RAL-TR-2000-009

Some sparse pattern selection strategies for robust Frobenius norm minimization preconditioners in electromagnetism[1]

Bruno Carpentieri[2], Iain S. Duff[3], Luc Giraud[4]

ABSTRACT
We consider preconditioning strategies for the iterative solution of dense complex symmetric non-Hermitian systems arising in computational electromagnetics. We consider in particular sparse approximate inverse preconditioners that use a static nonzero pattern selection. The novelty of our approach comes from using a different nonzero pattern selection for the original matrix from that for the preconditioner and from exploiting geometric or topological information from the underlying meshes instead of using methods based on the magnitude of the entries. 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. The results of our numerical experiments suggest that the new strategies are viable approaches for the solution of large-scale electromagnetic problems using preconditioned Krylov methods. In particular, our strategies are applicable when fast multipole techniques are used for the matrix-vector product on parallel distributed memory computers.

Keywords: Preconditioning techniques, Frobenius-norm minimization method, nonzero pattern selection strategies, electromagnetic scattering applications.

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

[1] Current reports available by anonymous ftp to ftp.numerical.rl.ac.uk in directory pub/reports. This report is in file cadgRAL2000009.ps.gz. Report also available through URL http://www.numerical.rl.ac.uk/reports/reports.html. Also published as Technical Report TR/PA/00/05 from CERFACS, 42 Ave G. Coriolis, 31057 Toulouse Cedex, France.

[2] carpenti@cerfacs.fr, CERFACS, 24 Ave G Coriolis, F 31057 Toulouse Cedex, France. 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).

[3] I.Duff@rl.ac.uk.

[4] giraud@cerfacs.fr, CERFACS, 24 Ave G Coriolis, F 31057 Toulouse Cedex, France.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxon OX11 0QX

March 20, 2000

Contents

1 Introduction

In recent years, there has been a significant amount of work on the simulation of electromagnetic wave propagation phenomena, addressing various topics ranging from radar cross section to electromagnetic compatibility, to absorbing materials, and antenna design. To address these problems the Maxwell equations are often solved in the frequency domain leading to singular integral equations of the first kind. The discretization by the boundary element method (BEM) results in linear systems with dense complex matrices which are very challenging to solve. 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.

In this paper, we consider the solution of linear systems of the form

$$Ax = b$$

where the coefficient matrix $A = [a_{ij}]$ is a large, dense, complex matrix of order n arising from the discretization. The coefficient matrix can be symmetric non-Hermitian in the EFIE (Electric Field Integral Equation) formulation, or unsymmetric in the CFIE (Combined Field Integral Equation) formulation. The unknowns in the vector x are associated with the edges of an underlying mesh on the surface of the object. In this paper, we will only consider numerical examples where A is symmetric because EFIE usually gives rise to linear systems that are more difficult to solve with iterative methods. The techniques considered here can be applied equally well to unsymmetric matrices. In fact in the numerical experiments we use non-symmetric solvers because the preconditioners that we construct are unsymmetric. We can, of course, construct either only the lower or only the upper part of the preconditioner and use a symmetric preconditioner obtained by reflecting this in the diagonal. One problem is that the resulting preconditioner depends on the ordering of the matrix. In previous tests [7], we investigated the effect of symmetrizing the preconditioner by averaging the off-diagonal entries after its construction and using such a symmetrized preconditioner with symmetric QMR but found that this caused a marked deterioration in the quality of the preconditioner leading to far more iterations of the iterative method. We plan to further investigate symmetric strategies in future work but, in this present study, we will stick with unsymmetric techniques and preconditioners.

Direct dense methods based on Gaussian elimination are often the method of choice because they are reliable and predictable both in terms of accuracy and cost. However, for large-scale problems, they become impractical even on large parallel platforms because they require storage of n^2 double precision complex entries of the coefficient matrix and $\mathcal{O}(n^3)$ floating-point operations to compute the factorization, where n denotes the size of the linear system. Iterative Krylov subspace based solvers are a promising alternative provided we have fast matrix-vector multiplications and robust preconditioners. There are active research efforts on multipole techniques to perform fast matrix-vector products with $\mathcal{O}(n \log(n))$ computational complexity including strategies for parallel distributed memory implementations (see [10, 11, 12]). In this paper, we focus on the other key component of Krylov methods in this context; that is, we study the design of robust preconditioning techniques.

The parallel framework suggests that sparse approximate inverses based on Frobenius-norm minimization techniques are promising candidates for the efficient preconditioning of these systems. Such techniques exhibit a good level of numerical efficiency on this class of applications when compared with the implicit approach based on incomplete factorization (see [7], [8]). The normal requirement for a good preconditioner is that it is easy to construct, cheap to store and to apply, is parallelizable and, of course, is effective in accelerating the convergence of Krylov solvers. To be computationally affordable on dense linear systems, Frobenius-norm minimization preconditioning techniques require a suitable strategy to identify the relevant entries to consider in the original matrix A, in order to define small least-squares problems, as well as an appropriate sparsity structure for the approximate inverse.

For sparse matrices, two strategies can be used to define the sparsity structure of the preconditioner. A dynamic approach constructs the nonzero pattern of the preconditioner by monitoring the residual in the least-squares problems during the computation. This is generally effective but is usually very expensive [19, 21]. A static approach that requires an *a priori* nonzero pattern for the preconditioner, introduces significant scope for parallelism and has the advantage that the memory storage requirements and computational cost for the setup phase are known in advance. However, it can be very problem dependent.

In this paper, we propose some new efficient static nonzero pattern selection strategies both for the preconditioner and for the selection of the entries of A in order to develop robust preconditioners for applications in electromagnetism. Amongst the many examples considered in [7], we select a subset of test examples, arising from both academic and industrial applications that are representative of the general numerical behaviour that we observed. More specifically, we here consider the following geometries where, for physical consistency, we have set the frequency of the wave so that there are about ten discretization points per wavelength [3]:

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 $n = 2016$; and

Example 5: a sphere, a matrix of order $n = 2430$.

We perform experiments with the following Krylov solvers:

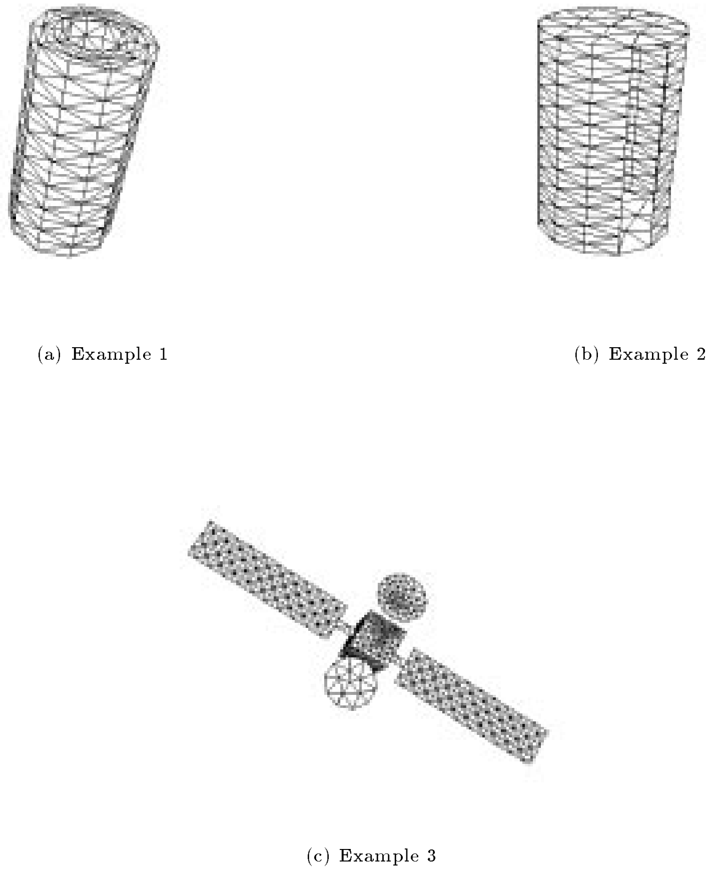
- restarted GMRES [25];
- Bi-CGSTAB [24];
- symmetric and unsymmetric QMR [18];
- TFQMR [17].

In Section 2, we describe the construction of the preconditioner using our proposed static pattern strategies and report on the associated numerical experiments. Finally, in Section 3, we present some remarks arising from the work.

2 Static pattern selection and dropping strategies

Frobenius-norm minimization is one of the most natural approaches for building explicit preconditioners. The idea is to compute the sparse approximate inverse as the matrix M which minimizes $\|I - MA\|_F$ (or $\|I - AM\|_F$ 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). In our present applications, these least-squares problems are small enough to be solved using a dense QR decomposition. The independence of these least-squares problems follows immediately from the identity:

$$\|I - MA\|_F^2 = \|I - AM^T\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_{j,*}\|_2^2 \tag{1}$$



(a) Example 1

(b) Example 2

(c) Example 3

Figure 1: Mesh associated with test examples

where e_j is the jth unit vector and $m_{j,*}$ is the column vector representing the jth row of M. In the case of right preconditioning, the analogous relation

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

holds, where $m_{*,j}$ is the column vector representing the jth column of M. Clearly, there is considerable scope for parallelism in this approach.

The main issue is the selection of the nonzero pattern of M. The idea is to keep M reasonably sparse while trying to capture the "large" entries of the inverse, which are expected to contribute the most to the quality of the preconditioner. For this purpose, two approaches can be followed: an adaptive technique that dynamically tries to identify the best structure for M; and a static technique, where the pattern of M is prescribed *a priori* based on some heuristics. Some early references to this latter class can be found in [4], [5], [6], [15] and in [1] for some applications to boundary element matrices in electromagnetism.

In addition, when the coefficient matrix is dense, the preconditioner should be constructed from a sparse approximation of A in order to reduce the computational cost of the least-squares solutions.

2.1 Strategies for the preconditioner

When the coefficient matrix has a special structure or special properties, for instance a banded matrix with a good degree of diagonal dominance or a banded *SPD* matrix, efforts have been devoted to find a pattern that can retain the entries of A^{-1} having large modulus, see [9] and [13] for example. Unfortunately, for general unstructured matrices, it is very difficult to predict a good pattern for the inverse in advance. Adaptive strategies that compute the pattern dynamically can provide very good preconditioners, even on hard problems, but at the cost of a very large amount of computing time and memory. In some cases it is possible to take advantage of special features of the underlying physical problem and compute a good *a priori* pattern for the approximate inverse.

2.1.1 Algebraic strategy

The boundary element method discretizes integral equations on the surface of the scattering object, generally introducing a very localized strong coupling among the edges in the underlying mesh. Each edge is strongly connected to only a few neighbours, while, although not null, far-away connections are much weaker. This means that a very sparse matrix can still retain the most relevant contributions from the singular integrals that give rise to dense matrices. Due to the decay of the Green's function, the inverse of A may exhibit a very similar structure to A as illustrated in Figure 2 where we display the pattern of A and A^{-1} when the smallest entries are dropped. Thus, in this case, a good pattern for the sparse approximate inverse is likely to be the nonzero pattern of a sparse approximation of A, constructed by dropping all the entries lower than a prescribed threshold, as suggested for instance in [22]. We refer to this approach as *the algebraic approach.* Several heuristics can be used to define the sparsity pattern based on the magnitude of the entries; all of them result in similar numerical behaviour [1] but some are particularly well suited for parallel implementation. In the numerical experiments, we have selected the strategy where, for each column of A, the k entries ($k \ll n$ is a positive integer) of largest modulus are retained.

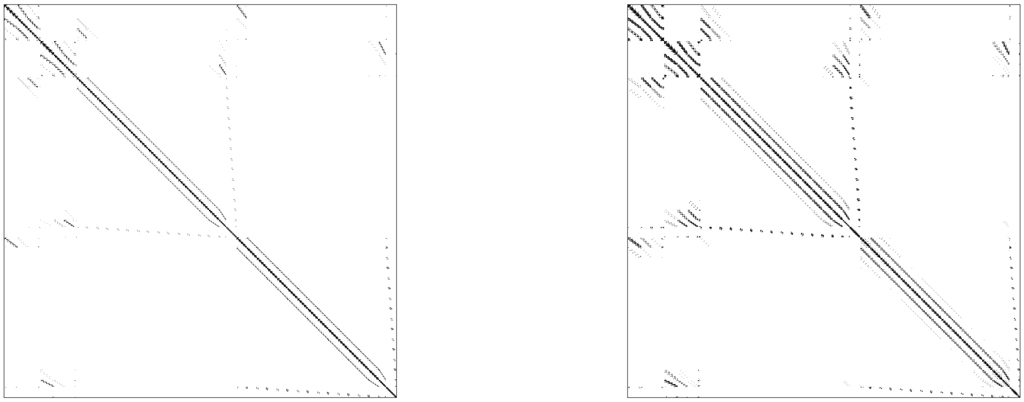

(a) $sparsified(A)$ (b) $sparsified(A^{-1})$

Figure 2: Nonzero pattern for A (left) and A^{-1} (right) when the smallest entries are discarded. The test problem is Example 1.

This strategy generally works well and competes with the approach that adaptively defines the nonzero pattern as implemented in the *SPAI* preconditioner described in [20] and [21]. Nevertheless it suffers some drawbacks that put severe limits on its use in practical applications. For large problems, accessing all the entries of the matrix A becomes too expensive or even impossible. This is the case in the fast multipole framework, where all the entries of the matrix A are not available. In addition on complex geometries, a pattern for the sparse approximate inverse computed by using information solely from A may lead to a poor preconditioner. These two main drawbacks motivate the investigation of more appropriate techniques to define a sparsity pattern for the preconditioner.

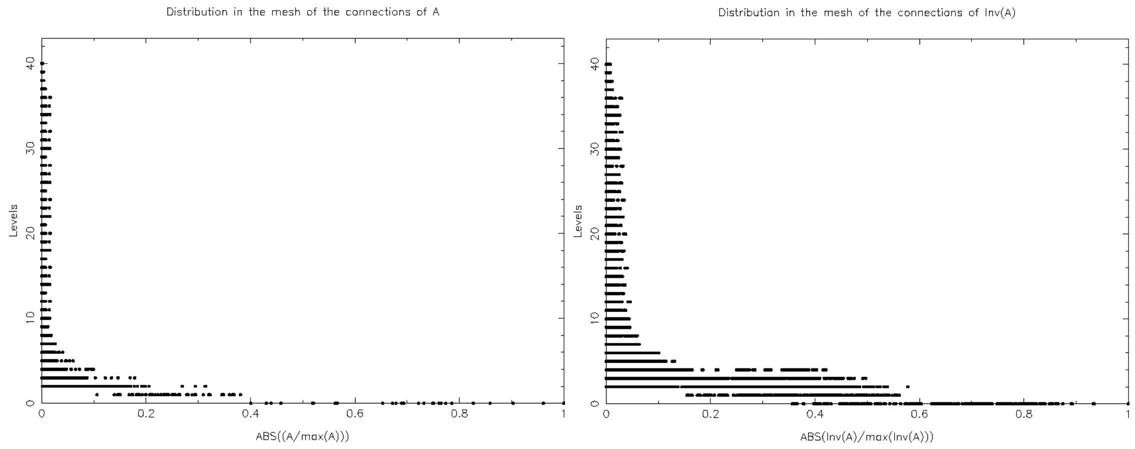
Because we work in an integral equation context, we can use more information than just the entries of the matrix of the discretized problem. In particular, we can exploit the underlying mesh and extract further relevant information to construct the preconditioner. Two types of information are available from the mesh:

the connectivity graph, describing the topological neighbourhood among the edges, and

the coordinates of the nodes in the mesh, describing geometric neighbourhoods among the edges.

2.1.2 Topological strategy

When the object geometries are smooth, only the neighbouring edges can have a strong interaction with each other, while far-away connections are generally much weaker. Thus an effective pattern for the sparse approximate inverse can be prescribed by exploiting topological information related to the near field. In the integral equation context, the surface of the object is discretized using a triangular mesh. Each degree of freedom (DOF), representing an unknown in the linear system, corresponds to an edge. The sparsity pattern for any row of the preconditioner can be defined according to the concept of level k neighbours, as introduced in [23]. Level 1 neighbours of a DOF are the DOF plus the four DOFs belonging to the two triangles that share the edge corresponding to the DOF itself. Level 2 neighbours are all the level 1 neighbours plus the DOFs in the triangles that are neighbours of the two triangles considered at level 1, and so forth. In Figure 3 we plot, for each DOF of the mesh for Example 1, the level of its neighbours with respect to the magnitude of the associated entry in A (the graph on the left) and in A^{-1} (the graph on the right). The large entries in A^{-1} derive from the interaction of a very localized set of edges in the mesh so that by retaining a few levels of neighbours for each DOF an effective preconditioner is likely to be constructed. Three levels can generally provide a good pattern for constructing an effective sparse approximate inverse. Using more levels increases the computational cost but does not improve substantially the quality of the preconditioner. We will refer to this pattern selection strategy as the *topological strategy*.



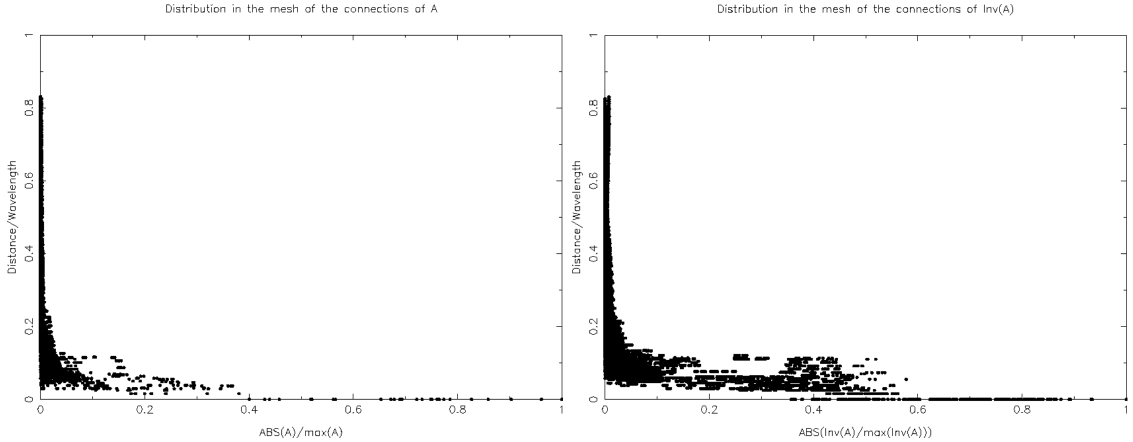
(a) Levels v.s. magnitude for A

(b) Levels v.s. magnitude for A^{-1}

Figure 3: Topological localization in the mesh for the large entries of A (left) and A^{-1} (right). The test problem is Example 1 and is representative of the general behaviour.

2.1.3 Geometric strategy

When the object geometries are not smooth, two far-away edges in the topological sense can have a strong interaction with each other so that they are strongly coupled in the inverse matrix. For the scattering problem on Example 1, we plot in Figure 4 the distance in terms of wavelength among pairs of edges in the mesh with respect to the magnitude of their associated entries in A and A^{-1}. The largest entries of A^{-1} are localized similarly to those of A, but, in many cases, small entries in A correspond to large entries in the inverse and vice-versa. This means that if we construct the sparse pattern for the inverse by only using information related to A, we may retain many small entries in the preconditioner, contributing marginally to its quality, but may neglect some of the large ones potentially damaging the quality of the preconditioner. Also, the surface of the object is very non-smooth, these large entries may come from the interaction of far-away or non-connected edges in a topological sense, which are neighbours in a geometric sense. Thus they cannot be detected by using only topological information related to the near field. Figure 4(b) suggests that we can select the pattern for the preconditioner using physical information, that is: 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. This selection strategy will be referred to as the *geometric strategy*.



(a) Distance v.s. magnitude for A

(b) Distance v.s. magnitude for A^{-1}

Figure 4: Geometric localization in the mesh for the large entries of A (left) and A^{-1} (right). The test problem is Example 1. This is representative of the general behaviour.

2.1.4 Numerical experiments

In this section, we compare the different strategies described above in the solution of our test problems.

Using the three pattern selection strategies for M, we denote by

- M_a, the preconditioner computed by using the algebraic strategy,
- M_t, the preconditioner computed by using the topological strategy,
- M_g, the preconditioner computed by using the geometric strategy,
- $SPAI$, the preconditioner constructed by using the dynamic strategy implemented by [20].

To evaluate the effectiveness of the proposed strategies, we first consider using the dense matrix A to construct the preconditioners M_a, M_t, M_g and $SPAI$. This requires the solution of large dense least-squares problems. The adaptive technique implemented in $SPAI$ computes the pattern of the preconditioner starting with a simple initial guess, like a diagonal matrix, and then improves it until a criterion of the form $\|Am_{j,*} - e_j\|_2 < \varepsilon$ (for each j) is satisfied for a given $\varepsilon > 0$, e_j being the jth column of the identity matrix, and $m_{j,*}$ being the column vector for the jth row of M according to the notation previously introduced, or until a maximum number k of nonzeros in the jth row of M has been generated (we refer the reader to [20] and [21] for further details).

The density of the preconditioner varies from one problem to another for the same value of the distance parameter chosen to define M_g. As Figure 4(b) shows, and tests on all the other examples confirm, those entries corresponding to edges contained within a sphere of radius 0.12 times the wavelength can retain many of the large entries of the inverse while giving rise to quite a sparse preconditioner. For all our numerical experiments, we choose a value for k in the construction of M_a and $SPAI$, and for the level of neighbours used to generate M_t so that they have the same density as M_g, when necessary discarding some small entries of the preconditioner so that all have the same number of entries.

For all the numerical experiments reported in this paper, for GMRES we use the implementation described in [14]. For the tests with Bi-CGSTAB, we derived a version for complex arithmetic from the Harwell Subroutine Library (HSL, [2]) routine MI06 and for those with unsymmetric QMR (referred to as UQMR in the forthcoming tables) and TFQMR, we used, respectively, the ZUCPL and ZUTFX routines available in QMRPACK [16]. The stopping criteria in all cases just consists in reducing the original residual by 10^{-5}. The symbol "-" means that convergence was not obtained after 500 iterations. In each case, we took as the initial guess $x_0 = 0$, and the right-hand side was such that the exact solution of the system was known. We performed different tests with different known solutions, observing identical results. All the numerical experiments were performed in double precision complex arithmetic on a SGI Origin 2000 and the number of iterations reported in this paper are for left preconditioning. Very similar results were obtained when preconditioning from the right.

From the results shown in Table 1, we first note that all the preconditioners accelerate the convergence of the Krylov solvers, and in some cases enable convergence when the unpreconditioned solver diverges or converges very slowly. These numerical experiments also highlight the advantages of the geometric strategy. It not only outperforms the algebraic approach and is more robust than the topological approach, which has a similar computational complexity, but it also generally outperforms the adaptive approach implemented in SPAI which is much more sophisticated and more expensive in execution time and memory. SPAI competes with M_g only on Example 1 where the density of the preconditioner is higher. This trend, namely the denser the preconditioner the more efficient SPAI is, has been observed on many other examples. However, for sparse preconditioners, SPAI may be quite poor as illustrated on Example 4, where preconditioned GMRES(30) or Bi-CGStab are slower than without a preconditioner and the iteration diverges for GMRES(10) with the SPAI preconditioner while it converges for the other three preconditioners. On the non-smooth geometry, that is Example 2, an explanation of why the geometric approach should lead to a better sparse preconditioner can be suggested by Figure 1(b). Some far-away edges in the connectivity graph, those from each side of the break, are weakly connected in the mesh but can have a strong interaction with each other, and can lead to large entries in the inverse matrix.

2.2 Strategies for the coefficient matrix

When the coefficient matrix of the linear system is dense, the construction of even a very sparse preconditioner may become too expensive in execution time as the problem size increases. Both memory and execution time are significantly reduced by replacing A with a sparse approximation. On general problems, this approach can cause a severe deterioration of the quality of the

Example 1 - Density of $M = 5.03\%$								
Precond.	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
Unprec.	-	-	-	251	202	223	231	175
M_j	-	-	465	222	174	239	210	169
M_a	219	135	96	72	72	86	107	72
M_t	100	49	36	36	36	35	42	32
M_g	124	68	46	46	46	44	58	38
SPAI	-	67	44	44	44	48	50	43

Example 2 - Density of $M = 1.59\%$								
Precond.	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
Unprec.	-	-	-	398	289	359	403	249
M_j	-	-	473	330	243	257	354	228
M_a	472	273	239	207	184	330	313	141
M_t	-	470	346	243	195	187	275	158
M_g	90	72	55	52	52	44	82	40
SPAI	-	-	99	61	61	168	97	111

Example 4 - Density of $M = 1.04\%$								
Precond.	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
Unprec.	-	224	191	158	147	177	170	118
M_j	350	211	178	153	140	188	152	110
M_a	212	157	141	132	123	131	145	115
M_t	288	187	160	146	139	145	156	98
M_g	63	51	41	41	41	37	47	32
SPAI	-	370	184	112	84	256	96	85

Table 1: Number of iterations using the preconditioners based on dense A.

preconditioner; in the BEM context, since a very sparse matrix can retain the most relevant contributions to the singular integrals, it is likely to be more effective. The use of a sparse matrix substantially reduces the size of the least-squares problems that can then be efficiently solved by direct methods.

The algebraic heuristic described in the previous sections is well suited for sparsifying A. In [1] the same nonzero sparsity pattern is selected both for A and M; in that case, especially when the pattern is very sparse, the computed preconditioner may be poor on some geometries. The effect of replacing A with its sparse approximation on some problems is highlighted in Figure 5 where we display the sparsified pattern of the inverse of the sparsified A. We see that the resulting pattern is very different from the sparsified pattern of the inverse of A shown in Figure 2.

A possible remedy is to increase the density in the patterns for both A and M. To a certain extent, we can improve the convergence, but the computational cost of generating the preconditioner grows almost cubicly with respect to density. A cheaper remedy is to choose a different number of nonzeros to construct the patterns for A and M, with less entries in the preconditioner than in the sparse approximation of A. To illustrate this effect, we report in Table 2 on the number of iterations of preconditioned GMRES(50), where the preconditioners are built by

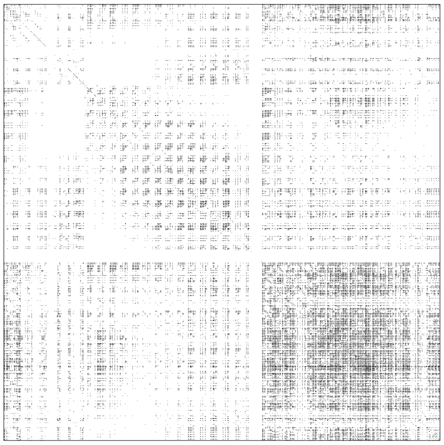


Figure 5: Sparsity pattern of the inverse of sparse A associated with Example 1. The pattern has been sparsified with the same value of threshold used to sparsify A displayed in Figure 2.

using either the same sparsity pattern for A or a two, three or five times denser pattern for A.

Example 1										
Density strategy	Percentage density of M									
	1	2	3	4	5	6	7	8	9	10
Same	-	-	299	146	68	47	47	42	37	39
2 times	-	-	248	155	76	46	40	39	39	38
3 times	-	253	207	109	49	39	39	37	35	34
5 times	-	258	213	99	48	37	38	34	33	33
Full A	364	359	144	96	46	35	35	34	32	31

Table 2: Number of iterations for GMRES(50) preconditioned with different values for the density of M using the same pattern for A and larger patterns. A geometric approach is adopted to construct the patterns. The test problem is Example 1. This is representative of the general behaviour observed.

Except when the preconditioner is very sparse, increasing the density of the pattern imposed on A for a given density of M accelerates the convergence as expected, getting quite rapidly very close to the number of iterations required when using a full A. The additional cost in terms of CPU time is negligible as can be seen in Figure 6 for experiments on Example 1. This is due to the fact that the complexity of the QR factorization used to solve the least-squares problems is the square of the number of columns times the number of rows. Thus, increasing the number of rows, that is the number of entries of A, does not penalize significantly the construction of the preconditioner. On the other hand, reducing the density of the preconditioner, that is the number of columns in the least-squares problems, can significantly reduce the overall CPU time. Notice that this observation is true for both left and right preconditioning because, according to (1) and (2) the smaller dimension of the matrices involved in the least-squares problems always corresponds to the entries of M to be computed, and the larger to the entries of the sparsified matrix from A.

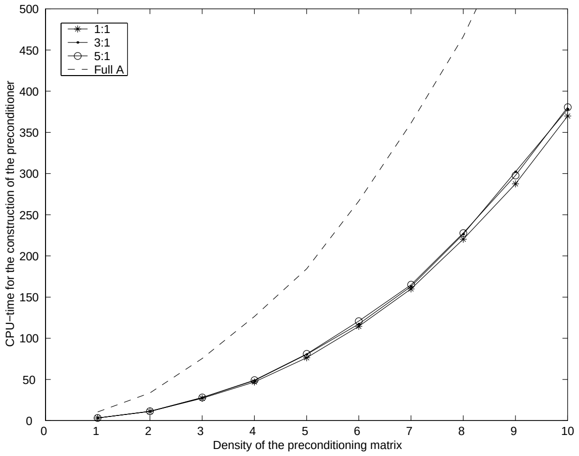


Figure 6: CPU time for the construction of the preconditioner using a different number of nonzeros in the patterns for A and M. The test problem is Example 1. This is representative of the other examples.

2.2.1 Numerical experiments

We report in this section on the numerical results obtained by replacing A with its sparse approximation in the construction of the preconditioner. In Table 3 we use the following notation:

- M_{a-a}, introduced in [1] and computed by using algebraic information from A. The same pattern is used for the preconditioner;
- M_{a-t}, constructed by using the algebraic strategy to sparsify A and the topological strategy to prescribe the pattern for the preconditioner;
- M_{a-g}, constructed by using the geometric approach and an algebraic heuristic for A with the same density as for the preconditioner;
- M_{2a-t}, similar to M_{a-t}, but the density of the pattern imposed on A is twice as dense as that imposed M_{a-t};
- M_{2a-g}, similar to M_{a-g} but, as in the previous case, the density of the pattern imposed on A is twice as dense as that imposed on M_{a-g}.

For the sake of comparison we also report the number of iterations without using a preconditioner and with only a diagonal scaling, denoted by M_j.

Other combinations are possible for defining the selection strategies for the patterns of A and M. Here we focus on the most promising ones, that use information from the mesh to retain the large entries of the inverse, and the algebraic strategy for A to capture the most relevant contributions to the singular integrals. We also consider the preconditioner M_{a-a} to compare with previous tests [1], that were performed on different geometries from those considered here. We show, in Table 3, the results of our numerical experiments. For each example, we give the number of iterations required by each preconditioned solver, and the CPU time to construct the preconditioner when the least-squares problems are solved using LAPACK routines. The CPU time for constructing M_{a-t} and M_{2a-t} is in some cases much larger than that needed to setup M_{a-g} and M_{2a-g}. The reason is that, in the topological strategy, it is not possible to prescribe exactly a value for the density. Thus, for each problem, we select a suitable number of levels of neighbours, to obtain the closest number of nonzeros to that retained in the pattern based on the geometric approach. After the construction of the preconditioner, we drop its smallest entries to ensure an identical number of nonzeros for the two strategies.

Example 1 - Density of $M = 5.03\%$

Precond.	CPU Time	GMRES(m)					Bi - CGStab	UQMR	TFQMR
		m=10	m=30	m=50	m=80	m=110			
Unprec.	0.00	-	-	-	251	202	223	231	175
M_j	0.00	-	-	465	222	174	239	210	169
M_{a-a}	83.42	284	170	138	114	92	120	156	94
M_{a-t}	91.07	179	61	45	45	45	43	58	36
M_{a-g}	79.47	147	93	68	59	59	55	73	53
M_{2a-t}	91.78	128	56	40	40	40	37	50	36
M_{2a-g}	80.18	131	79	52	51	51	59	65	44

Example 2 - Density of $M = 1.59\%$

Precond.	CPU Time	GMRES(m)					Bi - CGStab	UQMR	TFQMR
		m=10	m=30	m=50	m=80	m=110			
Unprec.	0.00	-	-	-	398	289	359	403	249
M_j	0.00	-	-	473	330	243	257	354	228
M_{a-a}	13.98	-	319	255	221	203	181	319	135
M_{a-t}	16.45	-	261	213	174	169	128	251	121
M_{a-g}	13.53	251	178	150	138	117	106	256	116
M_{2a-t}	16.73	-	370	284	202	182	176	276	127
M_{2a-g}	13.67	100	73	61	55	55	48	93	40

Example 3 - Density of $M = 2.35\%$

Precond.	CPU Time	GMRES(m)					Bi - CGStab	UQMR	TFQMR
		m=10	m=30	m=50	m=80	m=110			
Unprec.	0.00	-	-	-	-	488	-	444	308
M_j	0.00	-	-	-	491	427	375	356	306
M_{a-a}	82.59	436	316	240	193	125	144	166	135
M_{a-t}	146.44	137	108	93	71	71	64	93	66
M_{a-g}	109.45	-	464	296	203	108	240	166	144
M_{2a-t}	147.79	113	78	59	53	53	41	61	44
M_{2a-g}	110.30	122	84	72	59	59	53	67	50

Example 4 - Density of $M = 1.04\%$

Precond.	CPU Time	GMRES(m)					Bi - CGStab	UQMR	TFQMR
		m=10	m=30	m=50	m=80	m=110			
Unprec.	0.00	-	224	191	158	147	177	170	118
M_j	0.00	350	211	178	153	140	188	152	110
M_{a-a}	31.75	299	205	172	146	133	162	180	103
M_{a-t}	38.05	266	152	130	114	99	92	127	83
M_{a-g}	31.12	81	67	66	63	63	39	79	41
M_{2a-t}	38.23	269	167	143	136	116	107	137	93
M_{2a-g}	31.24	71	60	47	47	47	43	61	41

Continued on next page

									Continued from previous page
Example 5 - Density of $M = 0.63\%$									
Precond.	CPU Time	GMRES(m)					Bi - CGStab	UQMR	TFQMR
		m=10	m=30	m=50	m=80	m=110			
Unprec.	0.00	-	344	233	146	125	152	170	109
M_j	0.00	-	326	219	140	131	183	173	107
M_{a-a}	27.66	-	352	249	154	134	202	183	107
M_{a-t}	70.93	360	66	64	60	60	34	76	46
M_{a-g}	26.04	313	81	68	61	61	36	74	40
M_{2a-t}	71.29	71	48	47	47	47	25	54	30
M_{2a-g}	26.13	88	42	39	39	39	21	45	25

Table 3: Number of iterations to solve the set of test problems.

We first observe that using a sparse approximation of A reduces the convergence rate of the preconditioned iterations when the nonzero pattern imposed on the preconditioner is very sparse. However if we adopt the geometric strategy to define the sparsity pattern for the approximate inverse, the convergence rate is not affected very much. For even larger values of density, the difference in the number of iterations between using full A or an algebraic sparse approximation becomes negligible. For all the experiments, M_{a-g} still outperforms M_{a-a} and is generally more robust than M_{a-t}; the most efficient and robust preconditioner is M_{2a-g}. The multiple density strategy allows us to improve the efficiency and the robustness of the Frobenius-norm preconditioner on this class of problems without requiring any more time for the construction of the preconditioner. For all the test examples, it enables us to get the fastest convergence even for GMRES with a low restart parameter on problems where neither M_{a-a} nor M_{a-g} converge.

The effectiveness of this multiple density heuristic is illustrated in Figure 7 where we see the effect of preconditioning on the clustering of the eigenvalues of A for the most difficult problem, Example 2. The eigenvalues of the preconditioned matrices are in both cases well clustered around 1 (with a more effective clustering for M_{2a-g}), but those obtained by using the multiple density strategy are further from the origin. This is highly desirable when trying improve the convergence of Krylov solvers.

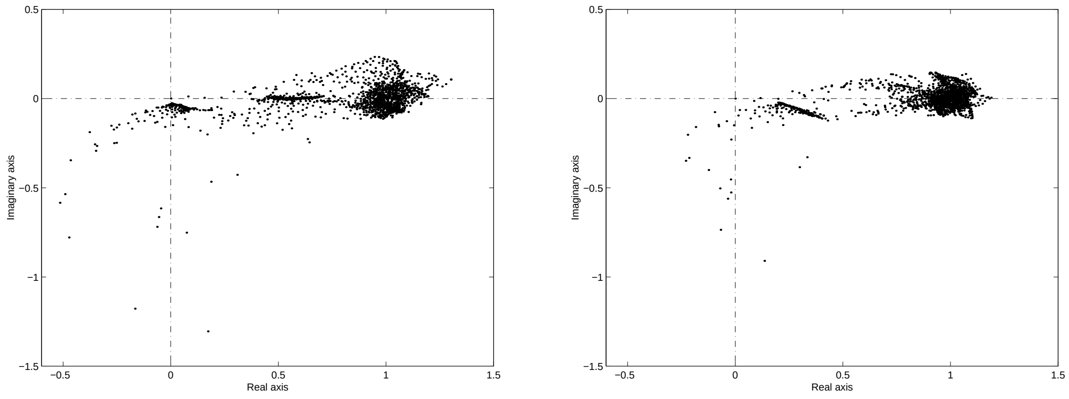


Figure 7: Eigenvalue distribution for the coefficient matrix preconditioned by using a single (on the left) and a multiple (on the right) density strategy on Example 2.

Another advantage of this multiple density heuristic is that it generally allows us to reduce the density of the preconditioner (and thus its construction cost), while preserving its numerical quality. Although no specific results are reported to illustrate this aspect, this behaviour may be partially observed in Table 2.

Example 1 - Density of $M = 5.03\%$								
Precond.	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
M_j	-	-	465	222	174	239	210	169
$SSOR$	-	-	216	136	98	147	177	135
$ILU(0)$	-	-	-	-	-	-	479	-
$SPAI$	-	-	192	68	68	150	83	94
SLU	160	53	38	38	38	46	50	39
M_{2a-g}	131	79	52	51	51	59	65	44
Example 2 - Density of $M = 1.59\%$								
Precond.	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
M_j	-	-	473	330	243	257	354	228
$SSOR$	-	413	245	164	134	185	281	266
$ILU(0)$	-	-	-	-	322	385	394	439
$SPAI$	-	-	-	-	-	-	-	-
SLU	-	-	-	-	282	-	-	-
M_{2a-g}	100	73	61	55	55	48	93	40

Table 4: Number of iterations with some classical preconditioners computed using sparse A (algebraic).

Finally, to assess the performance of the proposed Frobenius-norm minimization approach described in this paper, we show, in Table 4, the numerical results observed on Examples 1 and 2 with some classical preconditioners, of both explicit and implicit form. These are: diagonal scaling, SSOR, ILU(0) and SPAI applied to a sparse approximation of A constructed using the algebraic approach. The method referred to as SLU in that table uses the sparsified matrix A as an implicit preconditioner; that is, the sparsified matrix is factorized using ME47, a sparse direct solver from HSL, and those exact factors are used as the preconditioner. Thus it represents an extreme case with respect to $ILU(0)$, since a complete fill-in is allowed in the factors. This approach, although not easily parallelizable, is generally quite effective on this class of applications for dense enough sparse approximations of A. However, as shown in this table, when the preconditioner is very sparse, the numerical quality of this approach deteriorates and the Frobenius-norm minimization method is more robust. All these preconditioners, except SLU on Example 1, exhibit much poorer acceleration capabilities than that provided by M_{2a-g}. If we reduce the density of the preconditioner in Example 1, M_{2a-g} converges slowly but becomes the most efficient. It should also be noted that SPAI works reasonably well when computed using dense A (see Table 1) but with sparse A it does not converge on Example 2 (see Table 4).

3 Conclusions

We have presented some *a priori* pattern selection strategies for the construction of a robust sparse Frobenius-norm minimization preconditioner for electromagnetic scattering problems expressed in integral formulation. We have shown that, by using additional geometric information from the underlying mesh, it is possible to construct robust sparse preconditioners at an affordable computational and memory cost. The topological strategy requires less computational effort to construct the pattern, but since the density is a step function of the number of levels, the construction of the preconditioner can require some additional computation. Also it may not handle very well complex geometries where some parts of the object are not connected, as in Example 3 (see Figure 1(c)). By retaining two different densities in the patterns of A and M we can decrease very much the computational cost for the construction of the preconditioner, usually a bottleneck for this family of methods; preserving the efficiency while increasing the robustness of the resulting preconditioner. The numerical experiments have shown that, using this pattern selection strategy, we can compute a very sparse but effective preconditioner. With the same low density, none of the classical preconditioners that we considered can compete with it. An additional major feature of this pattern selection strategy is that it does not require access to all the entries of the matrix A, so that it is promising for an implementation in a fast multipole setting where A is not directly available but where only the near field entries are computed.

M_{2g-g}

Example	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
1	165	103	75	60	60	66	71	61
2	145	110	95	76	76	68	140	64
3	129	89	70	57	57	49	69	52
4	71	57	48	48	48	38	52	34
5	110	46	42	42	42	24	50	27

Table 5: Number of iterations to solve the set of test models by using a multiple density geometric strategy to construct the preconditioner. The pattern imposed on M is twice as dense as that imposed on A.

M_{2t-g}

Example	GMRES(m)					Bi - CGStab	UQMR	TFQMR
	m=10	m=30	m=50	m=80	m=110			
1	197	87	49	49	49	50	66	50
2	103	82	72	61	61	49	111	50
3	143	98	84	60	60	56	70	53
4	70	58	49	49	49	39	65	37
5	143	50	47	47	47	29	57	28

Table 6: Number of iterations to solve the set of test models by using a topological strategy to sparsify A and a geometric strategy for the preconditioner. The pattern imposed on M is twice as dense as that imposed on A.

The geometric approach can be also used to sparsify A, without noticeably deteriorating the quality of the preconditioner. This is showed in Table 5, where M_{2g-g} is constructed by exploiting geometric information in the patterns of both A and M, but chosing twice as dense a pattern for A as for M. As suggested by Figure 3(a), due to the strongly localized coupling introduced by the discretization of the integral equations, the topological approach can also provide a good sparse approximation of A, by retaining just a few levels of neighbouring edges for each DOF in the mesh. The numerical behaviour of this approach is illustrated in Table 6. In both cases the resulting preconditioner is still robust and better suited for a fast multipole framework since it does not require knowledge of the location of the largest entries in A.

Acknowledgments

We would like to thank the Computational Electromagnetism project at CERFACS for providing us with the code and the meshes to generate the matrices for the test examples.

References

[1] 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.

[2] Anon. *Harwell Subroutine Library. A Catalogue of Subroutines (Release 12).* Theoretical Studies Department, AEA Industrial Technology, 1993.

[3] A. Bendali. *Approximation par elements finis de surface de problemes de diffraction des ondes electro-magnetiques.* PhD thesis, *Université Paris VI* , 1984.

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

[5] 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.

[6] M.W. Benson, J. Krettmann, and M. Wright. Parallel algorithms for the solution of certain large sparse linear systems. *Int J. of Computer Mathematics*, 16:245–260, 1984.

[7] B. Carpentieri, I.S. Duff, and L.Giraud. Sparse preconditioning of dense problems from electromagnetic applications. Technical Report TR/PA/00/04, CERFACS, Toulouse, France, 1999.

[8] Ke Chen. On a class of preconditioning methods for dense linear systems from boundary elements. *SIAM J. Scientific Computing*, 20(2):684–698, 1998.

[9] E. Chow. A priori sparsity patterns for parallel sparse approximate inverse preconditioners. Tech. Rep. UCRL-JC-130719, Lawrence Livermore National Laboratory, Livermore, CA, 1998. To appear in SISC.

[10] B. Dembart and M.A. Epton. Low frequency multipole translation theory for the Helmholtz equation. Tech. Rep. SSGTECH-98-013, The Boeing Company, Seattle, WA, 1998.

[11] B. Dembart and M.A. Epton. Spherical harmonic analysis and synthesis for the fast multipole method. Tech. Rep. SSGTECH-98-014, The Boeing Company, Seattle, WA, 1998.

[12] B. Dembart and E. Yip. Matrix assembly in FMM-MOM codes. Tech. Rep. ISSTECH-97-002, The Boeing Company, Seattle, WA, 1997.

[13] S. Demko, W.F. Moss, and P.W. Smith. Decay rates for inverses of band matrices. *Mathematics of Computation*, 43:491–499, 1984.

[14] 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.

[15] P.O. Frederickson. Fast approximate inversion of large sparse linear systems. Math. Report 7, Lakehead University, Thunder Bay, Canada, 1975.

[16] R. W. Freund and Nachtigal. QMRPACK: a package of QMR algorithms. Technical Report, 1994. To appear in ACM Transactions on Mathematical Software.

[17] R.W. Freund. A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems. *SIAM J. Scientific Computing*, 14(2):470–482, 1993.

[18] R.W. Freund and N.M. Nachtigal. QMR: a quasi-minimal residual method for non-hermitian linear systems. *Numerische Mathematik*, 60(3):315–339, 1991.

[19] N.I.M. Gould and J.A. Scott. On approximate-inverse preconditioners. Tech. Rep. 95-026, RAL, 1995.

[20] N.I.M. Gould and J.A. Scott. Sparse approximate-inverse preconditioners using norm-minimization techniques. *SIAM J. Scientific Computing*, 19(2):605–625, 1998.

[21] M. Grote and T. Huckle. Parallel preconditionings with sparse approximate inverses. *SIAM J. Scientific Computing*, 18:838–853, 1997.

[22] L.Yu. Kolotilina. Explicit preconditioning of systems of linear algebraic equations with dense matrices. *J.Sov.Math.*, 43:2566–2573, 1988. English translation of a paper first published in Zapisli Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo im. V.A. Steklova AN SSSR 154 (1986) 90-100.

[23] J. Rahola. Experiments on iterative methods and the fast multipole method in electromagnetic scattering calculations. Technical Report TR/PA/98/49, CERFACS, Toulouse, France, 1998.

[24] H.A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Scientific and Statistical Computing*, 13:631–644, 1992.

[25] 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.