A two-level sparse approximate inverse preconditioner for unsymmetric matrices

Ke Chen* and Martyn D. Hughes†

Abstract

Sparse approximate inverse (SPAI) preconditioners are effective in accelerating iterative solution of a large class of unsymmetric linear systems and their inherent parallelism has been widely explored. The effectiveness of SPAI relies on the assumption of the unknown true inverse admitting a sparse approximation. Furthermore, for the usual right SPAI, one must restrict the number of nonzeros in each column to control the overall construction cost and this restriction can reduce the effectiveness of such preconditioners. To extend the applicability of SPAI, this paper proposes to use two-level preconditioning: possible dense columns of the true inverse, skipped by right SPAI (column-wise), will be better approximated by left SPAI (row-rise). Essentially we approximate the true inverse by sparse matrices via a Gauss-Jordan like decomposition. Numerical experiments on a class of benchmark test matrices show that our new idea of two-level preconditioning can lead to a major enhancement to the standard SPAI method.

Keywords: Sparse approximate inverses, iterative solution, two-level preconditioners, Gauss-Jordan decomposition, unsymmetric systems.

AMS subject class: 65F05, 65F10, 65F15, 65Y05, 65F50.

1 Introduction

Sparse approximate inverse (SPAI) represents one effective preconditioning idea, among many competing methods, for accelerating iterative solution of a class of unsymmetric linear systems. The most attractive feature of SPAI is its inherent parallelism, without which the setup process can be extremely expensive. In this paper, we are concerned with SPAI for a particular class of difficult problems and propose an improvement. These problems are characterized as those matrices whose inverse cannot be approximated by a sparse matrix in some of its columns. Thus our proposed method will improve the robustness of SPAI methods. However other preconditioning methods (e.g. the incomplete LU decomposition preconditioner (ILU) [44]) may be equally effective for the problem types concerned.

Below we shall first define the basic notation and then give a general introduction to the topic. We consider the iterative solution of a large linear system of equations

$$Ax = b, \tag{1}$$

*Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. Email: k.chen@liverpool.ac.uk (for correspondence), URL: http://www.liv.ac.uk/∼cmchenke

†Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. Email: mdhughes@liverpool.ac.uk.

where $x,\ b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ is sparse and unsymmetric, by Krylov subspace methods [1, 2, 31, 32, 44, 46]. Convergence of an iterative solver is usually accelerated by a preconditioner M i.e. instead of using (1), we solve (left preconditioning)

$$M_2 A x = M_2 b$$

or (right preconditioning)

$$AM_1 y = b, \quad x = M_1 y. \tag{2}$$

The design of an effective preconditioner M (here M_1 or M_2) for general problems is a mathematical challenge [18]. However, more and more specific problems are being successfully solved using preconditioned iterative solvers. Clearly one desires to ensure that $M \approx A^{-1}$ in some way and at the same time that minimal additional computation is needed.

In this paper we aim to improve a right preconditioner M_1, which is constructed as a sparse approximate inverse in the Frobenius norm

$$\min_{M_1} \|AM_1 - I\|_F^2, \tag{3}$$

by combining it with a left level 2 preconditioner M_2

$$\min_{M_2} \|M_2(AM_1) - I\|_F^2, \tag{4}$$

where the nonzeros of M_1, M_2 fall into some suitably prescribed sparsity patterns. The method (3) has been studied by many researchers [4, 21, 22, 26, 33, 35, 36, 39, 40, 47]. The use of the F-norm naturally decouples (3) into n least squares problems

$$\min_{m_j} \|Am_j - e_j\|_2^2, \quad j = 1, \cdots, n, \tag{5}$$

where $M_1 = [m_1,\ \ldots,\ \ m_n]$ and $I = [e_1,\ \ldots,\ \ e_n]$. This so-called right preconditioner is column based and frequently used; construction of a left preconditioner is similarly developed and is row based [33]. In this paper we shall consider both types of preconditioners. However, an alternative form of explicit sparse approximate inverse preconditioners is the factorized triangular preconditioner

$$\min_{W,\ Z} \|W^T AZ - I\|_F^2 \tag{6}$$

as proposed and studied in the factorized sparse approximate inverse (FSAI) work [39, 40, 41] for the symmetric case and the approximate inverse (AINV) work [5, 6, 9] for the unsymmetric case. Here both W and Z are sparse upper triangular matrices with $W = Z$ for the symmetric case. With (6), the preconditioned system of (1) is the following $W^T AZy = W^T b,\ x = Zy$. This preconditioning idea may be viewed as stemmed from the fractionization technique as discussed in [30, 51].

A common assumption in constructing M_1 in (3) is that A^{-1} admits a sparse approximation which is characterized by some pattern $\mathbb{S}$. However, only in relatively few cases can the pattern $\mathbb{S}$ be suitably

specified a priori: one well known case is when A is diagonally dominant [16, 17, 27] and another case is when A has a finger-like pattern arising from a wavelet discretization of special differential and integral operators [25]. For a general matrix A, searching for a suitable $\mathbb{S}$ adaptively has been suggested in [23, 24, 35, 33]. These adaptive methods have been shown to be effective for a large class of problems where the following estimate can be achieved efficiently

$$\|Am_j - e_j\|_2 \leq \epsilon \tag{7}$$

(where ϵ is small; say $\epsilon = 0.4$ as in [35]). Inherent parallelism is the main advantage of SPAI type preconditioners which has been explored in [22, 33, 35]. Applications of the SPAI idea to other problems are reported in [14, 15, 25, 48]. Nevertheless, there remain many problems that SPAI based methods cannot solve efficiently.

One of the main weakness of SPAI methods stems from the above mentioned restriction of the number of nonzeros in any single column of M_1 (since construction of the preconditioner with too many nonzeros can be as expensive as a direct solver). This restriction differs from that of the total number of nonzeros in M_1 being comparable to A. In the special case of a right preconditioner M_1 requiring some nearly dense columns, a left preconditioner generated using the row version of SPAI must be considered for efficiency consideration [33]. For either version, it is not possible for an approximate inverse to acquire both dense rows and columns. Therefore not surprisingly, some matrix problems are beyond the scope of the current SPAI methods [21, 33, 35]. One notes that the same argument may not be applicable to the FSAI type preconditioners [41] although other robustness problems exist.

This paper addresses some problem classes where a right preconditioner generated by SPAI is not very effective and finds an improved method based on two-level sparse approximate inverse preconditioning. Our proposed method appears to have increased the robustness of SPAI methods. Although this work applies to the unsymmetric case, similar idea of two-level preconditioning for the symmetric case but for a different FSAI method has been proposed in [38, 39, 41]. For the unsymmetric case, two-level preconditioning ideas based on deflation techniques can be found in [13] and the references therein. Beyond the class of problems which our improved preconditioner can solve effectively, other preconditioners such as the ILU should be considered and in fact it can be beneficial for the SPAI and ILU ideas to be combined [10].

2 Sparse approximate inverse preconditioners

SPAI preconditioners from (3) are essentially determined by the sparsity pattern $\mathbb{S}$ that approximates the inverse A^{-1}. Here we summarise the adaptive approaches of [33, 35] that find a pattern $\mathbb{S}$ from any initial guess. See also [42] for another exposition.

Consider the solution of the least squares problem from (5) i.e. $A\mathbf{m}_j = e_j,\ \ j = 1, \ldots, n$. The idea of selecting $\mathbb{S}$ is to approximate accurately the dense solution vector $\mathbf{m}_j$, column j of matrix A^{-1}, by some sparse vector with the least number of nonzeros. Let $\mathbf{m}_j$ be approximated by some initial sparse vector m_j which has k_j nonzeros. Denote by matrix C the k_j columns of A corresponding to the nonzero positions of m_j and $\hat{m}_j$ be the reduced vector of m_j containing only nonzeros. Then $Am_j = C\hat{m}_j$ and the idea of an adaptive approach is to augment C by new columns of A (i.e. increase nonzero positions in m_j) adaptively to better approximate $\mathbf{m}_j$.

To add one new column c to C, solve

$$\sigma_{+c} = \min_{z \in \mathbb{R}^{k_j},\ \xi \in \mathbb{R}} \|Cz + \xi c - e_j\|_2^2, \tag{8}$$

where c is chosen from the remaining columns of A that intersect with nonzeros' indices of $[C\ e_j]$ or the residual $r_j = e_j - Cm_j$; see [33]. A simplified version of (8) is the following approximation (taking $z = \hat{m}_j$)

$$\sigma_{+c}^{\text{approx}} = \min_{\xi \in \mathbb{R}} \|Cz + \xi c - e_j\|_2^2 = \min_{\xi \in \mathbb{R}} \|\xi c - r_j\|_2^2, \tag{9}$$

as seen in [35]. The two minimisation problems can be directly solved to give the following [8]

$$\sigma_{+c} = \sigma - \frac{(c^T r_j)^2}{\rho^2}, \qquad \sigma_{+c}^{\text{approx}} = \sigma - \frac{(c^T r_j)^2}{\|c\|_2^2}$$

respectively, where $\sigma = \|r_j\|^2$, $[C\ c] = [Y\ y] \begin{pmatrix} R & r \\ 0 & \rho \end{pmatrix}$ with $C = Y \begin{pmatrix} R \\ 0 \end{pmatrix}$ the reduced QR decomposition.

Once the least square problem is solved, a new approximation to $\mathbf{m}_j$ is obtained and the adaptive procedure is repeated if the new residual is not small enough with regard to (7).

Remark 1 *Theoretically minimizing σ_{+c} finds a better approximation than from $\sigma_{+c}^{\text{approx}}$, for the purpose of solving (3), at each step. However, there is no guarantee of a global minimizer for the essentially multi-dimensional problem — a situation somewhat mimicking the weakness of the steepest descent method [32]. The multi-dimensional problem may be posed as follows:*

$$\sigma_{+C} = \min_{C;\ m_j} \|Cm_j - e_j\|_2^2 = \min_{c_k \in (a_1, \ldots, a_n);\, \xi_k \in \mathbb{R}} \|c_1\xi_1 + \cdots + c_\ell \xi_\ell - e_j\|_2^2$$

where a_k is column k of A, ξ_k's are the elements of $\hat{m}_j$ (i.e. the nonzero components of m_j) and hence C is chosen from any ℓ (a prescribed integer) columns of matrix A.

Practically, for a class of sparse matrices, solving the multi-dimensional minimisation is not necessary so both methods shown above (based on one-dimensional minimisation) can work well. But it is not difficult to find surprising examples. For instance, the adaptive SPAI method of [35] applied to a triangular matrix A will not find a strictly triangular matrix as an approximate inverse unless one restricts the sparsity pattern (§3); of course, FSAI type methods will be able to return a triangular matrix

as an approximation for this case. Although it remains to find an efficient way to solve the above multi-dimensional problem, narrowing down the choice for pattern $\mathbb{S}$ is regarded as an effective approach to speeding up the basic SPAI methods of type (3).

3 Acceleration using a priori patterns

We now discuss methods of selecting the initial sparsity pattern $\mathbb{S}$. For a class of problems, specifying suitable a priori patterns for the approximate inverse removes the need for any adaptive procedure and thus dramatically speeds up the SPAI preconditioner construction. With such a pattern, solving (3) yields the required preconditioner M_1. This is possible for many useful problems. For matrices arising from discretization of a class of partial differential equations, the so-called powers of sparsified matrices (PSM) methods have been found to give satisfactory and desirable patterns $\mathbb{S}$ [21, 12, 47]. For matrices from boundary integral operators, the near neighbour patterns have been shown to be satisfactory [16, 17, 19, 50]. The analytical approach of near neighbours is different from but related to the algebraic approach of PSM.

The theoretical basis for PSM patterns comes from analytical expressions for A^{-1} in terms of A. From the Cayley-Hamilton theorem, A satisfies the characteristic polynomial

$$A^n + a_{n-1}A^{n-1} + \cdots + a_1 A + a_0 I = 0. \tag{10}$$

Hence, if A is non-singular, $A^{-1} \in \text{span}(I,\ A,\ A^2,\ \ldots,\ A^{n-1})$. Alternatively if $\|A\| < 1$, we may also use the geometric series to expand $A^{-1} = [I - (I - A)]^{-1}$ in terms of powers of $(I - A)$. These algebraic considerations may also be explained by using the graph theory [21] and have recently been explored in [12, 21, 36].

Suitable scaling of a matrix is important before sparsification. In [21], symmetric scaling by diagonal matrices is suggested. However our experience has shown that a better scaling method is the permutation and scaling method by Duff and Koster [28]. In one particular variant, matrix A is permuted and scaled from both sides so that the product of diagonal entries of the resulting matrix is the largest. For instance, the following matrix A on the left is permuted and scaled to B on the right

$$A = \begin{pmatrix} 100 & 20 & & \\ 20 & 2 & -40 & \\ & 2 & 1 & 3 \\ & & 5 & 2 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & & & \\ & \frac{1}{2} & & \\ & & 1 & \\ & & & \frac{3}{2} \end{pmatrix} A \begin{pmatrix} \frac{1}{10} & & & \\ & \frac{1}{2} & & \\ & & \frac{1}{10} & \\ & & & \frac{1}{3} \end{pmatrix} P = \begin{pmatrix} 1 & & 1 & \\ 1 & -1 & \frac{1}{2} & \\ & \frac{1}{20} & 1 & 1 \\ & \frac{3}{8} & & 1 \end{pmatrix},$$

which is more amenable to sparsification by *global thresholding*, where $P = [e_1\ e_3\ e_2\ e_4]$ is a permutation matrix. In fact, one can verify that the most important elements (the largest but always including the diagonals) of both matrix B^{-1} and B^3 follow some similar pattern $\mathbb{S}$.

However, even for a sparsified matrix A_0 (out of A), the number of nonzeros in its high powers can grow quickly to approach n^2 so in practice only low powers of A_0 can usually be used. Use 'drop' to

denote a sparsification process; then $A_0 = drop(A)$ and powers are given by $A_i = A_0^{i+1}$ or, if less nonzeros are desired, $A_i = drop(A_{i-1}A_0)$. Finally the sought pattern $\mathbb{S}$ is taken from the graph of A_i. In this paper we shall mainly use $i = 3$.

4 A two-level preconditioner

We now formulate our two-level preconditioner in details and specify the types of problems that are dealt with. Assume that a SPAI preconditioner M_1 from solving (3) has been obtained for (1) and the preconditioned system is thus

$$AM_1 y = b. \tag{11}$$

However we further assume that the right preconditioner $M_1 = [m_1, \ldots, m_n]$ does not satisfy (7) in k of its n columns. That is to say,

$$AM_1 = I + E_1 + E_2, \tag{12}$$

where $\|E_2\|_F$ is very small and $I + E_1$, not necessarily small, is an elementary Gauss-Jordan matrix with k dense columns. Here we are concerned with the case of k being relatively small with respect to n. In the extreme case of $k = 0$, M_1 is already effective and so there is no need to improve M_1 while the case of $k \approx n$ implies that M_1 is ineffective and our method will not apply; then further study is required.

Under the above assumptions, we propose to further precondition (12) by $M_2 = (I + E_1)^{-1}$

$$M_2 AM_1 y = M_2 b, \quad x = M_1 y. \tag{13}$$

This new preconditioner will be effective if the preconditioned matrix

$$M_2 AM_1 = (I + E_1)^{-1}(I + E_1 + E_2) = I + (I + E_1)^{-1} E_2$$

is a smaller perturbation of I than (12). For a simple case, we can establish this statement more precisely (see Appendix 2). Such smaller perturbations would help to accelerate an iterative solver [18, 29, 43].

Note that after a symmetric permutation, the Gauss-Jordan type matrix $(I + E_1)$ can be written in the lower block triangular form

$$P_1(I + E_1)P_1^T = \begin{pmatrix} A_1 & 0 \\ A_2 & I_2 \end{pmatrix} = T \tag{14}$$

where A_1 is a matrix of size $k \times k$, A_2 of $(n-k) \times k$ and I_2 of $(n-k) \times (n-k)$. The exact inverse of this lower block triangular matrix T is

$$T^{-1} = \begin{pmatrix} A_1^{-1} & 0 \\ -A_2 A_1^{-1} & I_2 \end{pmatrix} \quad so \quad M_2 = P_1^T T^{-1} P_1.$$

This suggests that we only need to approximate the small $k \times k$ matrix A_1^{-1} in order to work out our level 2 preconditioner — the left $n \times n$ approximate inverse M_2 in (13). The overall algorithm can be summarized as follows

Algorithm 1

1. *For a tolerance tol and integer nzmax (the maximal number of nonzeros allowed per column such that nzmax = NNZ/n), implement the SPAI algorithm, solving (3), to obtain the right preconditioner* M_1.

2. *Identify and record the k sparsified columns of* AM_1 *that do not satisfy the tolerance tol.*

3. *Identify the permutation matrix* P_1 *and compute the* $k \times k$ *SPAI approximation for* A_1.

4. *Compute the left preconditioner* M_2.

5. *Solve the preconditioned system (13) by a Krylov subspace method.*

Here sparsification of AM_1 is important as it usually much less sparse than A and M_1 alone, and hence A_1 is implicitly sparsified. Note that in the adaptive SPAI approaches (Section 2), AM_1 is available (as by-products) via the monitoring residual vectors r_j. Hence our algorithm can be coupled naturally with an adaptive SPAI approach. With non-adaptive SPAI, AM_1 has to be approximated.

It should also be remarked that, instead of the two-sided scheme of (13), we can similarly propose the second level preconditioner M_2 differently (from the right)

$$AM_1M_2y = b, \quad x = M_1M_2y. \tag{15}$$

All other discussions will follow as well. However we shall mainly study (13) in this work.

We now discuss the issue of complexity and the choice of A_1. Clearly the size k of matrix A_1 is an indication of the level of difficulties in approximating A^{-1} by M_1. In most cases where our algorithm is particularly useful, we can assume that k is small. Then we may use a direct solver to compute A_1^{-1}. Thus the additional cost of using M_2 is simply $O(k^3) + O(nk^2) \approx 2k^3 + 2nk^2$. However for large k (e.g. $k = n$), we simply call an existing SPAI solver for the second time and the overall cost may be doubled. As with all SPAI preconditioners, in practical realizations, one should use parallel versions of Algorithm 1 to gain efficiency. Note that one may also take $M_2 = A_1^{-1}$ directly in (14) and implement the level two preconditioner $y = M_2x$ from solving $A_1y = x$, and this gives rise to a mixed preconditioning strategy.

One simplification of A_1 may result from selecting at most a fixed number k_{fix} columns of AM_1 that have the largest least-squares-errors $\|Am_j - e_j\|_2$. For example, set $k_{\text{fix}} \leq nzmax$. However for some extremely hard problems, this selection may not be sufficient.

Another possibility is to reset these identified $n - k$ columns of M_1 to unit vectors and then $AM_1 = I + E_1 + E_2$ becomes more pronounced as a Gauss-Jordan decomposition. A drawback of this approach is a possible scaling problem associated with matrix AM_1 thus complicating the further approximation by M_2.

For a general method, leaving out the issue of complexity, we expect a continuing repeated application of the SPAI idea will asymptotically generate an identity matrix

$$\cdots M_{2t} \cdots M_2 A M_1 \cdots M_{2t-1} \cdots = I.$$

In this case, intermediate products are not sparsified. Thus one envisages that a difficult matrix problem may need more than 2 preconditioning matrices. However we have not investigated this possibility further.

Remark 2 *In finding the approximate inverse of matrix (14), one might apply a SPAI method to the whole matrix (instead of applying to submatrix A_1). In this case, care must be taken to ensure that the zero positions in the right $(n-k)$ columns are not filled otherwise the level 2 preconditioner M_2 will not be effective. Note also that this idea becomes less attractive for formulation (15) because the second preconditioner may not be allowed to contain dense columns for the sake of efficiency.*

Although we are concerned with unsymmetric systems, similar two-level preconditioning strategies based on triangular preconditioners for FSAI in solving symmetric systems have been suggested in [38, 39, 41]. There the choice of the second preconditioner M_2 is made to approximate a banded form of $M_1^T A M_1$. There does not appear to exist any two-level work generalizing the FSAI formulation (6) for unsymmetric systems.

5 Numerical experiments

We shall present numerical results that demonstrate the improvements obtained using the new two level preconditioner (Algorithm 1) on the standard SPAI method. The test examples are selected as those benchmark problems that either cannot be solved by the standard SPAI methods or can only be solved with relatively low efficiency. Of course, for many other problems that can already be solved efficiently by the standard SPAI methods, our method would give the same results. It should be remarked that some of test problems may equally solved efficiently (or even more efficiently) by other other methods e.g. the ILU type method. However we shall not compare with such methods, in order to focus on our main idea of improving SPAI.

We select two sets of experiments. In Set 1, we aim to compare the new two-level Algorithm 1 with other methods. In Set 2, we further test the effectiveness of our new algorithm for larger systems. Let ℓ denote the maximal number of nonzero elements allowed in preconditioner M_1. We shall compare our new method with two existing methods

1. STD — the standard SPAI method searching adaptively for the pattern $\mathbb{S}$ [35] for $\ell = nzmax$.
2. PSM — the SPAI method using a priori patterns $\mathbb{S}$ [21] of $A_3 = A_0^3$ for $\ell = 2\ nzmax$.
3. New I — our new Algorithm 1 that selects at most $\ell = nzmax$ columns of AM_1 in constructing M_2 as discussed in Remark 2.

4. New II — our new Algorithm 1 that selects all columns of AM_1 (that are difficult to approximate by M_1) in constructing M_2. Again $\ell = nzmax$ for M_1.

Here the total number of nonzeros in preconditioner M_1, NNZ, is specified as a multiple of $nnz(A)$ — the number of nonzeros in A. We then take $nzmax = NNZ/n$. For our new methods (New I, II), in computing M_1, we have allowed the first $NNZ/2$ nonzero positions defined by PSM patterns with the remaining ones by adaptive searches.

The following benchmark problems are selected[1]

Data Set 1

1. The NUCL set from modelling an advanced gas cooled reactor core:
   nnc261 with $n = 261$ and $nnz(A) = 1500$, and *nnc666* with $n = 666$ and $nnz(A) = 4044$.
2. *gemat11* from the optimal power flow problem with $n = 4929$ and $nnz(A) = 33185$.
3. *pores3* from the reservoir simulation with $n = 532$ and $nnz(A) = 3474$.
4. *sherman3* from the 3D oil reservoir challenge matrices with $n = 5005$ and $nnz(A) = 20033$.
5. *gre1107* from the computer systems simulation with $n = 1107$ and $nnz(A) = 5664$.
6. The finite element matrices from graded L-shapes:
   lshp1009 with $n = 1009$ and $nnz(A) = 3937$, *lshp2614* with $n = 2614$ and $nnz(A) = 10297$.
7. The finite element modelling using a penalty formulation:
   fidap023 with $n = 1409$ and $nnz(A) = 43481$, and *fidapm33* with $n = 2353$ and $nnz(A) = 23145$.

Data Set 2

1. Finite element analysis of cylindrical shells: *s2rmt3m1* with $n = 5489$ and $nnz(A) = 112505$.
2. The finite element modelling using a penalty formulation:
   fidapm15 with $n = 9287$ and $nnz(A) = 98519$.
3. Fluid flow modelling in a driven cavity: *e30r0500* with $n = 9661$ and $nnz(A) = 306356$ for a moderate Reynolds number of 500.
4. Structural engineering examples:
   bcsstk17 with $n = 10974$ and $nnz(A) = 219812$, and *bccstruc2* with $n = 11948$ and $nnz(A) = 80519$.

Without any scaling, SPAI preconditioner (3) cannot solve some of the above examples [21, 33, 35]. Therefore in our experiments, all these matrices are pre-processed by the Duff and Koster algorithm [28] (for minimizing the diagonal products) before constructing the preconditioners. We shall take $tol = 0.2$ in solving (7) specifying NNZ individually later in order to be as consistent with the literature as possible and the precise ratio ('NNZ ratio') between the number of nonzeros in M_1 and $nnz(A)$ will be displayed for each case. The selected two Krylov subspace solvers are the conjugate gradients squared (CGS) method by Sonneveld and the bi-conjugate gradients stabilized (BiCGStab) method by van de Vorst [49]. A third solver, the restarted generalized minimal residual method GMRES(k) [45], is also used and the convergence behaviour is found similar to CGS and BiCGStab; the results of all three solvers are shown for Set 2 examples.

[1] All such benchmark data are publicly available from Matrix Market http://math.nist.gov/MartixMarket/. Most examples in Set 1 have been attempted before in [35, 33, 21, 22] and were considered hard examples to precondition. Even with scaling by algorithm from [28], these problems are still hard for SPAI methods; see [7].

Table 1: Test of robustness with Set 1 examples using the CGS solver. Here the symbol $\surd$ indicates that the new methods are worthwhile, the symbol $\bigcirc$ shows that all methods are not very efficient to deal with a hard example, and $\bullet$ means that the standard method is already sufficient. Note the "NNZ ratio" for PSM is twice as large.

Test	Size	PSM		STD		New I		New II		NNZ	Final
Problem	n	Steps	CPU	Steps	CPU	Steps	CPU	Steps	CPU	ratio	Assess
nnc261	261	56	180	25	192	38	168	53	135	16.5	$\bigcirc$
nnc666	666	268	42	91	43	314	20	276	35	3.0	$\bigcirc$
pores3	532	59	29	88	32	54	6	43	15	3.9	$\bullet$
sherman3	5005	175	414	116	229	174	190	140	294	4.2	$\bullet$
lshp1009	1009	77	30	34	19	37	8	18	21	2.4	$\bullet$
lshp2614	2614	51	115	226	105	60	48	28	133	2.9	$\bullet$
gemat11	4929	97	643	*	*	107	345	65	611	0.5	$\surd$
fidap023	1409	43	2166	83	1361	43	1195	28	2909	5.0	$\surd$
fidapm33	2353	*	*	*	*	*	*	19	357	6.7	$\surd$

Table 2: Test of robustness with Set 1 examples using the BiCGStab solver. Here the symbol $\surd$ indicates that the new methods are worthwhile, the symbol $\bigcirc$ shows that all methods are not very efficient to deal with a hard example, and $\bullet$ means that the standard method is already sufficient. Note the "NNZ ratio" for PSM is twice as large.

Test	Size	PSM		STD		New I		New II		NNZ	Final
Problem	n	Steps	CPU	Steps	CPU	Steps	CPU	Steps	CPU	ratio	Assess
nnc261	261	27	120	21	192	27	128	22	135	16.5	$\bigcirc$
nnc666	666	261	41	71	43	248	20	244	33	3.0	$\bigcirc$
gre1107	1107	374	42862	*	*	*	*	258	34275	175	$\bigcirc$
pores3	532	39	29	59	32	42	6	38	15	3.9	$\bullet$
sherman3	5005	104	410	93	229	119	188	95	293	4.2	$\bullet$
lshp1009	1009	76	30	28	19	33	8	16	21	2.4	$\bullet$
lshp2614	2614	42	115	*	*	48	48	25	133	2.9	$\surd$
gemat11	4929	88	644	*	*	90	346	52	611	0.5	$\surd$
fidap023	1409	39	2166	71	1361	39	1195	25	2910	5.0	$\surd$
fidapm33	2353	125	399	*	*	152	251	17	357	6.7	$\surd$

Table 3: Test of improvements with Set 2 examples using three iterative methods. Here GM denotes the GMRES(20) method and BCG the BiCGStab method. The symbol $\surd$ indicates that the New method is worthwhile and $\bullet$ means that the standard method is already sufficient. Size k defines the level two preconditioner M_2 and "Flop+" shows the percentage increase in flops in computing M_2.

Test	Size	STD			New I					NNZ	Final
Problem	n	GM	CGS	BCG	GM	CGS	BCG	Size k	Flop+	ratio	Assess
s2rmt3m1	5489	98	30	27	14	17	6	543	3%	1.5	$\surd$
fidapm15	9287	90	54	38	59	33	25	3154	22%	1.0	$\surd$
e30r0500	9661	*	*	95	140	60	48	2167	15%	4.2	$\surd$
bcsstk17	10974	19	12	9	11	5	4	448	0.06%	3.0	$\surd$
bccstruc2	11948	6	3	2	6	3	2	5	0.0007%	4.5	$\bullet$

In Tables 1 and 2, we have compared the performance of PSM (Chow method [21]) and STD (a standard SPAI implementation as in [35]) with our new two-level preconditioners New I and New II. Here "Steps" refer to the number of iterations of an iterative method required to reduce the relative residual to below 10^{-8} and "CPU" the total cpu seconds taken from Sun Sparc-2 workstation running Matlab 5.3 where "*" indicates no convergence within 500 steps (in other experiments we also observed similar results from using Fortran). We display by 'NNZ ratio' the number of nonzeros allowed for M_1 (for STD and New I); the number of nonzeros for M for PSM is allowed to be larger (up to twice as large) in order for 'Steps' to be comparable. The overall NNZ for New II is only slightly more than New I as preconditioner M_2 is not contributing much to the NNZ and the CPU increase.

We have used three symbols in Tables 1–3 to remark on the final assessment of a test case. As commented in the Tables, the symbol $\surd$ indicates the cases where our new methods are particularly useful i.e. when the inaccuracy in $AM_1 = I$ is reflected more predominantly in a small number of columns.

From Tables 1-2, overall, the method New I (when it converges) is the fastest while New II is the most robust as it has less failures. One also observes that the number of convergence steps is mostly (but not always) in proportion to CPU timings. However it is still important for the new methods to achieve fast convergence and robustness. We shall compare flops counts for larger systems next.

In Table 3, we show the numerical results from three solvers: GMRES(20), CGS and BiCGstab for Set 2 examples. We here demonstrate that the proposed two-level preconditioner (New I) is effective even for larger systems. Notice that 'NNZ ratio' is not large in order to save storage and size k (that defines the level two preconditioner M_2 in (13)) is not large; consequently there is only marginal increase in flop counts and the speedup is then reflected in smaller "Steps" in Table 3. One can observe from Table 3 that the new two-level preconditioner has improved on the standard SPAI algorithm (STD), although STD (with the help of scaling by [28]) is already adequate for some examples (e.g. example *bccstruc2*).

As remarked several times, scaling is important for using SPAI methods. In other experiments, we have tried to skip the pre-processing step using the Duff and Koster algorithm [28] and found that none of the Set 1 problems can be solved effectively by the methods shown above. The main reason is that most columns in M_1 cannot approximate those of A^{-1} well (this is the case when global thresholding cannot identify an effective pattern or large elements of A^{-1}). This observation is consistent with other findings [21, 7, 33], suggesting that suitable scaling is essential for approximate inverse preconditioners. Of course, without preconditioning, all examples presented cannot be solved by Krylov subspace solvers. Combined with sparsification ideas [11, 12], our preconditioner is applicable to dense matrix problems. Another possibility would be to combine with deflation techniques such as [13] or to develop two level

methods for FSAI type preconditioners (6). Further work in these directions is under way.

6 Conclusions

We have presented a new two-level preconditioner of the approximate inverse type for accelerating iterative solution of large linear systems. We connected a sparse approximate inverse with a Gauss-Jordan decomposition and suggested a secondary preconditioner. While our new preconditioning strategy has been shown to have much improved existing preconditioners for a class of problems where SPAI ineffectiveness is due to a small number of columns in AM_1, there is much scope to develop better methods for other types of problems. We hope the new insight gained here may assist further research work on the topic. One avenue could be even better scaling methods and another idea may be a combination with compression techniques (e.g. wavelets [25, 14]) to force a matrix that has a dense inverse to possess a desirable sparse approximate inverse, before constructing such sparse preconditioners.

Acknowledgements

The authors are grateful to the anonymous referees who gave insightful and helpful comments on the early drafts of the paper. This work was partially supported by UK EPSRC grant GR/R22315. The second author wishes to thank the UK EPSRC and DERA (Winfrith, UK) for the support of a CASE studentship.

Appendix 1 – The Gauss-Jordan decomposition and sparse approximation

The SPAI preconditioner M in (3) approximates A^{-1} well if the underlying algorithm is successful in achieving $\|Am_j - e_j\|_2 \leq \epsilon$. In general, the preconditioner M satisfies

$$AM = I + E = F \tag{16}$$

where F is approximately a special matrix with E having mostly zero columns except some dense ones. This matrix F resembles the product of some selected factoring matrices F_j from the Gauss-Jordan decomposition of matrix A [3, 20] and [37, p.50] where

$$F_n F_{n-1} \cdots F_1 A = I \tag{17}$$

and each F_j is an identity matrix with its column j containing a dense vector. We shall call a sparse matrix that is of the form of $F = I + E$, in (16), an elementary Gauss-Jordan matrix of order k if E has k dense columns.

One can verify that the inverse of an elementary Gauss-Jordan matrix retains the sparsity pattern of the original matrix. Moreover the essential work in computing the exact inverse of an elementary Gauss-

Jordan matrix of order k is determined by the inverse of a $k \times k$ submatrix. We remark that, although the Gauss-Jordan method is well known [30, 37], the Gauss-Jordan matrix decomposition implicitly defined is less commonly used. It is not difficult to see that any matrix A can be decomposed into a product of elementary Gauss-Jordan matrices whose orders can be summed up to n (assuming no partial pivoting is needed or some pre-permutation has been carried out). For instance, a simple form follows from rewriting (17) as $(F_j \dots F_1)A(F_n \dots F_{j+1}) = I$ or $A = (F_n^{-1} \dots F_{j+1}^{-1})(F_j^{-1} \dots F_1^{-1})$, where each bracket defines an elementary Gauss-Jordan matrix.

We give a simple example for $n = 6$ to demonstrate the partial decomposition of a matrix A using an order 4 elementary Gauss-Jordan matrix M_2 (to yield an order 2 matrix M_1^{-1}):

$$M_2 A = \begin{bmatrix} 0.230 & -0.174 & 0.126 & -0.082 & 0 & 0 \\ -0.103 & 0.284 & -0.207 & 0.043 & 0 & 0 \\ -0.058 & -0.050 & 0.218 & -0.073 & 0 & 0 \\ -0.069 & 0.065 & -0.138 & 0.237 & 0 & 0 \\ -0.517 & 0.297 & -0.035 & -0.909 & 1 & 0 \\ -0.058 & -0.925 & 0.218 & 0.052 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 & 4 & 1 & 2 & 5 & 4 \\ 5 & 7 & 5 & 2 & 6 & 3 \\ 4 & 3 & 7 & 3 & 5 & 6 \\ 3 & 1 & 3 & 6 & 5 & 2 \\ 5 & 1 & 2 & 6 & 4 & 2 \\ 4 & 6 & 3 & 1 & 5 & 5 \end{bmatrix} = M_1^{-1} =$$

$$\left[\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 0.329 & 0.993 \\ 0 & 1 & 0 & 0 & 0.371 & -0.716 \\ 0 & 0 & 1 & 0 & 0.136 & 0.783 \\ 0 & 0 & 0 & 1 & 0.539 & -0.435 \\ \hline 0 & 0 & 0 & 0 & -1.520 & -1.200 \\ 0 & 0 & 0 & 0 & 0.511 & 3.410 \end{array}\right] = \left[\begin{array}{cccc|cc} 1 & 0 & 0 & 0 & 0.134 & -0.244 \\ 0 & 1 & 0 & 0 & 0.356 & 0.336 \\ 0 & 0 & 1 & 0 & 0.014 & -0.225 \\ 0 & 0 & 0 & 1 & 0.450 & 0.287 \\ \hline 0 & 0 & 0 & 0 & -0.746 & -0.263 \\ 0 & 0 & 0 & 0 & 0.112 & 0.333 \end{array}\right]^{-1}.$$

In the notation of (14), using $P_1 = [e_5\ e_6\ e_1\ e_2\ e_3\ e_4]$ would permute the above matrix to a lower block triangular form. Here one observes that $M_2 A M_1 = I$ and the last matrix and its inverse are related through the smaller submatrix

$$\begin{bmatrix} -1.520 & -1.200 \\ 0.511 & 3.410 \end{bmatrix} = \begin{bmatrix} -0.746 & -0.263 \\ 0.112 & 0.333 \end{bmatrix}^{-1}.$$

This observation prompts us to consider situations where matrix AM_1 (or $M_2 A$) is approximately an elementary Gauss-Jordan matrix. If this is the case, we may naturally employ another elementary Gauss-Jordan matrix M_2 (or M_1) to achieve $M_2 A M_1 \approx I$. We shall use this idea to propose a two-level preconditioner based on preconditioner M from (3) for (1).

Appendix 2 – Proof of a smaller upper bound

Here we show that our new method (13) defines a better preconditioner than the standard SPAI method (12) for the simple case of $\epsilon \le \|E_1\|_F < 1$ and $\|E_2\|_F < \epsilon < 1$ with $\|E_1\|_F + \|E_2\|_F < 1$.

First of all, the standard SPAI preconditioner (12) satisfies

$$\|AM_1 - I\|_F = \|E_1 + E_2\|_F \le \|E_1\|_F + \|E_2\|_F.$$

Now from $\|E_1\|_F < 1$, we have $\|(I+E_1)^{-1}\|_F \leq 1/(1-\|E_1\|_F)$. Then our new preconditioner (13) satisfies

$$\|M_2AM_1 - I\|_F = \|(I+E_1)^{-1}E_2\|_F \leq \frac{\|E_2\|_F}{1-\|E_1\|_F}.$$

As $\|E_1\|_F + \|E_2\|_F < 1$, from

$$\frac{\|E_2\|_F}{1-\|E_1\|_F} - \Big(\|E_1\|_F + \|E_2\|_F\Big) = \|E_1\|_F \frac{\|E_1\|_F + \|E_2\|_F - 1}{1-\|E_1\|_F} \;<\; 0,$$

One sees that our new method (13) defines a better preconditioner than the standard SPAI method (12).

For instance, if $\|E_1\|_F = 0.9, \quad \|E_2\|_F = 0.05$, then

$$\|AM_1 - I\|_F \leq \|E_1\|_F + \|E_2\|_F = 0.95, \qquad \|M_2AM_1 - I\|_F \leq \frac{\|E_2\|_F}{1-\|E_1\|_F} = 0.026.$$

References

[1] Axelsson O., *Iterative solution methods*, Cambridge University Press, UK, 1994.

[2] Barrett R., Berry M., Chan T. F., Demmel J., Donato J., Dongarra J., Eijkhout V., Pozo R., Romine C., Van der Vorst H., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd Edition, SIAM, Philadelphia, PA, USA, 1994.

[3] Barulli M. ad Evans D. J., *Implicit Gauss-Jordan scheme for the solution of linear systems*, Para. Algor. Applics., 10 (1996), pp.145-159.

[4] Benson M. W. and Frederickson P. O., *Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems*, Utilitas Math., 22 (1982), pp.127-140.

[5] Benzi M. and Tuma M., *A comparable study of sparse approximate inverse preconditioners*, Los Alamos Nat. Lab., LA-UR-98-0024, USA; http://www.uivt.cas.cz/~tuma/ps/final.ps.gz.

[6] Benzi M. and Tuma M., *A sparse approximate inverse preconditioner for nonsymmetric linear systems* SIAM J. Sci. Comput., 19 (1998), pp.968-994.

[7] Benzi M., Haws J. C. and Tuma M., *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp.1333-1353.

[8] Bjorck A., *Numerical methods for least squares methods*, SIAM Publications, 1996.

[9] Bridson R. and Tang W. P., *Multiresolution approximate inverse preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp.463-479.

[10] Bollhöfer M. and Saad Y., *Multilevel preconditioners constructed from inverse–based ILUs*, University of Minnesota, (2004), UMSI report 2004/75.

[11] Carpentieri B., Duff I. S. and Giraud L., *Experiments with sparse preconditioners of dense problems from electromagnetic applications*, CERFACS TR/PA/00/04, France; http://www.cerfacs.fr/algor/reports/2000/TR_PA_00_04.ps.gz.

[12] Carpentieri B., Duff I. S. and Giraud L., *Some sparse pattern selection strategies for robust Frobenius norm minimization preconditioners in electromagnetism*, Numer. Lin. Alg. Appl., 7 (2000), pp.667-685. (See also RAL-TR-2000-009, http://www.numerical.rl.ac.uk/reports/)

[13] Carpentieri B., Duff I. S. and Giraud L., *A class of spectral two-level preconditioners*, SIAM J. Sci. Comput., 25 (2003); also CERFACS TR/PA/02/55, France. http://www.cerfacs.fr/algor/reports/2002/TR_PA_02_55.ps.gz.

[14] Chan T. F. and Chen K., *On two variants of an algebraic wavelet preconditioner*, SIAM J. Sci. Comput., 24 (2002), pp.260-283; see also UCLA CAM report 02-02.

[15] Chan T. F., Tang W. P. and Wan W. L., *Wavelet sparse approximate inverse preconditioners*, BIT, 37 (1997), pp.644-660.

[16] Chen K., *On a class of preconditioning methods for dense linear systems from boundary elements*, SIAM J. Sci. Comput., 20 (1998), pp. 684-698.

[17] Chen K., *An analysis of sparse approximate inverse preconditioners for boundary integral equations*, SIAM J. Matr. Anal. Appl., 22 (2001), pp.1058-1078.

[18] Chen K., *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, UK, 2005.

[19] Chen K. and Harris P. J., *Iterative methods for solving the boundary element equations for the three dimensional Helmholtz equation*, Appl. Numer. Math., 36 (2001), pp.475-489.

[20] Chen K. and Lai C. H., *Parallel algorithms of the Purcell method for direct solution of linear systems*, Para. Comput., 28 (2002), pp.1275-1291.

[21] Chow E., *A priori sparsity patterns for parallel sparse inverse preconditioners*, SIAM J. Sci. Compt., 21 (2000), pp.1804-1822.

[22] Chow E., *Parallel implementation and practical use of sparse approximate inverse preconditioners with a priori sparsity patterns*, Int. J. High Perf. Comput. Appl., 15 (2001), 56-74.

[23] Chow E. and Saad Y., *Parallel Approximate Inverse Preconditioners*, Proc. 8th SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN, (1997), pp.14-17 (http://www.llnl.gov/CASC/people/chow/pubs/history.ps).

[24] Chow E. and Saad Y., *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp.995-1023.

[25] Cohen A. and Masson R., *Wavelet methods for second-order elliptic problems, preconditioning, and adaptivity*, SIAM. J. Sci. Comput., 21 (1999), pp.1006-1026.

[26] Cosgrove J. D. F., Diaz J. C. and Griewank A., *Approximate inverse preconditioners for sparse linear systems*, Int. J. Compt. Math., 44 (1992), pp.91-110.

[27] Demko S., Moss W. F. and Smith P. W., *Decay rates for inverse of band matrices*, Math. Comp., 43 (1984), pp.491-499.

[28] Duff I. S. and Koster J., *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matr. Anal. Appl., 20 (1999), pp.889-901. (also RAL-TR-1999-030, http://www.numerical.rl.ac.uk/reports/)

[29] Embree M., *How descriptive are GMRES convergence bounds?*, University of Oxford Computing Lab., NA Report 99/08 (1999);
http://web.comlab.ox.ac.uk/oucl/work/mark.embree/estimates.ps.gz.

[30] Fox L., *An Introduction to Numerical Linear Algebra*, Oxford University Press, UK, 1964.

[31] Freund R. W., Golub G. H. and Nachtigal N. M., *Iterative solution of linear systems*, Acta Numerica, Cambridge University Press, UK (1992), pp. 57-100.

[32] Golub G. H. and Van Loan C., *Matrix Computation*, 2nd ed, John Hopkins University Press, USA, 1989.

[33] Gould, N. I. M. and Scott, J. A., *Sparse Approximate-Inverse Preconditioners Using Norm-Minimization Techniques*, SIAM J. Sci. Comput., 19 (1998), pp.605-625. (also RAL-TR-1995-026, http://www.numerical.rl.ac.uk/reports/)

[34] Grosz L., *Preconditioning by incomplete block elimination*, Numer. Lin. Alg. Appl., 7 (2000), pp.527-541.

[35] Grote M. and Huckle T., *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp.838-853.

[36] Huckle T., *Approximate sparsity patterns for the inverse of a matrix and preconditioning*, Appl. Numer. Math., 30 (1999), pp.291-303.

[37] Isaacson E. and Keller H. B., *Analysis of Numerical Methods*, John Wiley, 1966.

[38] Kaporin I. E., *Two-level explicit preconditioning of the conjugate gradient method*, Diff. Eqn., 28 (1992), pp.329-339.

[39] Kolotilina L. Y. and Yeremin A. Y., *On a family of two-level preconditionings of the incomplete block factorization type*, Soviet J. Numer. Anal. Math. Model., 1 (1986), pp.293-320.

[40] Kolotilina L. Y. and Yeremin A. Y., *Factorized sparse approximate inverse preconditioners I - Theory*, SIAM J. Matr. Anal. Appl., 14 (1993), pp.45-58.

[41] Kolotilina L. Y., Nikishin A. A. and Yeremin A. Y., *Factorized sparse approximate inverse preconditioners IV: simple approaches to rising efficiency*, Numer. Lin. Alg. Applcs., 6 (1999), pp.515-531.

[42] Koschinski C., *Properties of approximate inverse and adaptive control concepts for preconditioning*, Internal Report 74/99, PhD thesis, University of Karlsrule, Germany, (1999); www.ubka.uni-karlsruhe.de/indexer-vvv/1999/mathematik/4.

[43] Nachtigal N. M., Reddy S. and Trefethen N., *How fast are nonsymmetric matrix iterations?*, SIAM J. Matr. Anal. Appl., 13 (1992), pp.778–795.

[44] Saad Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.

[45] Saad Y. and Schultz M. H., *GMRES: a generalized minimal residual algorithm for solving unsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp.856-869.

[46] Saad Y. and van der Vorst H. A., *Iterative solution of linear systems in the 20th century*, J. Comput. Appl. Math., 123 (2000), pp.1-33.

[47] Tang W. P., *Towards an effective sparse approximate inverse preconditioner*, SIAM J. Matr. Anal. Appl., 20 (1999), pp.970-986.

[48] Tang W. P. and Wan W. L., *Sparse approximate inverse smoother for multigrid*, SIAM J. Matr. Anal. Appl., 21 (2000), pp.1236-1252.

[49] van der Vorst H. A., *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp.631-644.

[50] Vavasis S., *Preconditioning for boundary integral equations*, SIAM J. Matr. Anal. Appl., 13 (1992), 905-925.

[51] Zollenkopf K., *Bi-factorisation – basic computational algorithm and programming technique*, in: Large Sparse Sets of Linear Equations, ed. J K Reid, Academic Press, 1971, pp.75-97.