(\widetilde{Z}\right)$	80,694	60,047	52,783	51,606	42,675	42
	iters.	43	77	112	166	235	35
bcsstk27	$\mathrm{nnz}\left(\widetilde{Z}\right)$	195,029	61,590	32,035	16,059	9568	69
	iters.	9	24	38	65	90	12
msc01050	$\mathrm{nnz}\left(\widetilde{Z}\right)$	488,146	216,140	140,945	125,907	112,625	67
	iters.	15	43	33	45	51	62
nos2	$\mathrm{nnz}\left(\widetilde{Z}\right)$	157,276	155,732	153,972	129,746	33,443	13
	iters.	6	24	49	131	†	†
nos3	$\mathrm{nnz}\left(\widetilde{Z}\right)$	95,461	25,815	11,402	4087	1953	13
	iters.	14	30	49	91	133	19
nos7	$\mathrm{nnz}\left(\widetilde{Z}\right)$	192,117	80,622	63,436	48,322	36,783	30
	iters.	5	8	8	13	16	21

Table 2
Multilevel algorithm, Schur complement formed by the coefficients of the CGS (17).

matrix		τ					
		0.01	0.05	0.1	0.2	0.4	0.6
bcsstk08	$\mathrm{nnz}\left(\widetilde{Z}\right)$	†	†	†	†	†	†
	iters.						
bcsstk09	$\mathrm{nnz}\left(\widetilde{Z}\right)$	147,156	†	24,193	10,934	2280	1963
	iters.	19		118	115	202	214
bcsstk10	$\mathrm{nnz}\left(\widetilde{Z}\right)$	53,366	†	†	†	†	†
	iters.	8					
bcsstk19	$\mathrm{nnz}\left(\widetilde{Z}\right)$	†	†	†	†	†	†
	iters.						
bcsstk27	$\mathrm{nnz}\left(\widetilde{Z}\right)$	109,770	59,031	32,188	16,186	9651	7144
	iters.	10	25	38	64	89	120
msc01050	$\mathrm{nnz}\left(\widetilde{Z}\right)$	224,408	†	†	†	†	†
	iters.	16					
nos2	$\mathrm{nnz}\left(\widetilde{Z}\right)$	†	†	†	†	†	†
	iters.						
nos3	$\mathrm{nnz}\left(\widetilde{Z}\right)$	27,517	17,525	11,470	4220	1956	1373
	iters.	15	32	49	91	133	189
nos7	$\mathrm{nnz}\left(\widetilde{Z}\right)$	95,905	38,478	31,376	24,675	19,851	17,135
	iters.	5	8	9	12	15	20

Table 3
Multilevel algorithm, Schur complement formed by the coefficients of the MGS (18).

matrix		τ					
		0.01	0.05	0.1	0.2	0.4	0.6
bcsstk08	$\mathrm{nnz}\left(\widetilde{Z}\right)$	148,576	84,247	62,277	42,505	29,008	24,148
	iters.	8	12	14	15	20	22
bcsstk09	$\mathrm{nnz}\left(\widetilde{Z}\right)$	143,326	36,615	23,218	10,934	2280	1963
	iters.	17	56	90	115	202	214
bcsstk10	$\mathrm{nnz}\left(\widetilde{Z}\right)$	50,817	36,758	31,757	26,934	23,652	22,564
	iters.	5	11	21	40	94	139
bcsstk19	$\mathrm{nnz}\left(\widetilde{Z}\right)$	24,042	17,616	15,169	13496	11,458	11,270
	iters.	72	346	386	410	499	†
bcsstk27	$\mathrm{nnz}\left(\widetilde{Z}\right)$	105,991	58,948	32,188	16,186	9651	7144
	iters.	9	24	38	64	89	120
msc01050	$\mathrm{nnz}\left(\widetilde{Z}\right)$	238,724	108,687	81,564	70,209	62,127	45,885
	iters.	15	37	31	44	49	62
nos2	$\mathrm{nnz}\left(\widetilde{Z}\right)$	81,732	80,605	79,478	69,321	17,611	9899
	iters.	6	24	54	165	†	†
nos3	$\mathrm{nnz}\left(\widetilde{Z}\right)$	30,582	17,888	11,470	4220	1956	1373
	iters.	13	31	49	91	133	189
nos7	$\mathrm{nnz}\left(\widetilde{Z}\right)$	96,621	38,914	32,157	25,821	20,903	17,904
	iters.	5	8	9	12	16	20

References

[1] J. Kopal, M. Rozložník, M. Tåuma, Factorized approximate inverses with adaptive dropping, SIAM J Sci Comput 38 (3) (2016) A1251–A1279.

[2] M. Rozložník, M. Tåuma, A. Smoktunowicz, J. Kopal, Rounding error analysis of orthogonalization with a non-standard inner product, BIT Numer Math 52 (2012) 1035–1058.

[3] T.A. Manteuffel, An incomplete factorization technique for positive definite linear systems, Math Comput 34 (1980) 473–497.

[4] M. Bollhöfer, A robust ILU with pivoting based on monitoring the growth of the inverse factors, Linear Algebra Appl 338 (2001) 201–218.

[5] M. Bollhöfer, A robust and efficient ILU that incorporates the growth of the inverse triangular factors, SIAM J Sci Comput 25 (1) (2003) 86–103.

[6] M. Bollhöfer, Y. Saad, On the relations between ILUs and factored approximate inverses, SIAM J Matrix Anal Appl 24 (1) (2002) 219–237.

[7] R. Bru, J. Marín, J. Mas, M. Tåuma, Balanced incomplete factorization, SIAM J Sci Comput 30 (5) (2008) 2302–2318.

[8] R. Bru, J. Marín, J. Mas, M. Tåuma, Improved balanced incomplete factorization, SIAM J Matrix Anal Appl 31 (5) (2010) 2431–2452.

[9] M. Benzi, C.D. Meyer, M. Tåuma, A sparse approximate inverse preconditioner for the conjugate gradient method, SIAM J Sci Comput 17 (5) (1996) 1135–1149.

[10] M. Benzi, R. Kouhia, M. Tåuma, An assessment of some preconditioning techniques in shell problems, Commun Numer Methods Eng 14 (1998) 897–906.

[11] M. Benzi, R. Kouhia, M. Tåuma, Stabilized and block approximate inverse preconditioners for problems in solid and structural mechanics, Comput Methods Appl Mech Eng 190 (49–50) (2001) 6533–6554.

[12] C. Brand, An incomplete-factorization preconditioning using repeated red-black ordering, Numer Math 61 (4) (1992) 433–454.

[13] O. Axelsson, V. Eijkhout, The nested recursive two-level factorization method for nine-point difference matrices, SIAM J Sci Stat Comput 12 (6) (1991) 1373–1400.

[14] A. Reusken, Multigrid with matrix-dependent transfer operators for convection-diffusion problems, Multigrid methods, IV (Amsterdam, 1993), Vol. 116 of Internat. Ser. Numer. Math., Birkhäuser, Basel, 1994269–280.

[15] A. van der Ploeg, E.F.F. Botta, F.W. Wubs, Nested grids ILU-decomposition (NGILU), J Comput Appl Math 66 (1–2) (1996) 515–526.

[16] E.F.F. Botta, A. van der Ploeg, Renumbering strategies based on multi-level techniques combined with ILU-decompositions, Comput Math Math Phys 37 (11) (1997) 1252–1258.

[17] E.F.F. Botta, F.W. Wubs, Matrix renumbering ILU: an effective algebraic multilevel ILU preconditioner for sparse matrices, SIAM J Matrix Anal Appl 20 (4) (1999) 1007–1026.

[18] G.G. Sleijpen, F. Wubs, Exploiting multilevel preconditioning techniques in eigenvalue computations, SIAM J Sci Comput 25 (4) (2003/04) 1249–1272.

[19] A.C. de Niet, F. Wubs, Numerically stable LDL^T-factorization of F-type saddle point matrices, IMA J Numer Anal 29 (1) (2009) 208–234.

[20] A.C. de Niet, F. Wubs, Two preconditioners for saddle point problems in fluid flows, Internat J Numer Methods Fluids 54 (4) (2007) 355–377.

[21] F. Wubs, J. Thies, A robust two-level incomplete factorization for (Navier-Stokes saddle point matrices, SIAM J Matrix Anal Appl 32 (4) (2011) 1475–1499.

[22] D. Osei-Kuffuor, R. Li, Y. Saad, Matrix reordering using multilevel graph coarsening for ILU preconditioning, SIAM J Sci Comput 37 (1) (2015) A391–A419.

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

[24] Y. Saad, B. Suchomel, ARMS: an algebraic recursive multilevel solver for general sparse linear systems, Numer Linear Algebra Appl 9 (5) (2002) 359–378.

[25] Z. Li, Y. Saad, M. Sosonkina, pARMS: a parallel version of the algebraic recursive multilevel solver, Numer Linear Algebra Appl 10 (5–6) (2003) 485–509.

[26] Y. Xi, R. Li, Y. Saad, An algebraic multilevel preconditioner with low-rank corrections for sparse symmetric matrices, SIAM J Matrix Anal Appl 37 (1) (2016) 235–259.

[27] R. Li, Y. Saad, Divide and conquer low-rank preconditioners for symmetric matrices, SIAM J Sci Comput 35 (4) (2013) A2069–A2095.

[28] R. Li, Y. Xi, Y. Saad, Schur complement based domain decomposition preconditioners with low-rank corrections, Numer Linear Algebra Appl 23 (2016) 706–729.

[29] C.K. Filelis-Papadopoulos, G.A. Gravvanis, A class of generic factored and multilevel recursive approximate inverse techniques for solving general sparse systems, Eng Comput 33 (1) (2016) 74–99.

[30] J. Kopal, M. Rozložník, M. Tåuma, Approximate inverse preconditioning for the conjugate gradient method, in: B.H.V. Topping, P. Iványi (Eds.), Proceedings of the third international conference on parallel, distributed, grid and cloud computing for engineering, civil-comp press, Stirlingshire, UK, 2013http://dx.doi.org/10.4203/ccp.101.

[31] E. Chow, Y. Saad, Experimental study of ILU preconditioners for indefinite matrices, J Comput Appl Math 86 (2) (1997) 387–414.

[32] N.J. Higham, Accuracy and stability of numerical algorithms, 2nd ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[33] M. Benzi, J.K. Cullum, M. Tåuma, Robust approximate inverse preconditioning for the conjugate gradient method, SIAM J Sci Comput 22 (4) (2000) 1318–1332.

[34] J. Kopal, M. Rozložník, M. Tåuma, Approximate inverse preconditioners with adaptive dropping, Adv Eng Software 84 (2015) 13–20.

[35] J. Kopal, Generalized Gram–Schmidt process: its analysis and use in preconditioning, Technical University in Liberec, 2014, Ph.d. thesis.

[36] C. Filelis-Papadopoulos, G. Gravvanis, Distributed generic approximate sparse inverses, J Supercomput 70 (2014) 365–384.

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