Parallel approximate inverse preconditioners*

Edmond Chow[†] Yousef Saad[†]

Abstract

There has been much excitement recently over the use of approximate inverses for parallel preconditioning. The preconditioning operation is simply a matrix-vector product, and in the most popular formulations, the construction of the approximate inverse seems embarassingly parallel. However, difficulties arise in practical parallel implementations. This paper will survey approximate inverse preconditioners, and discuss the wide variety of options, such as for sparsity pattern selection. We address the pros and cons of each method, and put the methods into perspective.

1 Introduction

Direct approximations to the inverse of a matrix have been developed in the past decade for preconditioning iterative methods. Their main advantage in the context of parallelism is that the preconditioning operation is a matrix-vector product. They are also essential in block incomplete factorization preconditioners, where large, sparse pivot blocks need to be inverted, possibly in parallel. A requirement here is that the product of the approximate inverse with another sparse block is cheap and remains sparse.

This paper considers parallel methods of approximating the inverse of a sparse matrix by minimizing a Frobenius norm. Section 2 reviews these methods and highlights their similarities and differences. The serial cost of constructing approximate inverses is usually high, higher than computing an incomplete factorization, for example, and thus parallelization must be considered. Issues related to parallelization are briefly discussed in Section 3. The main methods are summarized in Section 4.

2 Review of parallel approximate inverse preconditioners

We consider methods of approximating the matrix A^{-1} with a sparse matrix M by minimizing the Frobenius norm of a residual matrix such as $(I - AM)$. An important feature of this objective function is that it can be decoupled as the sum of squares of the 2-norms of the individual columns

$$\|I - AM\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_j\|_2^2 \tag{1}$$

in which e_j and m_j are the j-th columns of the identity matrix and of the matrix M, respectively. Thus, minimizing (1) is equivalent to minimizing the individual functions

$$\|e_j - Am_j\|_2, \quad j = 1, 2, \ldots, n. \tag{2}$$

*This work was supported in part by the National Science Foundation under grant NSF/CCR-9214116 and in part by NASA under grant NAG2-904.

[†]Department of Computer Science and Minnesota Supercomputer Institute, University of Minnesota, 4-192 EE/CSci Bldg., 200 Union St., S.E., Minneapolis, Minnesota, 55455-0154 (chow@cs.umn.edu and saad@cs.umn.edu).

Of course, if no restriction is placed on M, the exact inverse will be found. Thus a sparsity pattern (set of nonzero indices) $\mathcal{S}$ for M is prescribed, and (2) can be solved as a set of least-squares problems. If A is nonsingular, then the least-squares matrices have full rank.

If the normal equations are used to solve (1), then we need to solve the linear system

$$(A^T AM)_{ij} = A^T_{ij}, \quad (i,j) \in \mathcal{S}. \tag{3}$$

Explicitly, the independent linear system that needs to be solved for column j is

$$(A^T Am_j)_i = (A^T e_j)_i, \quad (i,j) \in \mathcal{S}.$$

Note that the objective function (1) will lead to a *right* approximate inverse. The function $\|I - MA\|_F = \|I - A^T M^T\|_F$ will give a *left* approximate inverse and is treated in exactly the same way. To be consistent in the following, we will use the form (1), which will be notationally clearer. The early developments of approximate inverses, however, used the form for the left approximate inverse.

The major developments are treated in order in the following sections. We begin with the least-squares and diagonal block method first defined by Benson [5]. Kolotilina and Yeremin [25] generalized the methods with weighted Frobenius norms. They also developed factorized approximate inverses [26] which guarantee the nonsingularity of the approximate inverse. Techniques to update the sparsity pattern $\mathcal{S}$ were developed by Cosgrove, Díaz and Griewank [15]. Later, this was shown to be effective by Grote and Huckle [20] and their paper popularized the method. At the same time, Chow and Saad [11] developed cheaper methods using approximate minimization for the least-squares subproblems which also have the benefit of finding sparsity patterns automatically.

2.1 Least-squares and diagonal block methods

The first approximations to the inverse using a norm of $(I - AM)$ were by Benson [5], in his M.S. thesis work with Frederickson. Techniques specific to circulant matrices were developed, and two techniques were extended to approximating the inverse of banded matrices A with banded matrices M of semibandwidth $2q + 1$.

The *least-squares* method determines each column m_j independently by minimizing $\|e_j - Am_j\|_2$ via solving normal equations of size $2q + 1$. The *diagonal block* method constructs M such that the $2q + 1$ central diagonals of $I - AM$ are zero. For each column m_j, independent linear systems involving diagonal blocks of A of size $2q + 1$ are solved.

In general terms, the diagonal block method seeks M such that

$$(AM)_{ij} = \delta_{ij}, \quad (i,j) \in \mathcal{S} \tag{4}$$

where δ_{ij} is the Kronecker delta function. The columns can be computed independently, as shown explicitly by

$$(Am_j)_i = (e_j)_i, \quad (i,j) \in \mathcal{S}, \quad j = 1, 2, \ldots, n.$$

The systems are consistent if the principal submatrices of A are nonsingular. Kolotilina [23] has shown that M computed this way is nonsingular if A is an H-matrix.

Another related method is the *truncation* method,

$$M_{ij} = \{A^{-1}\}_{ij}, \quad (i,j) \in \mathcal{S}, \tag{5}$$

i.e., the nonzero entries in M are exactly the corresponding entries in A^{-1}. For matrices with a general structure, this type of approximate inverse can be computed given an LU factorization of A [30, 2]. This type of approximate inverse was developed for incomplete block factorizations when the pivot blocks are banded.

In general, the least-squares and diagonal block methods seek an approximate inverse with a predetermined sparsity pattern, $\mathcal{S}$, with nnz nonzeros, and whose entries are determined through nnz equations. In the diagonal block method, each equation arises from controlling nnz entries in the product AM, in particular, the entries specified by $\mathcal{S}$, and ignoring the other entries (see Equation (4)). In the least-squares method, each equation is from the normal equations for a minimization based on all the entries in AM.

Benson's work was not published more widely until Benson and Frederickson [6] generalized the sparsity pattern $\mathcal{S}$ to *q-local operators*. In a grid consisting of nodes connected by edges, column i of the matrix of a q-local operator consists of nonzeros corresponding to node i and its q-th level nearest-neighbors. The matrix of a 0-local operator is a diagonal matrix. The graph of a symmetric matrix can be considered a 1-local operator; thus when $\mathcal{S}$ is the pattern of A, the 1-local approximate inverse is being sought.

The use of approximate inverses in a parallel environment was not mentioned until Benson et. al.[7]. Grote and Simon [21] implemented a least-squares method on the CM-5, where A and M were banded matrices.

Unfortunately, M determined by either the least-squares or diagonal block methods cannot be guaranteed to be nonsingular. Also, if A is symmetric, M is not necessarily symmetric. Ong [28] suggested the minimization of $\|AMA - A\|_F$, which produces a symmetric M when A is symmetric. This is true because the minimum is unique, and $\|AMA - A\|_F = \|AM^TA - A\|_F$. Other functionals that give a symmetric M are possible.

Ong also suggested a weighted diagonal block approximate inverse, by solving

$$(VAM)_{ij} = V_{ij}, \quad (i,j) \in \mathcal{S} \tag{6}$$

where V is a matrix, and noted that it gives the same result as the least-squares method when $W = A^T$ (although only symmetric A was considered).

2.2 Weighted Frobenius norms

Kolotilina and Yeremin [25] introduced the weighted Frobenius norm for the minimization

$$\min \|I - AM\|_W^2 = \min \operatorname{tr}\{(I - AM)W(I - AM)^T\}$$

where W is a positive definite, but not necessarily symmetric matrix. Minimization of this quadratic functional with respect to the entries in M yields the system of linear equations

$$\{A^T(W + W^T)AM\}_{ij} = \{A^T(W + W^T)\}_{ij}, \quad (i,j) \in \mathcal{S}$$

where $\mathcal{S}$ is again the prescribed sparsity pattern of M. If W is known explicitly, each column of M can be computed independently.

Remarkably, the least-squares, diagonal block, and truncation methods all fall under this generalization. If $W = I$, then we have the least-squares method using the normal equations. If $W = A^{-1}$ and A is symmetric, then we have the diagonal block method. If $W = A^{-T}A^{-1}$, we have the truncation method (5). Ong's weighted diagonal block method (6) was a similar generalization. Axelsson [1] has an excellent review of this work.

2.3 Factorized approximate inverses

To solve the problem of the possible singularity of the approximate inverse M, Kolotilina and Yeremin [26] consider factorized approximate inverses

$$A^{-1} \approx U_M L_M$$

where L_M and U_M are approximate inverses of L_A and U_A, the LU factors of A. When A is SPD, only one factor is computed, and if the diagonal of this factor does not contain any zeros, the resulting preconditioner is also SPD.

Of course, L_M and U_M can be found with the previous methods once the factors L_A and U_A are computed. However, Kolotilina and Yeremin noticed the remarkable fact that when minimizing (the form for the left approximate inverse is used here for notational convenience)

$$\|I - L_M L_A\|_F$$

using the normal equations, the factorization $A = L_A U_A$ does not need to be known.

Consider the symmetric case. Define $\mathcal{S}_L$ to be the sparsity pattern of the lower triangular factor, and substitute $\mathcal{S}_L$ for $\mathcal{S}$, L_M for M and L_A for A in the normal equations for the left approximate inverse

$$(MAA^T)_{ij} = A^T_{ij}, \quad (i,j) \in \mathcal{S}$$

to get

$$\{L_M L_A L_A^T\}_{ij} = \{L_A^T\}_{ij}, \quad (i,j) \in \mathcal{S}_L$$

or

$$\{L_M A\}_{ij} = \{L_A^T\}_{ij}, \quad (i,j) \in \mathcal{S}_L.$$

Since $\{L_A^T\}_{ij}$ for $(i,j) \in \mathcal{S}_L$ is a diagonal matrix, the off-diagonal entries of L_A do not need to be known.

Since the diagonal of L_A is also usually not known, Kolotilina and Yeremin solve instead

$$(\widetilde{L_M} A)_{ij} = \delta_{ij}, \quad (i,j) \in \mathcal{S}_L$$

and then scale $L_M = D^{-1}\widetilde{L_M}$ where D is the diagonal of $\widetilde{L_M}$. This produces a preconditioned matrix $L_M A L_M^T$ such that its diagonal contains all ones. The above equation looks like (4), and thus precautions such as diagonal compensation [3] must be taken to ensure L_M will not be singular.

The factorized approximate inverse can be easily extended to the nonsymmetric case by using weighted Frobenius norms or the weighted diagonal block method. The minimization of the functional $\|I - LAU\|_F$ has been suggested by Saad [29]. Factorized forms based on conjugation [8, 9] also deserve mention.

2.4 Updating the sparsity pattern

In the methods described above, the sparsity pattern $\mathcal{S}$ is chosen beforehand and is kept constant during the computation of M. Although there are strategies to select $\mathcal{S}$ based on the cost of constructing and using M [27], a good *a priori* choice based on the structure of the inverse is very difficult. A wavelet basis has been proposed to make the approximation better for PDE problems [10].

For the least-squares method, Cosgrove, Díaz and Griewank [15] introduced the idea of augmenting $\mathcal{S}$ while M is being constructed. An initial approximate inverse is constructed with an initial sparsity pattern $\mathcal{S}$. Then, new nonzero locations are introduced into $\mathcal{S}$, and the new minimization problem is solved. This process is repeated until a maximum number of nonzeros has been reached, or $\|I - AM\|_1$ satisfies a lower bound. The 1-norm was used since it can be checked column-by-column.

A numerical test is used to determine which nonzero locations should be added to $\mathcal{S}$. A second issue is which locations to check, and in what order, since it is expensive to check all of them.

For the j-th column, the numerical test is based on the reduction in the optimal residual norm, $\min \|e_j - Am\|_2$ when the sparsity pattern of m is augmented. This reduction can be determined without computing the corresponding new solution. Denote $\mathcal{M}$ as the set of vectors with a given sparsity pattern, and $\mathcal{M}^*$ as the set of vectors with the same pattern plus an additional nonzero at position k. Also define

$$\begin{aligned} m_j &= \arg \min_{m \in \mathcal{M}} \|e_j - Am\|_2, & r &= e_j - Am_j \\ m_j^* &= \arg \min_{m \in \mathcal{M}^*} \|e_j - Am\|_2, & r^* &= e_j - Am_j^*. \end{aligned}$$

The best improvement to the residual norm for column j can be measured by

$$\|r^*\|_2^2 - \|r\|_2^2 = -\frac{(r^T A e_k)^2}{\|PAe_k\|_2^2}$$

where P is the appropriate orthogonal projector, and Ae_k is the k-th column of A.

Since PAe_k is rather expensive to compute for many k's, Cosgrove, Díaz and Griewank [15] use the lower bound

$$\frac{|r^T A e_k|}{\|PAe_k\|_2} \geq \frac{|r^T A e_k|}{\|Ae_k\|_2} \tag{7}$$

and consider locations k whose lower bound is above a certain tolerance.

Instead of finding the exact minimizer m_j^* for the augmented problem, Grote and Huckle [20] hold the current solution and find the scalar μ that gives

$$\min \|r - \mu A e_k\|_2.$$

The solution is simply $\mu = -r^T A e_k / \|Ae_j\|_2^2$, and the new residual r^* satisfies

$$\|r^*\|_2^2 - \|r\|_2^2 = -\frac{(r^T A e_k)^2}{\|Ae_k\|_2^2}.$$

The right-hand side is used to test new locations. By comparing this with the lower bound in (7), we see that the approximation made by Cosgrove, Díaz and Griewank [15] gives the same numerical test.

In later work, Gould and Scott [18] found a relatively cheap computation of $\|PAe_k\|_2$ and use the exact test (the left-hand side of (7)).

If the least-squares problems are solved via a QR factorization, then the solution m_j^* to the augmented least-squares problem should be found via updating the QR factorization. Huckle [22] discusses many possible ways of solving the least-squares problem that take advantage of the sparse structure of the least-squares matrix.

Notice that for most locations k, there will be no improvement to the residual norm, specifically, when the nonzero locations of r and Ae_k do not intersect (r is sparse since A

and m_k are sparse). Grote and Huckle [20] determine which columns in A intersect with r to make the process more efficient. The best s locations (e.g., $s \leq 5$) are added to the pattern of the j-th column of $\mathcal{S}$.

Although their work was earlier, Cosgrove, Díaz and Griewank [15] used a more complicated scheme that depends on the order in which locations are checked. Locations that introduce fewer new rows h in the least-squares subproblems are checked first. For each h, the set of all possible candidates is divided into three prioritized classes. The first location that satisfies their numerical test is added. See [15] for details.

It is possible to interpret the candidate locations as nodes on a grid when A is the result of a discretization. Consider approximating column j starting with a single entry at location j. The residual r corresponds to the first-level neighbors of node j, and the candidate locations consist of these plus the second-level neighbors. When candidates are actually added, the new residual contains nonzeros corresponding to the neighbors of these added candidates. The pattern of r grows from a point on the grid.

There is no simple way to update the sparsity pattern if A is not available explicitly, as in the case of factorized approximate inverses, where L_A and U_A are not known. It is also difficult to update the sparsity pattern in the diagonal block methods.

2.5 Approximate minimization and automatic sparsity pattern

Since the cost of solving the individual normal equations can be quite high, particularly when $\mathcal{S}$ is relatively full, Kolotilina, Nikishin, and Yeremin [24] proposed solving it approximately with a few steps of the conjugate gradient method.

Chow and Saad [11] approximately minimize (2) directly with a few steps of an iterative method, such as minimal residual, starting with a sparse initial guess. In general, we can consider any iterative method that computes an approximate solution to

$$Am_j = e_j, \quad j = 1, 2, \ldots, n. \tag{8}$$

Suppose a zero initial guess is used. The first few iterates are sparse because $e_j, Ae_j, A^2e_j, \cdots$ are sparse. For this method to be efficient, sparse-sparse operations must be used. The product of a sparse matrix by a sparse vector with p nonzeros only involves p columns of the sparse matrix. An initial sparsity pattern $\mathcal{S}$ does not need to be prescribed, as the pattern of M fills in automatically. To maintain sparsity, a dropping strategy is used.

Consider an update to the solution

$$m^* = m + \alpha d$$

where d is the search direction, and α is the step length that minimizes the new residual. After this update, m^* will have the pattern of the union of the patterns of m and d. To maintain sparsity, there are two choices: drop small elements in m^*, or drop small elements in d before the update is made, i.e.,

$$\begin{aligned} d' &= d - f \\ m^* &= m + \alpha d' \end{aligned}$$

where f contains the small elements of d. In the former case, the minimal residual property is lost, and so it is not recommended. In the latter case, the pattern of d' can be selected to control the pattern of m^*, by keeping only the largest elements in d' not in the pattern of m.

If d is the residual direction r, those nonzero positions in r not in m can be considered the candidate locations, or locations for possible fill-in. The new residual $r^* = e_j - Am^*$ contains the new candidate positions, the neighbors of r on the grid. The pattern of m^* grows from a point on the grid, like the methods in the previous section.

We can also choose $d = A^T r$, the normal equations search direction. If A is additionally scaled so that the norm of each column is 1, then the largest elements in d are also those that correspond to the largest decrease in the residual norm in some sense. This is identical to the test (7). To see this, note that the k-th entry in $d = A^T r$ can be written $r^T A e_k$.

The approximate solution of the linear subproblems (8) using an iterative method suffers from the same problems as solving the original problem if A is indefinite or poorly conditioned. However, the linear systems may be preconditioned with the columns that have already been computed. More precisely, each system (8) for approximating column j may be preconditioned with M' where the first $j-1$ columns of M' are the m_k that already have been computed, $1 \leq k < j$, and the remaining columns are the initial guesses for the m_k, $j \leq k \leq n$. This is called *self-preconditioning.*

2.6 Applications of the approximate inverse

The approximate inverse may be used as an explicit preconditioner for the entire matrix, or used to precondition matrices partitioned as

$$A = \begin{pmatrix} B & F \\ E & C \end{pmatrix}.$$

If it is easy to solve with B, then one solution method is to solve the Schur complement system with the matrix $S = C - EB^{-1}F$ with an iterative method and preconditioner. Kolotilina and Yeremin [27] choose B to be diagonal and construct a scaled version of S to which they apply their factorized approximate inverse.

If it is difficult to solve with B, then iterating with the matrix S will be expensive, since one must solve with it accurately. As an alternative, a fully-coupled type of preconditioner (i.e., involving all the variables) can be constructed, for example the factorization

$$\begin{pmatrix} B & 0 \\ E & M_S \end{pmatrix} \begin{pmatrix} I & B^{-1}F \\ 0 & I \end{pmatrix} \tag{9}$$

where M_S is an approximation to S. Approximate inverse techniques may be used for B^{-1} (or $B^{-1}F$) and M_S^{-1} when they are required. The former is essential if a sparse approximation to S is constructed. This is the approach of Grote and Huckle [19]. Tang [31] also uses the same approach, but he chooses a specific block partitioning that is effective with approximate inverses. The general framework of this approach is described in Chow and Saad [12], who also describe other block preconditioners and a variety of approximate inverse techniques to approximate S or S^{-1}.

The block factorization (9) can be viewed as the outer-product form of incomplete block factorization [2, 14, 17, 25]. An additional issue in this case is that M_S must be sparse for the next stage of the factorization.

In general, the approximate inverse technique of minimizing (2) can be considered the sparse approximate solution to a sparse linear system with a sparse right-hand side, e.g., $B^{-1}F$ used in the Schur complement above. This technique has application, for example, in constructing matrices such as preconditioners which need to be sparse [13].

3 Parallelism issues

The main advantage of approximate inverses in a parallel environment is that the preconditioning operation is a matrix-vector product, which can be parallelized relatively easily. A secondary advantage is that the construction of the approximate inverse itself can be parallelized.

Both A and M are usually partitioned by rows, since this is more efficient for the matrix-vector product. To determine the rows of M independently, the form $\|I - MA\|$ for the left approximate inverse is used. Each processor works on a set of rows. In a shared-memory approach, or if the sparse operator can be stored on each processor, then the only issue for all the methods is the possible load imbalance [18]. In the case that the matrix is only available as an operator, the operator must be sparse and the nonzero indices in the product must be available for the algorithms to work efficiently.

In a distributed memory approach, each processor will need rows from other processors to do its computation. If the sparsity pattern is chosen beforehand, then it is possible to distribute all the rows required by each processor before any computations begin. Also, if the sparsity pattern of M is chosen to be the same as that for A, then the communication pattern for multiplications with these matrices is the same. Otherwise, it is also advantageous to choose a sparsity pattern that is conducive to the network architecture, or minimizes the amount of work in some way [27].

If the sparsity pattern is updated dynamically, it is still possible to distribute the rows that might be needed to each processor. A processor needs the rows for the nodes that it owns, and the rows for the neighbors of these nodes. A partitioning that minimizes the number of cut edges is required here. In the most general case, each row that is needed is requested from the processor that owns it. For PDE problems, a combination of these two methods may be best.

Two recent implementations of [20] are [16] and [4]. The former implements the method approximately, by only considering the first-level neighbors of a node as candidate fill-in positions. The latter uses dynamic load balancing and other optimizations in its parallel implementation.

4 Summary and conclusion

The three major methods are the factorized sparse approximate inverse (FSAI) [26], the least-squares method with dynamic pattern selection (SPAI) [15, 20] and the approximate minimization method [11].

FSAI preserves symmetry and positive definiteness of the approximate inverse by solving least-squares problems involving the Cholesky factors with the normal equations. The approximate inverse is effectively denser than the factors themselves. However, there is no simple technique to dynamically update the sparsity pattern. Also, techniques such as diagonal compensation [3] are required for non-H-matrices. Factorized approximate inverses also require an extra communication step to carry out the matrix-vector multiply.

The SPAI preconditioner does not have these restrictions, but it is practically impossible to guarantee that the approximate inverse will be nonsingular in nonfactored form. It allows the dynamic update of the sparsity pattern, but is relatively expensive.

The approximate minimization method reduces the cost of the minimizations, and the sparsity pattern emerges automatically, without the need to maintain lists of candidates. However, this latter feature requires sparse-sparse operations. Self-preconditioning, if it is used, will also incur a high communication cost when parallelized.

References

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.

[2] O. Axelsson, S. Brinkkemper, and V. P. Il'in, *On some versions of incomplete block-matrix factorization iterative methods*, Lin. Alg. Appl., 58 (1984), pp. 3–15.

[3] O. Axelsson and L. Yu. Kolotilina, *Diagonally compensated reduction and related preconditioning methods*, Num. Lin. Alg. Appl., 1 (1995).

[4] S. T. Barnard and R. Clay, *A portable MPI implementation of the SPAI preconditioner in ISIS++*, in Proc. Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, Minnesota, March, 1997.

[5] M. W. Benson, *Iterative solution of large scale linear systems*, Master's thesis, Lakehead University, Thunder Bay, ON, 1973.

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

[7] M. W. Benson, J. Krettmann, and M. Wright, *Parallel algorithms for the solution of certain large sparse linear systems*, Intl. J. Comp. Math., 16 (1984), pp. 245–260.

[8] M. Benzi, C. D. Meyer, and M. Tuma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.

[9] M. Benzi and M. Tuma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., (to appear).

[10] T. F. Chan, W.-P. Tang, and W. L. Wan, *Wavelet sparse approximate inverse preconditioners*, Tech. Rep. 96-33, CAM, University of California, Los Angeles, Los Angeles, CA., 1996.

[11] E. Chow and Y. Saad, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., (to appear).

[12] ——, *Approximate inverse techniques for block-partitioned matrices*, SIAM J. Sci. Comput., (to appear).

[13] ——, *ILUS: an incomplete LU factorization for matrices in sparse skyline format*, Intl. J. Num. Meth. Fluids, (to appear).

[14] P. Concus, G. H. Golub, and G. Meurant, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 309–332.

[15] J. D. F. Cosgrove, J. C. Díaz, and A. Griewank, *Approximate inverse preconditioning for sparse linear systems*, Intl. J. Comp. Math., 44 (1992), pp. 91–110.

[16] V. Deshpande, M. J. Grote, P. Messmer, and W. Sawyer, *Parallel implementation of a sparse approximate inverse preconditioner*, in Parallel Algorithms for Irregularly Structured Problems (Proc. IRREGULAR '96), A. Ferreira, J. Rolim, Y. Saad, and T. Yang, eds., Santa Barbara, California, 1996, pp. 63–74.

[17] J. C. Díaz and C. G. Macedo, Jr., *Fully vectorizable block preconditionings with approximate inverses for non-symmetric systems of equations*, Intl. J. Num. Meth. Engrg., 27 (1989), pp. 501–522.

[18] N. I. M. Gould and J. A. Scott, *On approximate-inverse preconditioners*, Tech. Rep. RAL-95-026, Rutherford Appleton Laboratory, Oxfordshire, England, 1995.

[19] M. Grote and T. Huckle, *Effective parallel preconditioning with sparse approximate inverses*, in Proc. Seventh SIAM Conference on Parallel Processing for Scientific Computing, San Francisco, California, 1994, pp. 466–471.

[20] ——, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., (to appear).

[21] M. Grote and H. D. Simon, *Parallel preconditioning and approximate inverses on the Connection Machine*, in Parallel Processing for Scientific Computing, R. F. Sincovec, D. E. Keyes, L. R. Petzold, and D. A. Reed, eds., SIAM, Philadelphia, PA, 1993, pp. 519–523. Vol. 2.

[22] T. Huckle, *Efficient computation of sparse approximate inverses*, Tech. Rep. 342/04/96, Technische Universität München, München, Germany, 1996.

[23] L. Yu. Kolotilina, *Explicit preconditioning of H-matrices*, in Numerical analysis and mathematical modeling, 1989, pp. 97–108.

[24] L. Yu. Kolotilina, A. A. Nikishin, and A. Yu. Yeremin, *Factorized sparse approximate inverse (FSAI) preconditionings for solving 3D FE systems on massively parallel computers. II. Iterative construction of FSAI preconditioners*, in Proc. IMACS Intl. Symp. Iterative Methods in Linear Algebra, R. Beauwens and P. de Groen, eds., Brussels, Belgium, 1992, pp. 311–312.

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

[26] ——, *Factorized sparse approximate inverse preconditionings I. Theory*, SIAM J. Mat. Anal., 14 (1993), pp. 45–58.

[27] ——, *Factorized sparse approximate inverse preconditionings II. Solution of 3D FE systems on massively parallel computers*, Intl. J. High Speed Computing, 7 (1995), pp. 191–215.

[28] H. L. Ong, *Fast approximate solution of large scale sparse linear systems*, J. Comput. Appl. Math., 10 (1984), pp. 45–54.

[29] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, New York, 1995.

[30] K. Takahishi, J. Fagan, and M.-S. Chin, *Formation of a sparse bus impedance matrix and its application to short circuit study*, in Proc. 8th PICA Conf., Minneapolis, Minnesota, June 4–6, 1973, pp. 63–69.

[31] W.-P. Tang, *Effective sparse approximate inverse preconditioners*, Tech. Rep., RIACS, NASA Ames Research Center, Moffet Field, California, 1996, in preparation.