

Sparse symmetric preconditioners for dense linear systems in electromagnetism¹

Bruno Carpentieri², Iain S. Duff³, Luc Giraud⁴, and M. Magolu monga Made⁵

ABSTRACT

We consider symmetric preconditioning strategies for the iterative solution of dense complex symmetric non-Hermitian systems arising in computational electromagnetics. In particular we report on the numerical behaviour of the classical Incomplete Cholesky factorization as well as some of its recent variants and consider also well known factorized approximate inverses. We illustrate the difficulties that those techniques encounter on the linear systems under consideration and give some clues to explain their disappointing behaviour. We propose two symmetric preconditioners based on Frobenius-norm minimization that use a prescribed sparsity pattern. The numerical and computational efficiency of the proposed preconditioners are illustrated on a set of model problems arising both from academic and from industrial applications.

Keywords: Preconditioning techniques, Frobenius-norm minimization method, factorized approximate inverse, Incomplete Cholesky factorization, nonzero pattern selection strategies, electromagnetic scattering applications.

AMS(MOS) subject classifications: 65F10, 65F50, 65N38, 65R20, 78A45, 78A50, 78-08.

¹Current reports available by anonymous ftp to [ftp.numerical.rl.ac.uk](ftp://ftp.numerical.rl.ac.uk) in directory `pub/reports`. This report is available in compressed postscript as file `cdgmRAL2002016.ps.gz` or as the PDF file `cdgmRAL2002016.pdf`. Report also available through URL <http://www.numerical.rl.ac.uk/reports/reports.html>.

An earlier version was published as Technical Report TR/PA/01/35 from CERFACS.

²b.carpentieri@dm.uniba.it, the work of this author was supported by I.N.D.A.M. (Rome, Italy) under a grant (Borsa di Studio per l'estero, Provvedimento del Presidente del 30 Aprile 1998)

³i.s.duff@rl.ac.uk, the work of this author was supported in part by the EPSRC Grant GR/R46441.

⁴giraud@cerfacs.fr, CERFACS, 42 Ave G. Coriolis, 31057 Toulouse Cedex, France.

⁵magolu@ulb.ac.be, Université Libre de Bruxelles, Faculté des Sciences Appliquées, Service de Metrologie Nucléaire, CPI 165/84, 50, av. F.D. Roosevelt, B-1050 Bruxelles.

Computational Science and Engineering Department

Atlas Centre

Rutherford Appleton Laboratory

Oxon OX11 0QX

May 7, 2002

Contents

1	Introduction	1
2	Symmetric preconditioning for dense problems	2
2.1	Frobenius-norm minimization methods	2
2.2	Classical symmetric preconditioners	4
2.2.1	Factorized sparse approximate preconditioners	4
2.2.2	Incomplete Cholesky factorization	5
3	Numerical experiments	6
3.1	Possible causes of failure of factorized approximate inverses	9
3.2	Numerical experiments with Incomplete Cholesky factorization	11
3.3	Study of symmetric Frobenius-norm minimization type preconditioners	17
4	Conclusions	20

1 Introduction

In electromagnetism calculations, finding the scattered wave of a given incident field on a scattering obstacle requires the solution of a linear system of equations. Such analysis, relying on Maxwell's equations, is required in the simulation of many industrial processes coming from antenna design to absorbing materials, electromagnetic compatibility, and so on. Recently the *Boundary Element Method* (BEM) has been successfully employed in the numerical solution of this class of problems, proving to be an effective alternative to common discretization schemes like Finite Element Methods (FEM's), Finite Difference Methods (FDM's) or Finite Volume Method (FVM's). The idea of BEM is to shift the focus from solving a partial differential equation defined on a closed or unbounded domain to solving a boundary integral equation over the finite part of the boundary. The discretization by BEM results in linear systems with dense complex matrices. The coefficient matrix can be symmetric non-Hermitian in the Electric Field Integral Equation formulation (EFIE), or unsymmetric in the Combined Field Integral Equation formulation (CFIE) (see Peterson, Ray and Mittra (1997) for further details). The unknowns are associated with the edges of an underlying mesh on the surface of the object. With the advent of parallel processing, this approach has become viable for large problems and the typical problem size in the electromagnetics industry is continually increasing. Nevertheless, nowadays, many problems can no longer be solved by parallel out-of-core direct solvers as they require too much memory, CPU and disk resources and iterative solvers appear as a viable alternative. Here we will only consider cases where the matrix is symmetric because EFIE usually gives rise to linear systems that are more difficult to solve with iterative methods. Another motivation to focus only on EFIE formulation is that it does not require any restriction on the geometry of the scattering obstacle as CFIE does and in this respect is more general.

Thus, in this paper, we are concerned with symmetric preconditioning of linear systems of equations of the form

$$Ax = b, \tag{1.1}$$

where the n by n coefficient matrix $A = [a_{ij}]$ is dense, complex, symmetric and non-Hermitian, and arises from the discretization of boundary integral equations in electromagnetism. When iterative methods are used for the solution of (1.1), preconditioning plays a key role. Earlier works (Alléon, Benzi and Giraud 1997, Carpentieri, Duff and Giraud 2000) showed that sparse approximate inverse methods based on Frobenius-norm minimization give rise to effective and robust preconditioners. However, the preconditioners considered in these papers were not symmetric and consequently might not have fully exploited all the characteristics of the linear system. To complete these earlier studies, we now investigate implicit and explicit symmetric preconditioners, with an emphasis again on approximate inverse techniques.

In Section 2, we consider different approaches to construct symmetric

preconditioners. More precisely in Section 2.1 we introduce some strategies for building symmetric preconditioners based on Frobenius-norm minimization. In the later sections, we briefly present more classical techniques like a factorized approximate inverse preconditioner namely *AINV* (Benzi 1993, Benzi, Meyer and Tũma 1996, Benzi and Tũma 1998) and *FSAI* (Kolotilina and Yeremin 1993), and incomplete Cholesky factorization (Meijerink and van der Vorst 1977). In Section 3, we study the numerical behaviour of those preconditioners on a set of model problems representative of real calculations in electromagnetics applications. In particular, we give some clues to explain the poor behaviour of some of them. We conclude this paper with some remarks in Section 4.

2 Symmetric preconditioning for dense problems

In this section we consider different methods which compute symmetric preconditioners of both implicit and explicit type for the iterative solution of the linear system (1.1). All the preconditioners are computed using as input \tilde{A} , a sparse approximation of the dense coefficient matrix A . Several heuristics can be used for defining the sparsity pattern for \tilde{A} based either on algebraic considerations (Alléon et al. 1997) or on using information from the underlying mesh (Carpentieri et al. 2000). In the following, we only consider a geometric approach, which is the only one that can be efficiently implemented in a parallel fast multipole environment (Carpentieri, Duff, Giraud and Sylvand 2002). In BEM calculations, each equation is associated with one edge of the mesh and the pattern of \tilde{A} is defined as follows: for each edge we select all those edges within a sufficiently large sphere centered on that edge that defines its geometric neighbourhood. By using a suitable size for this sphere and because of the rapid decay of the Green’s functions, we hope to include the most relevant contributions from A in the approximate matrix \tilde{A} .

2.1 Frobenius-norm minimization methods

A natural way to compute an explicit preconditioner is based on Frobenius-norm minimization (Benson 1973, Benson and Frederickson 1982, Benson, Krettmann and Wright 1984, Frederickson 1975). The idea is to compute the sparse approximate inverse as the matrix M which minimizes $\|I - M\tilde{A}\|_F$ (or $\|I - \tilde{A}M\|_F$ for right preconditioning) subject to certain sparsity constraints. The Frobenius norm is usually chosen since it allows the decoupling of the constrained minimization problem into n independent linear least-squares problems, one for each column of M (when preconditioning from the right) or row of M (when preconditioning from the left).

The independence of these least-squares problems follows immediately from the

identity:

$$\|I - M\tilde{A}\|_F^2 = \|I - \tilde{A}M^T\|_F^2 = \sum_{j=1}^n \|e_j - \tilde{A}m_{j\bullet}\|_2^2 \quad (2.1)$$

where e_j is the j th canonical unit vector and $m_{j\bullet}$ is the column vector representing the j th row of M .

In the case of right preconditioning, the analogous relation

$$\|I - \tilde{A}M\|_F^2 = \sum_{j=1}^n \|e_j - \tilde{A}m_{\bullet j}\|_2^2 \quad (2.2)$$

holds, where $m_{\bullet j}$ is the column vector representing the j th column of M . Clearly, there is considerable scope for parallelism in this approach. By construction, the sparse approximate inverse computed is not guaranteed to be symmetric, and usually is not, even if a symmetric pattern is imposed on M . This fact prevents the use of symmetric Krylov solvers.

In this section we consider two possible symmetrization strategies for Frobenius-norm minimization using a prescribed pattern for the preconditioner based on geometric information. Similarly to the procedure used to define \tilde{A} , for each edge we select all those edges within a sufficiently large sphere that defines our geometric neighbourhood. By using a suitable size for this sphere, we hope to include the most relevant contributions to the inverse and consequently to obtain an effective sparse approximate inverse. For those preconditioners, we select a smaller sphere than the one used to define \tilde{A} . Consequently we compute less nonzeros than the number retained in the sparse approximation of A (we refer to Carpentieri et al. (2000) for the complete and detailed description). If M_{Frob} denotes the unsymmetric matrix resulting from the minimization (2.1), the first strategy simply averages its off-diagonal entries. That is

$$M_{Aver-Frob} = \frac{M_{Frob} + M_{Frob}^T}{2}. \quad (2.3)$$

An alternative way to construct a symmetric sparse approximate inverse is to only compute the lower triangular part, including the diagonal, of the preconditioner. The nonzeros calculated are reflected with respect to the diagonal and are used to update the right-hand sides of the subsequent least-squares problems involved in the construction of the remaining columns of the preconditioner. More precisely, in the computation of the k -th column of the preconditioner, the entries m_{ik} for $i < k$ are set to m_{ki} that are already available and only the lower diagonal entries are computed. The entries m_{ki} are then used to update the right-hand sides of the least-squares problems which involve the remaining unknowns m_{ik} , for $k \geq i$. The least-squares problems are as follows:

$$\min \|\hat{e}_j - \tilde{A}\hat{m}_{\bullet j}\|_2^2 \quad (2.4)$$

where $\hat{e}_j = e_j - \sum_{k < j} \tilde{a}_{\bullet k} m_{kj}$ and $\hat{m}_{\bullet j} = (0, \dots, 0, m_{jj}, \dots, m_{nj})^T$. In the following, this preconditioner is referred to as $M_{Sym-Frob}$. It should be noted that the preconditioner built using this approach no longer minimizes any Frobenius norm and it might be sensitive to the ordering of columns. In addition, if ℓ denotes the number of nonzeros entries in $M_{Sym-Frob}$, this method only computes $(\ell + n)/2$ nonzeros. Thus the overall computational complexity for the construction of $M_{Sym-Frob}$ can be considerably less than for $M_{Aver-Frob}$ as the least-squares problems are usually solved by QR factorizations whose complexity is of the order of the square in the number of unknowns and is linear in the number of equations.

2.2 Classical symmetric preconditioners

2.2.1 Factorized sparse approximate preconditioners

An alternative way to construct a symmetric sparse approximate inverse is to compute it in factorized form. In this paper we consider two classical techniques, the first constructs an approximation of the inverse of the factors using an \tilde{A} -biconjugation process (Benzi and Tuma 1998) and the other one a Frobenius-norm minimization technique (Kolotilina and Yereimin 1993).

If the matrix \tilde{A} can be written in the form LDL^T where L is unit lower triangular and D is diagonal, then its inverse can be decomposed as $\tilde{A}^{-1} = L^{-T}D^{-1}L^{-1} = ZD^{-1}Z^T$ where $Z = L^{-T}$ is unit triangular. Factorized sparse approximate inverse techniques compute sparse approximations $\tilde{Z} \approx Z$ so that the resulting preconditioner will be $M = \tilde{Z}\tilde{D}^{-1}\tilde{Z}^T \approx \tilde{A}^{-1}$, for $\tilde{D} \approx D$.

In the approach known as *AINV*, the triangular factor is computed by means of a set of \tilde{A} -biconjugate vectors $\{z_i\}_{i=1}^n$ such that $z_i^T \tilde{A} z_j = 0$ if and only if $i \neq j$. Then, introducing the matrix $Z = [z_1, z_2, \dots, z_n]$ the relation

$$Z^T \tilde{A} Z = D = \begin{pmatrix} p_1 & 0 & \dots & 0 \\ 0 & p_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & p_n \end{pmatrix}$$

holds, where $p_i = z_i^T \tilde{A} z_i \neq 0$, and the inverse is equal to

$$\tilde{A}^{-1} = Z D^{-1} Z^T = \sum_{i=1}^n \frac{z_i z_i^T}{p_i}.$$

The two sets of \tilde{A} -biconjugate vectors are computed by means of a (two-sided) Gram-Schmidt orthogonalization process with respect to the bilinear form associated with \tilde{A} . *AINV* does not require a pattern prescribed in advance for the approximate inverse factors, and sparsity is preserved during the process, by discarding elements having magnitude smaller than a given positive threshold. The value of this threshold governs the density of the preconditioner.

An alternative approach was proposed by Kolotilina and Yeremin in a series of papers (Kolotilina and Yeremin 1993, Kolotilina and Yeremin 1995, Kolotilina, Yeremin and Nikishin 2000, Kolotilina, Yeremin and Nikishin 1999). This approach, known as *FSAI*, approximates \tilde{A}^{-1} by the factorization $G^T G$, where G is a sparse lower triangular matrix approximating the inverse of the lower triangular Cholesky factor, \tilde{L} , of \tilde{A} . This technique has obtained good results on some difficult problems and is suitable for parallel implementation, but it requires an *a priori* prescription for the sparsity pattern for the approximate factors. Minimizing $\|I - G\tilde{L}\|_F^2$ can be accomplished without knowing the Cholesky factor \tilde{L} by solving the normal equations

$$\{G\tilde{L}\tilde{L}^T\}_{ij} = \tilde{L}_{ij}^T, \quad (i, j) \in \mathcal{S}_{\tilde{L}} \quad (2.5)$$

where $\mathcal{S}_{\tilde{L}}$ is a lower triangular nonzero pattern for G . Equation (2.5) can be replaced by

$$\{\tilde{G}\tilde{A}\}_{ij} = I_{ij}, \quad (i, j) \in \mathcal{S}_{\tilde{L}} \quad (2.6)$$

where $\tilde{G} = \tilde{D}^{-1}G$ and \tilde{D} is the diagonal of \tilde{L} . Then, each row of \tilde{G} can be computed independently by solving a small linear system. The preconditioned linear system has the form

$$G\tilde{A}G^T = \tilde{D}\tilde{G}\tilde{A}\tilde{G}^T\tilde{D}.$$

The matrix \tilde{D} is not known and is generally chosen so that the diagonal of $G\tilde{A}G^T$ is all ones. The FSAI preconditioner is well defined for Hermitian matrices, but might not be defined for general matrices. In our implementation, the prescription for the sparsity pattern for the approximate factors is performed using the same heuristic that is used to sparsify the dense matrix A and is based on geometric information.

2.2.2 Incomplete Cholesky factorization

In this section we consider another classical symmetric preconditioner, that is the incomplete Cholesky factorization normally denoted by *IC*. We assume that the standard *IC* factorization matrix M of \tilde{A} is given in the following form

$$M = \mathcal{L}\mathcal{D}\mathcal{L}^T, \quad (2.7)$$

where \mathcal{D} and \mathcal{L} stand for, respectively, the diagonal matrix and the unit lower triangular matrix whose entries are computed by means of the algorithm given in Figure 2.1. The set \mathcal{F} of fill-in entries to be kept is given by

$$\mathcal{F} = \{ (k, i) \mid lev(l_{k,i}) \leq \ell \},$$

where integer ℓ denotes a user specified maximal fill-in level. The level $lev(l_{k,i})$ of the coefficient $l_{k,i}$ of \mathcal{L} is defined by:

Initialization

$$lev(l_{k,i}) = \begin{cases} 0 & \text{if } l_{k,i} \neq 0 \text{ or } k = i \\ \infty & \text{otherwise} \end{cases}$$

Factorization

$$lev(l_{k,i}) = \min \{ lev(l_{k,i}), lev(l_{i,j}) + lev(l_{k,j}) + 1 \} .$$

The resulting preconditioner is usually denoted by $IC(\ell)$. Alternative strategies that dynamically discard fill-in entries are summarized in (Saad 1996).

Compute \mathcal{D} and \mathcal{L}

Initialization phase
 $d_{i,i} = \tilde{a}_{i,i} , \quad i = 1, 2, \dots, n$
 $l_{i,j} = \tilde{a}_{i,j} , \quad i = 2, \dots, n , \quad j = 1, 2, \dots, i - 1$

Incomplete factorization process

do $j = 1, 2, \dots, n - 1$

do $i = j + 1, j + 2, \dots, n$

$d_{i,i} = d_{i,i} - \frac{l_{i,j}^2}{d_{j,j}}$

$l_{i,j} = \frac{l_{i,j}}{d_{j,j}}$

do $k = i + 1, i + 2, \dots, n$

if $(i, k) \in \mathcal{F}$ $l_{k,i} = l_{k,i} - l_{i,j} l_{k,j}$

end do

end do

end do

Figure 2.1: Incomplete factorization algorithm - $M = \mathcal{L}\mathcal{D}\mathcal{L}^T$.

3 Numerical experiments

To study the numerical behaviour of the preconditioners described in the previous section, we consider a set of test examples representative of calculations in electromagnetics applications. Those tests examples are defined by:

Example 1: a cylinder with a hollow inside, a matrix of order $n = 1080$, see Figure 1(a);

Example 2: a cylinder with a break on the surface, a matrix of order $n = 1299$, see Figure 1(b);

Example 3: a satellite, a matrix of order $n = 1701$, see Figure 1(c);

Example 4: a parallelepiped, a matrix of order $n = 2016$, see Figure 1(d); and

Example 5: a sphere, a matrix of order $n = 2430$, Figure 1(e).

We mention that, for physical consistency, we have set the frequency of the incident wave so that there are about ten discretization points per wavelength (Bendali 1984).

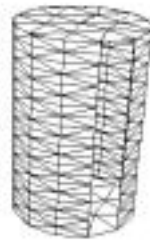
We investigate the behaviour of the preconditioners when used to accelerate restarted GMRES (Saad and Schultz 1986) and symmetric QMR (Freund and Nachtigal 1994). For all the numerical experiments with GMRES we use the implementation described by Frayssé, Giraud and Gratton (1997). In each case, we take as the initial guess $x_0 = 0$, and the right-hand side is such that the exact solution of the system is known. We perform different tests with different known solutions, observing identical results. The stopping criterion in all cases just consists in reducing the original residual by 10^{-5} that then can be related to a norm-wise backward error. In all the tables, the symbol “-” means that convergence is not obtained after 500 iterations. All the numerical experiments are performed in double precision complex arithmetic on a SGI Origin 2000 and the number of iterations reported in this paper are for right preconditioning. Finally we mention that no explicit scaling is performed as the matrices are already well scaled.

In order to illustrate the trend in the behaviour of these preconditioners, we first show in Table 3.1 the number of iterations required to compute the solution on Example 2. All the preconditioners are computed using the same sparse approximation of the original matrix. For all the preconditioners we tune their governing parameters so that they all have roughly the same number of nonzeros entries. In Table 3.1 we give the number of iterations for both GMRES and SQMR that actually also corresponds to the number of matrix-vector products that is the most time consuming part of the algorithms. Nevertheless, it should be noted that for the other parts of the algorithms the coupled two term recurrences of SQMR is much cheaper than the orthogonalization and least-squares solution involved in GMRES. From a memory point of view, SQMR is also much less demanding; if we used the same memory workspace for GMRES as for SQMR, the largest restart would be 5.

Frobenius-norm minimization type methods can deliver a good rate of convergence compared to standard IC , while the results observed with factorized approximate inverses are disappointing. For AINV, we can obtain a preconditioner that leads to similar numerical convergence as the one reported for the M_{Frob} variants if we use a threshold equal to 10^{-3} that gives rise to a preconditioner



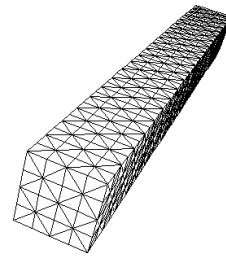
(a) Example 1



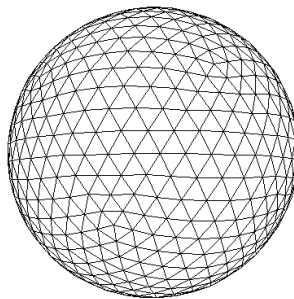
(b) Example 2



(c) Example 3



(d) Example 4



(e) Example 5

Figure 3.1: Mesh associated with test examples

Example 2 - Density of $\tilde{A} = 3.18\%$ - Density of $M = 1.99\%$				
Precond.	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR
M_{Frob}	57	43	43	*
$M_{Aver-Frob}$	59	44	44	34
$M_{Sym-Frob}$	60	46	46	41
$IC(0)$	-	238	117	128
$AINV$	-	-	-	-
$FSAI$	-	-	-	-

Table 3.1: Number of iterations using both symmetric and unsymmetric preconditioners and Krylov methods on a test example.

that is about 75 % dense. Another alternative for $AINV$ is to consider a twice as dense approximation of A , in that case the threshold that ensures a comparable numerical convergence is 10^{-2} with a preconditioner that only is about 35 % dense. For $FSAI$ comparable observations can be made. Using the same \tilde{A} as in Table 3.1 a preconditioner that is 75 % dense exhibits a similar convergence behaviour. In any case, the resulting preconditioners are still much denser than those required for the M_{Frob} variants. We intend, in the following sections, to understand the numerical behaviour of these methods on electromagnetic problems and to identify some potential causes of failure.

3.1 Possible causes of failure of factorized approximate inverses

One potential difficulty with the factorized approximate inverse method $AINV$ is the tuning of the threshold parameter that controls the fill-in in the inverse factors. For a typical example we display in Figure 3.2 the sparsity pattern of A^{-1} (a) and L^{-1} (b), the inverse of its Cholesky factor, where all the entries smaller than 5.0×10^{-2} have been dropped after a symmetric scaling such that $\max_i |a_{ji}| = \max_i |\ell_{ji}| = 1$. The location of the large entries in the inverse matrix exhibit some structure. In addition, only a very small number of its entries have large magnitude compared to the others that are much smaller. This fact has been successfully exploited to define various *a priori* pattern selection strategies for Frobenius norm minimization preconditioners (Alléon et al. 1997, Carpentieri et al. 2000) in a non-factorized form. On the contrary, the inverse factors can be totally unstructured as shown in Figure 3.2 (b). In this case, the *a priori* selection of a sparse pattern for the factors can be extremely hard as no real structures are revealed, preventing the use of techniques like $FSAI$. In Figure 3.3 we plot the magnitude of the entries in the first column of A^{-1} and L^{-1} with respect to their row index. Those plots indicate that any dropping strategy, either static or dynamic, may be very difficult to tune as it can easily discard relevant information and potentially lead to a very

poor preconditioner. Selecting too small a threshold would retain too many entries and lead to a fairly dense preconditioner. A larger threshold would yield a sparser preconditioner but might discard too many entries of moderate magnitude that are important for the preconditioner. For those problems, finding the appropriate threshold to enable a good trade-off between sparsity and numerical efficiency is challenging and very problem-dependent.

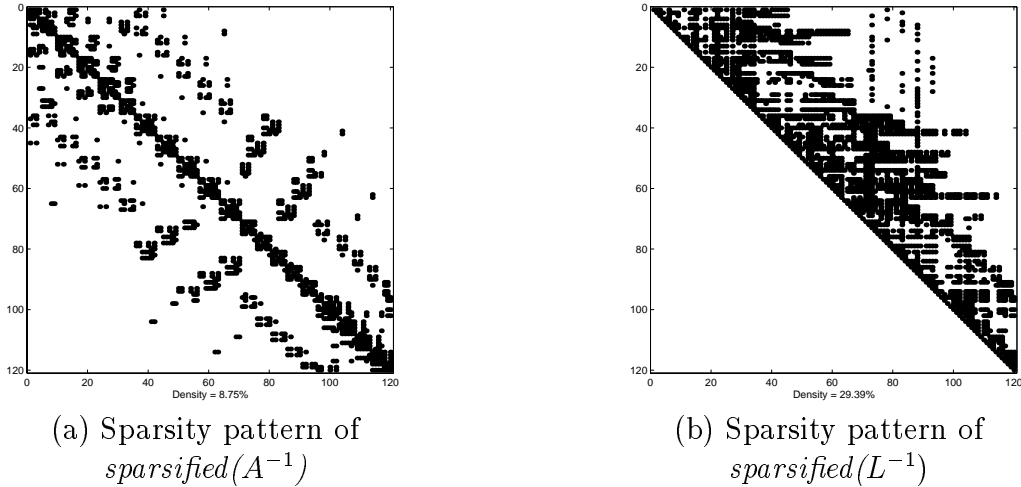


Figure 3.2: Sparsity patterns of the inverse of A (on the left) and of the inverse of its lower triangular factor (on the right), where all the entries whose relative magnitude is smaller than 5.0×10^{-2} are dropped. The test problem, representative of the general trend, is a small sphere.

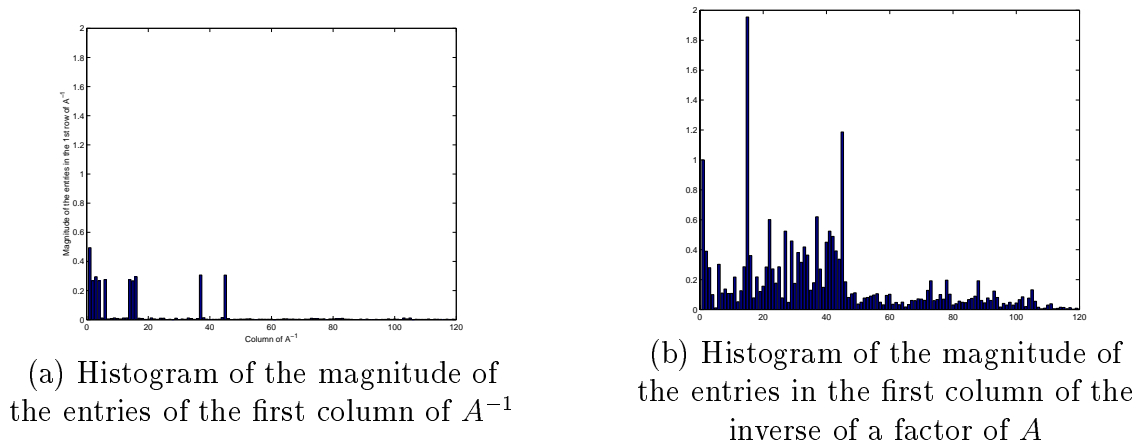


Figure 3.3: Histograms of the magnitude of the entries of one column of A^{-1} and its lower triangular factor. A similar behaviour has been observed for all the other columns. The test problem, representative of the general trend, is a small sphere.

3.2 Numerical experiments with Incomplete Cholesky factorization

In Table 3.2, we display the number of iterations using an incomplete Cholesky factorization preconditioner on Example 3. The reported behaviour is representative of what was observed on the other test examples. We show results for increasing values of the density for the sparse approximation of A as well as various levels of fill-in. The general trend is that increasing the fill-in generally produces a much more robust preconditioner than $IC(0)$ applied to a denser sparse approximation of the original matrix. Moreover, $IC(\ell)$ with $\ell \geq 1$ might deliver a good rate of convergence if the coefficient matrix is not too sparse.

Example 3				
Density of $\tilde{A} = 2.97\%$				
$IC(\text{level})$	Density of M	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	2.97 %	–	300	154
$IC(1)$	5.70 %	239	131	102
$IC(2)$	8.27 %	173	95	80
Density of $\tilde{A} = 4.05\%$				
$IC(\text{level})$	Density of M	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	4.05 %	–	–	207
$IC(1)$	7.90 %	80	50	54
$IC(2)$	11.39 %	39	33	33
Density of $\tilde{A} = 5.03\%$				
$IC(\text{level})$	Density of M	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	5.03 %	–	–	205
$IC(1)$	9.75 %	100	75	58
$IC(2)$	13.94 %	24	24	21
Density of $\tilde{A} = 6.05\%$				
$IC(\text{level})$	Density of M	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	6.05 %	–	–	173
$IC(1)$	12.03 %	75	48	50
$IC(2)$	17.07 %	19	19	18

Table 3.2: Number of iterations varying the sparsity level of \tilde{A} and the level of fill-in on Example 3.

However, on indefinite problems the numerical behaviour of IC can be fairly chaotic. In Table 3.3 we show the number of iterations for Example 5. As in the previous table, the preconditioner is computed from a sparse approximation of the dense coefficient matrix. Different values of density in the sparsified matrix are

considered and various levels of fill-in are allowed in the factors. The factorization of a very sparse approximation (up to 2 %) of the coefficient matrix can be stable and deliver a good rate of convergence, especially if at least one level of fill-in is retained. For higher values of density for the approximation of A , the factors may become very ill-conditioned and consequently the preconditioner is very poor. This behaviour has been already observed on sparse real indefinite systems, see for instance Chow and Saad (1997). As an attempt for a possible remedy, following monga Made (2001) and monga Made, Beauwens and Warzee (2000), we apply $IC(\ell)$ to a perturbation of \tilde{A} by a complex diagonal matrix, more specifically, we use

$$\tilde{A}_\tau = \tilde{A} + \mathbf{i}\tau h \Delta_r, \quad (3.1)$$

where $\Delta_r = \text{diag}(\text{Re}(A)) = \text{diag}(\text{Re}(\tilde{A}))$, and τ stands for a nonnegative real parameter, while

$$h = n^{-\frac{1}{d}} \quad \text{with} \quad d = 3 \quad (\text{the space dimension}). \quad (3.2)$$

The intention is to move the eigenvalues of the preconditioned system along the imaginary axis and thus avoid a possible eigenvalue cluster close to zero.

In Table 3.4, we show the number of SQMR iterations for different values of τ , the shift parameter, and various level of fill-in in the preconditioner. Although it is not easy to tune and its effect is difficult to predict, a small diagonal shift can help to compute a more stable factorization, and in some cases the performance of the preconditioner can significantly improve.

In Figures 3.4 and 3.5, we illustrate the effect of this shift strategy on the eigenvalue distribution of the preconditioned matrix. For each value of the shift parameter τ , we display $\kappa(\mathcal{L})$, the condition number of the computed \mathcal{L} factor, and the number of iterations required by SQMR. The eigenvalues are scattered all over the complex plane when no shift is used, whereas they look more clustered when a shift is applied. A clustered spectrum of the preconditioned matrix is usually considered as a desirable property for a fast convergence of Krylov solvers. However, for incomplete factorizations the condition number of the factors plays a more important role than the eigenvalue distribution on the rate of convergence of the Krylov iterations.

One possibility for constructing a robust shifted IC factorization would be to implement an auto-tuned strategy. This would consist in incrementing the value of the shift and computing a new incomplete factorization if the condition number of the current factor is too large. Such a procedure might be very time consuming; for this reason we do not explore it further here.

Density of $\tilde{A} = 1.00\%$ - $\kappa_\infty(\tilde{A}) = 198$					
$IC(\text{level})$	Density of M	$\kappa_\infty(\mathcal{L})$	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	1.00 %	$2 \cdot 10^4$	–	–	–
$IC(1)$	1.92 %	$4 \cdot 10^2$	148	127	63
$IC(2)$	3.04 %	$8 \cdot 10^2$	92	86	48
Density of $\tilde{A} = 2.01\%$ - $\kappa_\infty(\tilde{A}) = 263$					
$IC(\text{level})$	Density of M	$\kappa_\infty(\mathcal{L})$	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	2.00 %	$2 \cdot 10^3$	–	–	–
$IC(1)$	4.17 %	$1 \cdot 10^3$	–	–	–
$IC(2)$	6.71 %	$9 \cdot 10^2$	–	–	145
Density of $\tilde{A} = 3.00\%$ - $\kappa_\infty(\tilde{A}) = 270$					
$IC(\text{level})$	Density of M	$\kappa_\infty(\mathcal{L})$	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	3.00 %	$1 \cdot 10^6$	–	–	–
$IC(1)$	6.62 %	$1 \cdot 10^5$	–	–	–
$IC(2)$	10.68 %	$3 \cdot 10^3$	362	125	77
Density of $\tilde{A} = 3.99\%$ - $\kappa_\infty(\tilde{A}) = 253$					
$IC(\text{level})$	Density of M	$\kappa_\infty(\mathcal{L})$	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	3.99 %	$6 \cdot 10^9$	–	–	–
$IC(1)$	8.95 %	$2 \cdot 10^5$	–	–	–
$IC(2)$	14.21 %	$7 \cdot 10^3$	58	49	46
Density of $\tilde{A} = 5.00\%$ - $\kappa_\infty(\tilde{A}) = 285$					
$IC(\text{level})$	Density of M	$\kappa_\infty(\mathcal{L})$	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	5.00 %	$6 \cdot 10^{10}$	–	–	–
$IC(1)$	11.60 %	$1 \cdot 10^5$	–	–	–
$IC(2)$	18.16 %	$3 \cdot 10^4$	49	44	31
Density of $\tilde{A} = 6.01\%$ - $\kappa_\infty(\tilde{A}) = 294$					
$IC(\text{level})$	Density of M	$\kappa_\infty(\mathcal{L})$	GMRES(30)	GMRES(50)	SQMR
$IC(0)$	6.01 %	$8 \cdot 10^{11}$	–	–	–
$IC(1)$	14.09 %	$5 \cdot 10^{11}$	–	–	–
$IC(2)$	21.57 %	$7 \cdot 10^4$	44	41	27

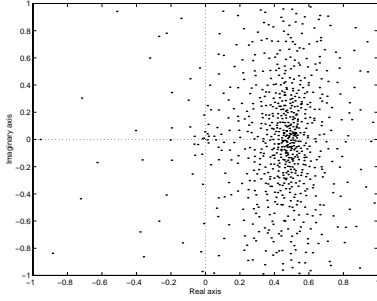
Table 3.3: Number of iterations and condition numbers of \tilde{A} and \mathcal{L} varying the sparsity level of \tilde{A} and the level of fill-in on Example 5.

Example 1 - Density of $\tilde{A} = 5.03\%$								
$IC(\text{level})$	Density of M	τ						
		0.0	0.1	0.3	0.5	0.7	0.9	1.1
$IC(0)$	5.0%	–	–	318	193	133	130	120
$IC(1)$	16.9%	–	–	313	161	117	104	95
$IC(2)$	32.3%	–	–	275	138	121	90	99

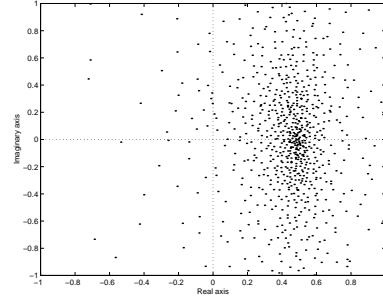
Example 3 - Density of $\tilde{A} = 2.97\%$								
$IC(\text{level})$	Density of M	τ						
		0.0	0.1	0.3	0.5	0.7	0.9	1.1
$IC(0)$	3.0%	154	309	–	334	254	247	224
$IC(1)$	7.1%	102	197	439	436	272	244	237
$IC(2)$	11.3%	80	264	–	334	327	243	224

Example 5 - Density of $\tilde{A} = 3.99\%$								
$IC(\text{level})$	Density of M	τ						
		0.0	0.1	0.3	0.5	0.7	0.9	1.1
$IC(0)$	4.0%	–	–	–	–	–	–	–
$IC(1)$	11.7%	–	–	–	–	–	–	–
$IC(2)$	19.0%	25	77	279	–	–	–	–

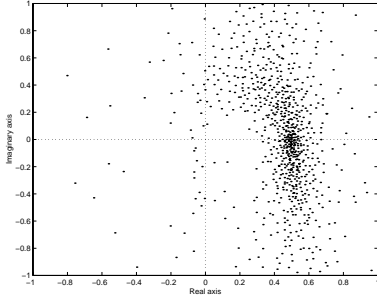
Table 3.4: Number of SQMR iterations varying the shift parameter for various level of fill-in in IC .



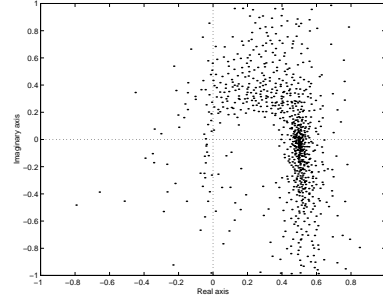
(a) $\tau = 0.0 - \kappa(\mathcal{L}) = 526284 - \text{iter.} = +500$



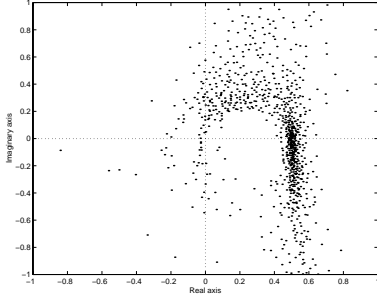
(b) $\tau = 0.1 - \kappa(\mathcal{L}) = 134975 - \text{iter.} = +500$



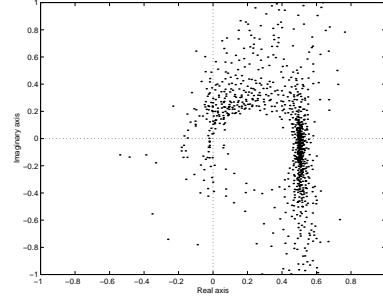
(a) $\tau = 0.3 - \kappa(\mathcal{L}) = 9608 - \text{iter.} = 313$



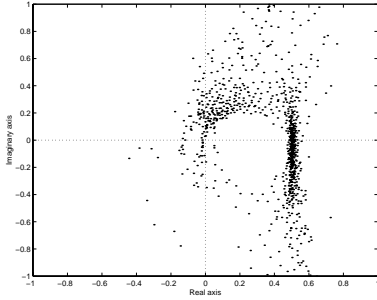
(b) $\tau = 0.5 - \kappa(\mathcal{L}) = 2165 - \text{iter.} = 161$



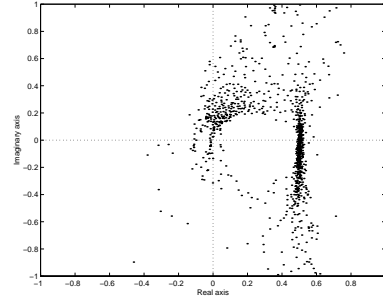
(c) $\tau = 0.7 - \kappa(\mathcal{L}) = 777 - \text{iter.} = 117$



(d) $\tau = 0.9 - \kappa(\mathcal{L}) = 434 - \text{iter.} = 104$

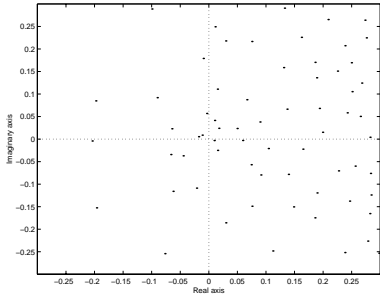


(c) $\tau = 1.1 - \kappa(\mathcal{L}) = 261 - \text{iter.} = 95$

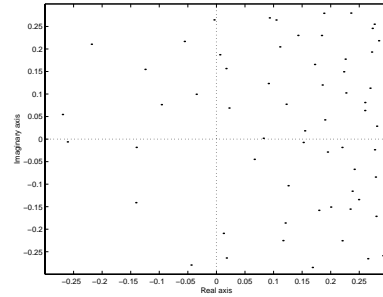


(d) $\tau = 1.3 - \kappa(\mathcal{L}) = 183 - \text{iter.} = 94$

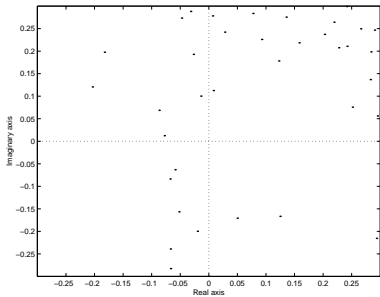
Figure 3.4: The eigenvalue distribution on the square $[-1, 1]$ of the matrix preconditioned with $IC(1)$, the condition number of \mathcal{L} and the number of iterations with SQMR for various values of the shift parameter τ . The test problem is Example 1.



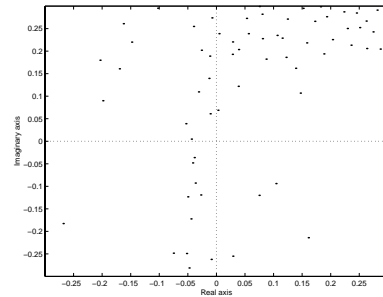
(a) $\tau = 0.0 - \kappa(\mathcal{L}) = 526284 - \text{iter.} = +500$



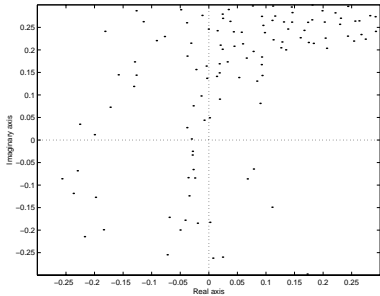
(b) $\tau = 0.1 - \kappa(\mathcal{L}) = 134975 - \text{iter.} = +500$



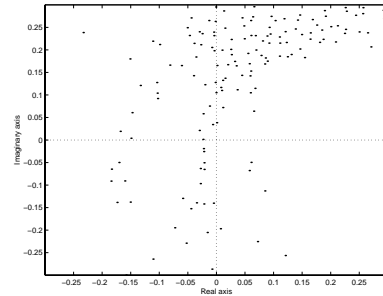
(a) $\tau = 0.3 - \kappa(\mathcal{L}) = 9608 - \text{iter.} = 313$



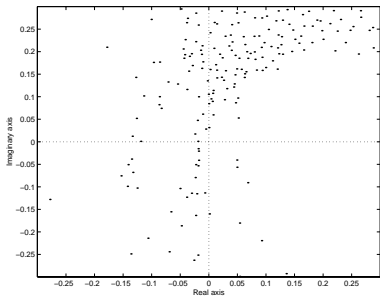
(b) $\tau = 0.5 - \kappa(\mathcal{L}) = 2165 - \text{iter.} = 161$



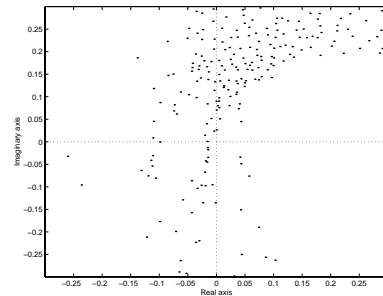
(c) $\tau = 0.7 - \kappa(\mathcal{L}) = 777 - \text{iter.} = 117$



(d) $\tau = 0.9 - \kappa(\mathcal{L}) = 434 - \text{iter.} = 104$



(c) $\tau = 1.1 - \kappa(\mathcal{L}) = 261 - \text{iter.} = 95$



(d) $\tau = 1.3 - \kappa(\mathcal{L}) = 183 - \text{iter.} = 94$

Figure 3.5: The eigenvalue distribution on the square $[-0.3, 0.3]$ of the matrix preconditioned with $IC(1)$, the condition number of \mathcal{L} and the number of iterations with SQMR for various values of the shift parameter τ . The test problem is Example 1.

3.3 Study of symmetric Frobenius-norm minimization type preconditioners

In Table 3.5, we show the numerical behaviour of the different Frobenius-norm minimization type preconditioners, both symmetric and unsymmetric. We compare the unsymmetric preconditioner M_{Frob} and the two symmetric preconditioners $M_{Aver-Frob}$ and $M_{Sym-Frob}$. The column entitled “Relative Flops” displays the ratio $\frac{\sigma_{QR}(M)}{\sigma_{QR}(M_{Frob})}$, where the $\sigma_{QR}(M)$ represents the number of floating-point operations required by the sequence of QR factorizations used to build the preconditioner M , that is either $M = M_{Aver-Frob}$ or $M = M_{Sym-Frob}$. In this table, it can be seen that $M_{Aver-Frob}$ almost always requires less iterations than $M_{Sym-Frob}$ that imposes the symmetry directly and consequently only computes half of the entries. Since $M_{Sym-Frob}$ computes less entries the associated values in the column “Relative Flops” are all less than one and close to a third in all cases. On the hardest test cases (Examples 1 and 3) the combination SQMR and $M_{Aver-Frob}$ needs less than half the iterations of M_{Frob} with GMRES(30) and is only very slightly less efficient than M_{Frob} and GMRES(80). On the less difficult problems, SQMR plus $M_{Aver-Frob}$ converges between 21 and 37 % faster than GMRES(80) plus M_{Frob} and between 31 and 43 % faster than GMRES(30) plus M_{Frob} . $M_{Sym-Frob}$, that only computes half of the entries of the preconditioner, has a poor convergence behaviour on the hardest problems and is slightly less efficient than $M_{Aver-Frob}$ on the other problems when used with SQMR. Nevertheless, we should mention that, for the sake of comparison, those preliminary results have been performed using the set of parameters for the density of \tilde{A} and M that were the best for M_{Frob} and consequently nearly optimal for $M_{Aver-Frob}$; the performance of $M_{Sym-Frob}$ might be improved as shown by the results depicted in Table 3.6. Nevertheless, those first experiments reveal the remarkable robustness of SQMR when used in combination with a symmetric preconditioner. This combination generally outperforms GMRES even for large restarts.

The best alternative for significantly improving the behaviour of $M_{Sym-Frob}$ is to enlarge significantly the density of \tilde{A} and only marginally increase the density of the preconditioner. In Table 3.6, we show the number of iterations observed with this strategy that consists in using a density of \tilde{A} that is three times larger than that for $M_{Sym-Frob}$; we recall that for $M_{Aver-Frob}$ and M_{Frob} a density of \tilde{A} twice as large as that of the preconditioner is usually the best trade-off between computing cost and numerical efficiency. It can be seen that $M_{Sym-Frob}$ is slightly better than $M_{Aver-Frob}$ (as in Table 3.5) but it is less expensive to build. In this table, we consider the same values for $\sigma_{QR}(M_{Frob})$ as those in Table 3.5 to evaluate the ratio “Relative Flops”.

To illustrate the effect of the densities of \tilde{A} and of the preconditioners, we performed experiments with preconditioned SQMR, where the preconditioners are built by using either the same sparsity pattern for \tilde{A} or a two, three or five times

Example 1 - Density of $\tilde{A} = 10.13\%$ - Density of $M = 5.03\%$					
Precond.	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR	Relative Flops
M_{Frob}	108	60	60	*	1.00
$M_{Aver-Frob}$	171	79	79	74	1.00
$M_{Sym-Frob}$	–	–	301	–	0.25
Example 2 - Density of $\tilde{A} = 3.17\%$ - Density of $M = 1.99\%$					
Precond.	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR	Relative Flops
M_{Frob}	57	43	43	*	1.00
$M_{Aver-Frob}$	59	44	44	34	1.00
$M_{Sym-Frob}$	60	46	46	41	0.28
Example 3 - Density of $\tilde{A} = 4.72\%$ - Density of $M = 2.35\%$					
Precond.	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR	Relative Flops
M_{Frob}	89	57	57	*	1.00
$M_{Aver-Frob}$	122	63	63	58	1.00
$M_{Sym-Frob}$	318	135	91	102	0.29
Example 4 - Density of $\tilde{A} = 2.08\%$ - Density of $M = 1.04\%$					
Precond.	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR	Relative Flops
M_{Frob}	58	48	48	*	1.00
$M_{Aver-Frob}$	59	47	47	30	1.00
$M_{Sym-Frob}$	63	51	51	33	0.30
Example 5 - Density of $\tilde{A} = 1.25\%$ - Density of $M = 0.62\%$					
Precond.	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR	Relative Flops
M_{Frob}	35	33	33	*	1.00
$M_{Aver-Frob}$	35	34	34	24	1.00
$M_{Sym-Frob}$	51	38	38	32	0.31

Table 3.5: Number of iterations on the test examples using the same pattern for the preconditioners.

denser pattern for \tilde{A} . We report in Tables 3.7 and 3.8 respectively the number of SQMR iterations for $M_{Sym-Frob}$, and for $M_{Aver-Frob}$ respectively. In these tables, $M_{Sym-Frob}$ always requires more iterations than $M_{Aver-Frob}$ for the same values of density for \tilde{A} and for the preconditioner, but its computation costs about a fourth flops for each test.

Because the construction of $M_{Sym-Frob}$ is dependent on the ordering selected, a natural question concerns the sensitivity of the quality of the preconditioner to this. In particular Duff and Meurant (1989) show that the numerical behaviour of IC is very much dependent on the ordering. The effect of the ordering is also highlighted by Benzi and Tuma (2000) for AINV. In Table 3.9, we display the number of iterations with SQMR, selecting the same density parameters as those

Example	Density	GMRES(30)	GMRES(80)	GMRES(∞)	SQMR	Relative Flops
1	$\tilde{A} = 11.98\%$ $M = 6.10\%$	172	68	68	67	0.40
2	$\tilde{A} = 5.94\%$ $M = 2.04\%$	56	41	41	33	0.30
3	$\tilde{A} = 11.01\%$ $M = 3.14\%$	88	57	57	56	0.66
4	$\tilde{A} = 2.08\%$ $M = 1.19\%$	56	50	50	32	0.47
5	$\tilde{A} = 1.98\%$ $M = 0.62\%$	33	33	33	15	0.34

Table 3.6: Number of iterations for $M_{Sym-Frob}$ combined with SQMR using 3 times more nonzeros in \tilde{A} than in the preconditioner.

Example 1										
Density strategy	Percentage density of M									
	1	2	3	4	5	6	7	8	9	10
Same	–	–	–	–	–	180	150	118	105	55
2.0 times	–	–	–	–	–	67	56	48	91	42
3.0 times	–	–	–	–	393	55	52	47	74	39
5.0 times	–	–	–	–	346	53	50	45	56	39

Table 3.7: Number of iterations of SQMR with $M_{Sym-Frob}$ with different values for the density of M using the same pattern for A and larger patterns. The test problem is Example 1.

Example 1										
Density strategy	Percentage density of M									
	1	2	3	4	5	6	7	8	9	10
Same	–	–	–	336	78	55	55	45	38	40
2.0 times	–	–	426	105	81	50	48	43	43	44
3.0 times	–	426	293	113	92	49	45	36	35	35
5.0 times	–	315	248	114	80	44	38	37	37	35

Table 3.8: Number of iterations of SQMR with $M_{Aver-Frob}$ with different values for the density of M using the same pattern for A and larger patterns. The test problem is Example 1.

used for the experiments reported in Table 3.9, but using different orderings to permute the original pattern of $M_{Sym-Frob}$. More precisely we consider the reverse Cuthill-McKee ordering (RCM) (Cuthill and McKee 1969), the minimum degree

ordering (MD) (George and Liu 1989, Tinney and Walker 1967), the spectral nested dissection ordering (SND) (Pothen, Simon and Liou 1990) and lastly we reorder the matrix by putting the denser rows and columns first (DF). It can be seen that $M_{Sym-Frob}$ is not too sensitive to the ordering and none of the tested orderings appears superior to the others.

Example	Density	Original	RCM	MD	SND	DF
1	$\bar{A} = 11.98\%$ $M = 6.10\%$	67	93	93	75	87
2	$\bar{A} = 5.94\%$ $M = 2.04\%$	33	41	40	40	44
3	$\bar{A} = 11.01\%$ $M = 3.14\%$	56	51	68	73	77
4	$\bar{A} = 2.08\%$ $M = 1.19\%$	32	42	40	39	39
5	$\bar{A} = 1.98\%$ $M = 0.62\%$	15	26	25	26	23

Table 3.9: Number of iterations of SQMR with $M_{Sym-Frob}$ with different orderings.

4 Conclusions

In this work we have investigated the use of symmetric preconditioners for the solution of symmetric non-Hermitian complex linear systems in electromagnetics applications. The motivations are twofold, first to reflect the symmetry of the original matrix in the associated preconditioner, second to use a symmetric Krylov solver that might be cheaper than GMRES iterations; since, with an unsymmetric preconditioner, GMRES appears to be the most efficient iterative method (Carpentieri et al. 2000).

The classical *IC* preconditioner exhibits a rather poor and chaotic behaviour. It appears that this disappointing behaviour is due to the ill-conditioning of the computed factors. The use of a shift in some cases improved this situation but its effect is difficult to predict.

The classical factorized approximate inverses namely *AINV* and *FSAI*, that are also appropriate candidates, only show poor convergence behaviour. We present some clues to explain that disappointing behaviour. Although no numerical experiments are reported, we have tried both re-ordering and shift strategies but without success to improve their convergence rate.

Both $M_{Aver-Frob}$ and $M_{Sym-Frob}$ appear to be efficient and robust. Through numerical experiments, we have shown that $M_{Sym-Frob}$ was not too sensitive to

column ordering while $M_{Aver-Frob}$ is totally insensitive. In addition $M_{Aver-Frob}$ is straightforward to parallelize even though it requires more flops for its construction. It would probably be the preconditioner of choice in a parallel distributed fast multipole environment but possibilities for parallelizing $M_{Sym-Frob}$ also exist although they are more complex to implement. Finally, the major benefit of these two preconditioners is the remarkable robustness they exhibit when used in conjunction with SQMR.

References

- G. Alléon, M. Benzi, and L. Giraud. Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics. *Numerical Algorithms*, **16**, 1–15, 1997.
- A. Bendali. *Approximation par éléments finis de surface de problèmes de diffraction des ondes électro-magnétiques*. PhD thesis, Université Paris VI, 1984.
- M. W. Benson. Iterative solution of large scale linear systems. Master’s thesis, Lakehead University, Thunder Bay, Canada, 1973.
- M. W. Benson and P. O. Frederickson. Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems. *Utilitas Mathematica*, **22**, 127–140, 1982.
- M. W. Benson, J. Krettmann, and M. Wright. Parallel algorithms for the solution of certain large sparse linear systems. *Int J. of Computer Mathematics*, **16**, 1984.
- M. Benzi. *A Direct Row-projection Method for Sparse Linear Systems*. PhD thesis, Department of Mathematics, North Carolina State University, Raleigh, NC, 1993.
- M. Benzi and M. Tuma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Scientific Computing*, **19**, 968–994, 1998.
- M. Benzi and M. Tuma. Orderings for factorized approximate inverse preconditioners. *SIAM J. Scientific Computing*, **21**, 1851–1868, 2000.
- M. Benzi, C. D. Meyer, and M. Tuma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Scientific Computing*, **17**, 1135–1149, 1996.
- B. Carpentieri, I. S. Duff, and L. Giraud. Sparse pattern selection strategies for robust Frobenius-norm minimization preconditioners in electromagnetism. *Numerical Linear Algebra with Applications*, **7(7-8)**, 667–685, 2000.

- B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand. Combining fast multipole techniques and an approximate inverse preconditioner for large parallel electromagnetics calculations. Technical Report in preparation, CERFACS, Toulouse, France, 2002.
- E. Chow and Y. Saad. Experimental study of ILU preconditioners for indefinite matrices. *Journal of Computational and Applied Mathematics*, **86**, 387–414, 1997.
- E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. in ‘Proceedings 24th National Conference of the Association for Computing Machinery, Brandon Press, New Jersey’, pp. 157–172. Brandon Press, New Jersey, 1969.
- I. S. Duff and G. A. Meurant. The effect of ordering on preconditioned conjugate gradient. *BIT*, **29**, 635–657, 1989.
- V. Frayssé, L. Giraud, and S. Gratton. A set of GMRES routines for real and complex arithmetics. Tech. Rep. TR/PA/97/49, CERFACS, 1997.
- P. O. Frederickson. Fast approximate inversion of large sparse linear systems. Math. Report 7, Lakehead University, Thunder Bay, Canada, 1975.
- R. W. Freund and N. M. Nachtigal. An implementation of the QMR method based on coupled two-term recurrences. *SIAM J. Scientific Computing*, **15(2)**, 313–337, 1994.
- J. George and J. W. H. Liu. The evolution of the minimum degree ordering algorithm. *SIAM Review*, **31**, 1–19, 1989.
- L. Yu Kolotilina and A. Yu. Yeremin. Factorized sparse approximate inverse preconditionings. I: Theory. *SIAM J. Matrix Analysis and Applications*, **14**, 45–58, 1993.
- L. Yu Kolotilina and A. Yu. Yeremin. Factorized sparse approximate inverse preconditionings. II: Solution of 3D FE systems on massively parallel computers. *Int J. High Speed Computing*, **7**, 191–215, 1995.
- L. Yu Kolotilina, A. Yu. Yeremin, and A. A. Nikishin. Factorized sparse approximate inverse preconditionings. IV: Simple approaches to rising efficiency. *Numerical Linear Algebra with Applications*, **6**, 515–531, 1999.
- L. Yu Kolotilina, A. Yu. Yeremin, and A. A. Nikishin. Factorized sparse approximate inverse preconditionings. III: Iterative construction of preconditioners. *Journal of Mathematical Sciences*, **101**, 3237–3254, 2000. Originally published in Russian in *Zap. Nauchn. Semin. POMI*, 248:17-48, 1998.

- J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, **31**, 148–162, 1977.
- M. Magolu monga Made. Incomplete factorization based preconditionings for solving the Helmholtz equation. *Int. Journal for Numerical Methods in Engineering*, **50**(5), 1077–1101, 2001.
- M. Magolu monga Made, R. Beauwens, and G. Warzee. Preconditioning of discrete helmholtz operators perturbed by a diagonal complex matrix. *Communications in Numerical Methods in Engineering*, **11**, 801–817, 2000.
- A. F. Peterson, S. L. Ray, and R. Mittra. *Computational Methods for Electromagnetics*. IEEE Press, 1997.
- A. Pothen, H. D. Simon, and K. P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Analysis and Applications*, **11**(3), 430–452, 1990.
- Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, New York, 1996.
- Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Scientific and Statistical Computing*, **7**, 856–869, 1986.
- W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proc. of the IEEE*, **55**, 1801–1809, 1967.