SYMMETRIC INVERSE-BASED MULTILEVEL ILU PRE-CONDITIONING FOR SOLVING DENSE COMPLEX NON-HERMITIAN SYSTEMS IN ELECTROMAGNETICS

B. Carpentieri[1, *] and M. Bollhöfer[2]

[1]Institute of Mathematics and Computing Science, University of Groningen Nijenborgh, P. O. Box 407, 9700 AK Groningen, Netherlands

[2]Institute of Computational Mathematics, Technische Universität Braunschweig, Germany

Abstract—Boundary element discretizations of exterior Maxwell problems lead to dense complex non-Hermitian systems of linear equations that are difficult to solve from a linear algebra point of view. We show that the recently developed class of inverse-based multilevel incomplete LU factorization has very good potential to precondition these systems effectively. This family of algorithms can produce numerically stable factorizations and exploits efficiently the possible symmetry of the underlying integral formulation. The results are highlighted by calculating the radar-cross-section of a full aircraft, and by a numerical comparison against other standard preconditioners.

1. INTRODUCTION

The numerical solution of Maxwell's equations in large unbounded domains may be efficiently carried out using the boundary element method, which reformulates the Maxwell's equations as a set of integral equations defined only on the surface of the scattering object and gives rise to a system of linear equations. Consider a perfectly conducting object Ω with boundary Γ (see Figure 1), illuminated by an incident plane wave $(\vec{E}_{inc}, \vec{H}_{inc})$ of angular frequency $\omega = ck = 2\pi c/\lambda$, where we denote by c the speed of light, k is the wavenumber and $\lambda = c/f$ is the wavelength (f is the frequency). The scattering problem in the unbounded region Ω_e may be described in the frequency domain by
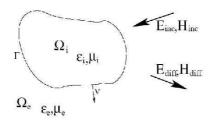
Figure 1. The standard electromagnetic scattering problem, considered in our study.

the following variational formulation: find the surface current $\vec{j}$ such that for all tangential test functions $\vec{j}^{\,t}$, we have

$$\int_\Gamma \int_\Gamma G(|y-x|) \left(\vec{j}(x) \cdot \vec{j}^{\,t}(y) - \frac{1}{k^2} div_\Gamma \vec{j}(x) \cdot div_\Gamma \vec{j}^{\,t}(y) \right) dx dy$$

$$= \frac{i}{kZ_0} \int_\Gamma \vec{E}_{inc}(x) \cdot \vec{j}^{\,t}(x) dx. \tag{1}$$

In Eq. (1), known as Electric Field Integral Equation (or shortly EFIE), we denote by $G(|y-x|) = \dfrac{e^{ik|y-x|}}{4\pi |y-x|}$ the Green's function and by $Z_0 = \sqrt{\mu_0/\varepsilon_0}$ the characteristic impedance of vacuum (ϵ is the electric permittivity and μ the magnetic permeability).

Equation (1) is not the only possible integral formulation for this problem. For closed targets, the Magnetic Field Integral Equation (MFIE) can be used, which reads

$$\int_\Gamma \left(\vec{R}_{ext}\, j \wedge \vec{\nu} \right) \cdot \vec{j}^{\,t} + \frac{1}{2} \int_\Gamma \vec{j} \cdot \vec{j}^{\,t} = - \int_\Gamma \left(\vec{H}_{inc} \wedge \vec{\nu} \right) \cdot \vec{j}^{\,t}.$$

The operator $\vec{R}_{ext}j$ is defined as

$$\vec{R}_{ext}\, j(y) = \int_\Gamma \vec{grad}_y G(|y-x|) \wedge \vec{j}(x) dx,$$

and is evaluated in the domain exterior to the object. Both the EFIE and MFIE formulations suffer from interior resonances which make the numerical solution more problematic at resonant frequencies. The numerical instabilities due to resonance problems can be solved by combining linearly EFIE and MFIE. The resulting integral equation is known as Combined Field Integral Equation (CFIE) and it may be considered the formulation of choice for modelling closed targets.

On discretizing Eq. (1) in space by using the Method of Moments (MoM) over a mesh containing n edges, the surface current $\vec{j}$ is

expanded into a set of basis functions $\{\vec{\varphi}_i\}_{1 \leq i \leq n}$ with compact support (the Rao-Wilton-Glisson basis [34], is a popular choice), and then the integral equation is applied to a set of tangential test functions $\vec{j}^t$. Selecting $\vec{j}^t = \vec{\varphi}_j$, we are led to compute the set of coefficients $\{\lambda_i\}_{1 \leq i \leq n}$ such that

$$\sum_{1 \leq i \leq n} \lambda_i \left[\int_\Gamma \int_\Gamma G(|y-x|) \left(\vec{\varphi}_i(x) \cdot \vec{\varphi}_j(x) - \frac{1}{k^2} div_\Gamma \vec{\varphi}_i(x) \cdot div_\Gamma \vec{\varphi}_j(y) \right) dxdy \right]$$

$$= \frac{i}{kZ_0} \int_\Gamma \vec{E}_{inc}(x) \cdot \vec{\varphi}_j(x) dx, \tag{2}$$

for each $1 \leq i \leq n$. System (2) can be recast in matrix form as

$$A\lambda = b, \tag{3}$$

where $A = [A_{ij}]$ and $b = [b_i]$ have elements, respectively,

$$A_{ij} = \int_\Gamma \int_\Gamma G\left(|y-x|\right) \left(\vec{\varphi}_i(x) \cdot \vec{\varphi}_j(y) - \frac{1}{k^2} div_\Gamma \vec{\varphi}_i(x) \cdot div_\Gamma \vec{\varphi}_j(y) \right) dxdy,$$

$$b_j = \frac{i}{kZ_0} \int_\Gamma \vec{E}_{inc}(x) \cdot \vec{\varphi}_j(y) dx.$$

Each unknown $\{\lambda_i\}$ is associated with the vectorial flux across the ith edge in the mesh. Vector b varies with the frequency and the direction of the illuminating wave. Matrix A is dense complex symmetric non-Hermitian, and has size $n \times n$ for metallic bodies and $2n \times 2n$ for dielectric bodies (in this case, at each node are associated one electric and one magnetic current). Notice that the number of edges n increases linearly with the geometry of the object and quadratically with the frequency of the illuminating radiation, leading to very large dense systems to be solved at high frequency. We like to remark that nowadays MoM-based solvers are very popular techniques in Electromagnetics, see e.g., [15, 28, 29, 38, 41] for some recent studies.

Direct methods, based on variants of the Gaussian elimination algorithm, can solve system (3) in $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ floating-point operations. They are feasible to use only for solving small size linear systems, even when parallel platforms are considered. On the other hand, iterative methods may solve the bottlenecks of memory of direct methods by exploiting the structure of A both in the search of a good preconditioner and in the matrix-vector (M-V) multiply. Krylov subspace methods solve system (3) in $\alpha \cdot N_{\text{iter}} \cdot \mathcal{O}$ (M-V) flops, where α is a constant which depends on the implementation of the specific iterative algorithm, N_{iter} is the number of iterations to attain convergence to a given user-defined accuracy, and $\mathcal{O}$ (M-V) is the number of operations required for one M-V product. In

recent years, a significant amount of work has been devoted to design fast parallel algorithms that can reduce the $\mathcal{O}\left(n^2\right)$ computational complexity for the M-V product with boundary element equations, like the Fast Multipole Method (FMM) by V. Rokhlin [35], the $\mathcal{H}$-matrix approach by W. Hackbush [25], the Adaptive Cross Approximation by M. Bebendorf [4], and other approaches. Since the pioneering work by Rokhlin and his co-authors, the Fast Multipole Algorithm continues to receive considerable attention in Electromagnetics, see e.g., [20, 21, 31, 32, 39].

Two-level implementations of FMM have $\mathcal{O}\left(n^{3/2}\right)$ complexity, which reduces to $\mathcal{O}\left(n^{4/3}\right)$ using three levels, and to $\mathcal{O}\left(n \log n\right)$ in the multilevel implementation of the Multilevel Fast Multipole Algorithm (MLFMA). The number of iterations of Krylov methods may vary significantly depending on the choice of the integral formulation as well as on the characteristics of the geometry and of the materials. For closed targets, the CFIE formulation typically converges in $\mathcal{O}\left(n^{0.25}\right)$ iterations. The EFIE formulation is more tough to solve; the condition number of the pertinent linear system may grow like the square root of the size of the scatterer in terms of the wavelength, and linearly with the number of points per wavelength [14]. On EFIE, the number of iterations of Krylov subspace methods scale as $\mathcal{O}\left(n^{0.5}\right)$ and preconditioning is crucial to accelerate the convergence.

In this paper, we address preconditioning techniques for this problem class. We analyse the recently developed class of inverse-based multilevel incomplete LU (ILU) factorization algorithms [5], and we apply them to the solution of dense complex symmetric non-Hermitian linear systems arising from the boundary element discretization of Maxwell's equations in exterior domains. After describing shortly, in Section 2, earlier work on preconditioning for boundary integral equations, in Section 3 we present the inverse-based multilevel incomplete LU factorization. In Section 4, we report on our experiments of realistic radar-cross section calculations, also in comparison to other popular methods used for this problem class.

2. PRECONDITIONING BOUNDARY INTEGRAL EQUATIONS

The construction of robust and economic preconditioners for system (3) may be challenging. Operator-dependent techniques, which attempt to use information from the underlying physical problem (see e.g., [3, 8, 40]), may be optimal in some sense for one integral equation but there is little guarantee that one method working well for one problem, works well for another. Algebraic methods use

only information contained in the coefficient matrix of the linear system; they may be far from optimal for any problem, but they can be applied to different operators and geometries by tuning a few parameters [9, 24, 33], and they can be adapted from existing numerical software.

In our study, we follow a purely algebraic approach. We consider splitting the discretized operator A in the form $A = S + B$, where S is a sparse matrix retaining the most relevant contributions to the singular integrals in Eq. (3), and is easy to invert, while B can be dense. For Fredholm-type boundary integral equations, if the continuous operator $\mathcal{S}$ underlying S is bounded and the operator $\mathcal{B}$ underlying B is compact, then $\mathcal{S}^{-1}\mathcal{B}$ is compact and

$$\mathcal{S}^{-1}\left(\mathcal{S} + \mathcal{B}\right) = \mathcal{I} + \mathcal{S}^{-1}\mathcal{B}$$

so that we may expect that the preconditioned system $(I + S^{-1}B)x = S^{-1}b$ has a good clusterization of eigenvalues close to one (see e.g., [13, pp. 182–185]). The local matrix S may be computed by dropping the small entries of A below a threshold or, when all the entries of A are not available, by using information from the physical mesh. The fast exponential decay of the discrete Green's function induces strong local coupling amongst the currents in the mesh, so that retaining two or three levels of neighbours per grid node may suffice to compute a good sparse approximation S to A. Comparative experiments revealed that pattern selection strategies based on matrix- and mesh-based approaches can both provide very good approximations for moderately low sparsity ratio, between 1% and 2% [10].

The mesh-based approach is to be preferred in combination with multipole techniques, in particular with MLFMA. Multipole algorithms partition the object into small cubes, whose length is a fraction of the wavelength of the incident radiation. From a linear algebra point of view, they yield a matrix decomposition of the form

$$A = A_{\mathrm{diag}} + A_{\mathrm{near}} + A_{\mathrm{far}}, \tag{4}$$

where A_{diag} is the *exact* sparse block diagonal part of matrix A, which is associated with interactions of basis functions belonging to the same cube, A_{near} is the *exact* sparse block near-diagonal part of A, associated with interactions within one level of neighboring cubes (they are 8 in 2D and 26 in 3D), and A_{far} is the *approximate* far-field part of A. In a multipole setting, the matrix $A_{\mathrm{diag}} + A_{\mathrm{near}}$ is a suitable choice for S [11].

Sparse preconditioners approximately factorize or invert S. Clearly, they may maintain the $\mathcal{O}(n \log n)$ memory and algorithmic cost of the multipole method. However, the compact support of S

allows exchanges only of the near-field mesh information. This means that on some applications it may be necessary to introduce some mechanism in the construction of the preconditioner to recover the far-field information from the discrete Green's function, enhancing the solver scalability. As these mechanisms always require the inversion of the near-field matrix at some stage of the iterative solution (see e.g., [11, 24, 30]), in this study we consider efficient matrix solvers for S.

2.1. Earlier Work

Many techniques that have proved successful for the field of partial differential equations, have been adopted for integral equations. However, while there may be general lessons to be learned from results in other areas, it is still not clear which solution technology is better to use. In recent years, considerable interest have received methods that compute a sparse approximate inverse M of the coefficient matrix by minimizing the quantity, for $A = S$,

$$\|I - AM\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_{\bullet j}\|_2^2, \tag{5}$$

where e_j is the jth canonical unit vector, $m_{\bullet j}$ is the column vector representing the jth column of M, see e.g., [11, 24, 33]. In the case of left preconditioning, because S is symmetric, the analogous relation

$$\|I - MA\|_F^2 = \|I - AM^T\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_{j\bullet}\|_2^2 \tag{6}$$

may be used, where $m_{j\bullet}$ is the column vector representing the jth row of M. The sparse approximate inverse, referred to as SPAI, can be applied at each step of an iterative solver by performing a sparse M-V product.

Owing to the decay of the discrete Green's function, the inverse of A has a very similar structure to A with many small entries far from the diagonal, as it is visible in Figure 2 for the case of a sphere. The discrete Green's function can be considered as a row or as a column of the exact inverse depicted on the physical computational grid. A good pattern for the approximate inverse is the nonzero pattern of a sparse approximation to A (see Figure 2(a)), which corresponds to the pattern of the near-field multipole matrix in the multipole context. The pattern can be computed and applied in advance, before calculating the entries of M.

On the other hand, disappointing results are reported with factorized approximate inverses for this problem class. From Figure 3,
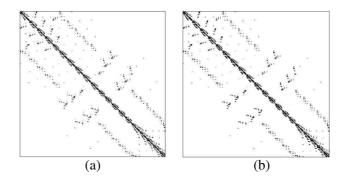
Figure 2. Structure of A (on the left) and A^{-1} (on the right) after scaling the matrix and discarding all the entries of relative magnitude smaller than 5.0×10^{-2}. The test problem is a small sphere. (a) Pattern of the large entries of A. (b) Pattern of the large entries of A^{-1}.
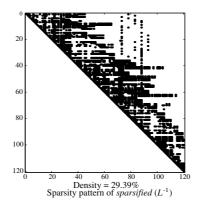

Density = 29.39%
Sparsity pattern of *sparsified* (L^{-1})

Figure 3. Sparsity patterns of the inverse factor L^{-1} of A after dropping all the entries of relative magnitude smaller than 5.0×10^{-2}. The test problem is a small sphere.

we see that the inverse factors of A can be totally unstructured and any dropping strategy, either computed in advance or dynamically updated during the construction of M, may be very difficult to tune as it can easily discard relevant information and potentially lead to a very poor preconditioner.

Due to indefineteness of the coefficient matrix, the important class of standard incomplete LU (ILU) factorization may fail because of the presence of small pivots and/or may produce ill-conditioned L and U factors [10, 30]. For the CFIE formulation, ILU preconditioners

proved to be effective. Selecting the nonzero pattern of the lower/upper triangular part of S for the factors may deliver rates of convergence comparable to a complete LU factorization, and decidedly faster than the block Jacobi preconditioners [30]. On the EFIE formulation, the factors L and U are often ill-conditioned and the triangular solves are unstable. By shifting the eigenvalues of the coefficient matrix along the imaginary axis may sometimes help to compute a more stable preconditioner. However, the shift is not easy to tune *a priori* and its effect on the convergence is difficult to predict [10]. Pivoting may yield better results [30]; in this case, the ith row of the triangular factor is computed as long as $permtol \cdot |s_{ij}| > |s_{ii}|$, where $permtol$ is the permutation tolerance and s_{ij} are the entries of S. However, pivoting may completely destroy the system symmetry.

3. INVERSE-BASED MULTILEVEL ILU FACTORIZATION PRECONDITIONER

In this study, we investigate symmetric multilevel preconditioners based on an incomplete LDL^T factorization for solving system (3). Following [5, 7], we initially rescale and reorder the initial matrix $A = S$ as

$$P^T DADP = \hat{A} \qquad (7)$$

which yields $\hat{A}\hat{x} = \hat{b}$ for appropriate $\hat{x}$, $\hat{b}$. Since the given system matrix A is assumed to be complex symmetric, we employ symmetric maximum weight matchings [17, 18] by default prior to any fill-reducing ordering. i.e., using symmetric maximum weight matchings we first rescale A such that $\tilde{A} = DAD$ has entries $|\tilde{a}_{ij}| \leq 1$ whilst $|\tilde{a}_{i,\pi_i}| = |\tilde{a}_{\pi_i,i}| = 1$ for a suitable permutation π. Following [18] one can easily construct a permutation matrix P_0 such that the entries $\tilde{a}_{i,\pi_i}, \tilde{a}_{\pi_i,i}$ are at least located in the tridiagonal part of $P_0^T \, DAD \, P_0$ having many 2×2 blocks

$$\begin{pmatrix} a_{ii} & a_{i,i+1} \\ a_{i+1,i} & a_{i+1,i+1} \end{pmatrix}$$

such that $|a_{i,i+1}|=|a_{i+1,i}| = 1$, 1×1 blocks a_{ii} such that $|a_{ii}| = 1$, while in exceptional cases also 1×1 blocks with $a_{ii} = 0$ may occur. For details we refer to [18]. This typically improves the block diagonal dominance of the system significantly. After preprocessing A such that P_0DADP_0 consists of an improved block diagonal part, a symmetric reordering is applied to reduce the fill-in bandwidth. The symmetric reordering is applied to the compressed graph of P_0DADP_0, i.e., block rows and block columns associated with a 2×2 block are replaced by a single row and column having the union of the block column/row

pattern as nonzero pattern. This leads to a block-structure-preserving permutation P_1 and the total reordered system is $P^T D A D P$, where $P = P_0 P_1$. We point out that these scalings and reorderings maintain the symmetry of A. Next, an inverse-based ILU with static block diagonal pivoting is computed where the block diagonal pivots are either 1×1 or 2×2 pivots depending on which of the two choices yields more block diagonal dominance. More precisely, during the approximate incomplete factorization $\hat{A} \approx LDL^T$ such that L is unit lower triangular and D is block diagonal, the norm $\|L^{-1}\|$ is estimated and if at factorization step l a prescribed bound is exceeded, the current row l and column l are permuted to the lower right end of the matrix. Otherwise the approximate factorization is continued. One single pass leads to an approximate partial factorization

$$
\Pi^T \hat{A} \Pi = \begin{pmatrix} B & E^T \\ E & C \end{pmatrix} \approx \begin{pmatrix} L_B & 0 \\ L_E & I \end{pmatrix} \begin{pmatrix} D_B & 0 \\ 0 & S_C \end{pmatrix} \begin{pmatrix} L_B^T & L_E^T \\ 0 & I \end{pmatrix}
$$
$$
\equiv L_1 D_1 L_1^T, \tag{8}
$$

with a suitable leading block B and a suitable permutation matrix, where $\|L^{-1}\| \leq \kappa$. The remaining system S_C approximates $C - EB^{-1}E^T$ and from the relations

$$
\begin{cases} B\hat{x}_1 + E^T \hat{x}_2 = \hat{b}_1 \\ E\hat{x}_1 + C\hat{x}_2 = \hat{b}_2 \end{cases} \Rightarrow \begin{cases} \hat{x}_1 = B^{-1} \left(\hat{b}_1 - E^T \hat{x}_2 \right) \\ \left(C - EB^{-1}E^T \right) \hat{x}_2 = \hat{b}_2 - EB^{-1}\hat{b}_1 \end{cases},
$$

at each step of an iterative solver we need to store and invert only blocks with B and $S_C \approx C - EB^{-1}E^T$ while for reasons of memory efficiency, L_E is discarded and implicitly represented via $L_E \approx E U_B^{-1} D_B^{-1}$. When the scaling, preordering and the factorization is successively applied to S_C, a multilevel variant of (7) is computed. After a one additional level we obtain

$$
\tilde{P}^T \tilde{D} A \tilde{D} \tilde{P} = \left[\begin{array}{cc|c} B & E_1^T & E_2^T \\ E_1 & C_{11} & C_{21}^T \\ \hline E_2 & C_{21} & C_{22} \end{array} \right]
$$

$$
\approx \left[\begin{array}{cc|c} L_B & 0 & 0 \\ L_{E_1} & I & 0 \\ \hline L_{E_2} & L_{C_{21}} & I \end{array} \right] \left[\begin{array}{cc|c} D_B & 0 & 0 \\ 0 & D_{C_{11}} & 0 \\ \hline 0 & 0 & S_{22} \end{array} \right] \left[\begin{array}{cc|c} L_B & L_{E_1}^T & L_{E_2}^T \\ 0 & I & L_{C_{21}}^T \\ \hline 0 & 0 & I \end{array} \right].
$$

The multilevel algorithm ends at some step m when either S_C is factored completely or it becomes considerably dense and switches

to a dense LAPACK solver [2]. We also like to point out that intermediate Schur complements S_C can be discarded as soon as the Schur complement matrix S_{22} of the next level is computed. After computing an m-step ILU decomposition, for preconditioning we have to apply $L_m^{-1} A L_m^{-T}$. From the error equation $E_m = A - L_m D_m L_m^T$, we see that $\|L_m^{-1}\|$ contributes to the inverse error $L_m^{-1} E_m L_m^{-T}$. Monitoring the growth of this quantity during the partial factorization is essential to preserve the numerical stability of the solver. Furthermore, at least for real symmetric matrices, eigenvalue inclusion bounds for the eigenvalues near the origin of S_C and A can be shown [7]. It can be shown that eigenvalues μ of S_C near zero are bounded by their counterparts λ of A and $\kappa^2 \lambda$. This in turn means that eigenvalues with small modulus are inherited by S_C, at least when dropping is avoided. An important feature of the algorithm is that it preserves explicitly the symmetry of the coefficient matrix, enabling in this case the use of symmetric Krylov methods such as the simplified QMR method for complex symmetric matrices [22].

4. EXPERIMENTS

In our experiments, we selected linear systems arising from radar-cross-section (RCS) calculation of realistic targets, modelled using the EFIE formulation, that is Eq. (1). As discussed in Section 1, the pertinent systems are dense complex symmetric non-Hermitian. The advantages of this formulation are numerous. In particular, it does not require any hypothesis on the geometry of the object, which makes it the model of choice for solving problems with cavities. We should nevertheless mention that, for closed geometries, also the CFIE can be used. The pertinent linear systems typically have a more favourable eigenvalue distribution, and can be solved efficiently by, e.g., the block Jacobi or the additive Schwarz preconditioners. Because the linear systems arising from the CFIE model are not challenging from a linear algebra point of view, we do not consider them further in this study.

We show the characteristics of the test cases in Table 1. The selected problems are representative of realistic RCS calculations in industry. Although not very large in absolute sense, their solution demanded considerable resources on our computers. Storing the coefficient matrix of the linear system for the Airbus A318 problem required approximately 18 GB of RAM. The solution of larger problems would necessitate the use of multipole techniques for the M-V products and a parallel implementation of the multilevel ILU algorithm, which are beyond the scope of this paper and will be subject of future analysis. In our study, we carried out dense M-V products at each iteration using

the LAPACK library [2]. We declared convergence if the initial residual was reduced by 12 orders of magnitude within 3000 iterations. All the experiments were run on one node of the Millipede cluster facility located at University of Groningen. Each node features 12 Opteron 2.6 GHz cores and 24 GB of physical RAM. The codes are compiled in Fortran using the `gfortran` compiler version 9.

We computed the preconditioner from a sparse approximation matrix S to the dense coefficient matrix A, as described in Section 2. The matrix S was extracted from A by selecting the p largest entries per row in A. We chose a value for the parameter p that returns a number of nonzeros in S approximately equal to that of the multipole operator for the same problem. For the multilevel ILU method, we used the algorithmic implementation available in the ILUPACK package [6] developed by the second author. In all our experiments, we used four level of recursive factorization, we prescribed the bounds $\|L_1^{-1}\| \leq 100$, $\|U_1^{-1}\| \leq 100$ for the inverse factors, and we set $t = 10^{-2}$ for the threshold parameter that is applied for dropping the small entries in the triangular factors. On the tough aircraft problem, we had to use $t = 10^{-3}$ to achieve convergence.

We combined the inverse-based multilevel incomplete LU factorization with three Krylov methods, that are the restarted GMRES method [37], the recently developed CORS method [12] and the SQMR method [22]. We summarise the relative costs of the three solvers in Table 3. Restarted GMRES method is virtually always used for solving non-Hermitian dense linear systems, as it can significantly reduce the expensive re-orthogonalization costs of the GMRES algorithm and sometimes result in considerably faster simulation time [11]. The CORS iterative solver is a non-optimal Krylov subspace method developed from a variant of the non-symmetric Lanczos procedure, and is based on cheap three-term vector recurrences. In the experiments reported in [26], it turned out to be the

Table 1. Characteristics of the model problems.

Ex.	Mesh description	Size/Memory (Gb)	Frequency (MHz)	$\kappa_1(A)$
1	Satellite	1699/0.1	57	$1 \cdot \mathcal{O}(10^5)$
2	Cube	7200/1.7	249	$2 \cdot \mathcal{O}(10^5)$
3	Open cylinder	6268/1.3	362	$1 \cdot \mathcal{O}(10^5)$
4	Sphere	12000/4.6	535	$6 \cdot \mathcal{O}(10^5)$
5	Airbus A318	23676/18.0	800	$1 \cdot \mathcal{O}(10^7)$

Table 2. Experiments with inverse-based multilevel ILU preconditioner.

prec	mem	time	GMRES (50) its	GMRES (50) time	CORS its	CORS time	SQMR its	SQMR time
Ex.1 – density (S) = 1.69% - ordering: *metise*								
none	–	–	+3000	+101.99	742	44.99	809	24.14
MILUT($t=10^{-2}$)	6.0	0.47	78	2.74	60	3.36	70	2.37
Ex.2 – density (S) = 0.47% – ordering: *amd*								
none	–	–	+3000	+1862.99	1133	1032.76	1453	697.62
MILUT($t=10^{-2}$)	4.6	7.81	430	236.57	145	164.69	155	115.66
Ex.3 – density (S) = 0.57% – ordering: *amd*								
none	–	–	+3000	1361.71	1683	1299.71	1947	698.02
MILUT($t=10^{-2}$)	4.2	4.33	926	369.83	176	138.03	168	62.05
Ex.4 – density (S) = 0.65% – ordering: *amd*								
none	–	–	+3000	+4891.44	809	2570.93	999	1260.70
MILUT($t=10^{-2}$)	5.2	64.12	274	461.49	128	469.02	185	472.48
Ex.5 – density (S) = 2.73% – ordering: *pq*								
none	–	–	+3000	+15278.08	+3000	+27287.82	+3000	+13723.47
MILUT($t=10^{-3}$)	6.8	753.50	98	906.59	69	802.67	70	480.46

fastest non-optimal Krylov solver for this problem class. The SQMR method is also developed upon short-term vector recurrences like CORS, but in addition it can exploit the symmetry of the coefficient matrix A, provided a symmetric preconditioner is used. Indeed the inverse-based multilevel ILU method preserves symmetry, provided the given matrix is exactly symmetric. Furthermore, in this case only one of the factors L and U is computed and symmetry is fully exploited, i.e., the preconditioner is equivalent to some kind of multilevel incomplete Cholesky decomposition. The problem of having an indefinite system is resolved using symmetric maximum weight matchings and pivots of size 1×1, 2×2.

The results of Table 2 show that the proposed preconditioner was very effective to reduce the number of iterations of all the three Krylov solvers, and highlight the good performance of the SQMR method. In our runs, SQMR was competitive with GMRES for large restarts and was always faster than CORS. Exploiting the symmetry

Table 3. Algorithmic cost and memory expenses of the implementation of Krylov algorithms that are used for the experiments. We denote by n the problem size and by i the iteration number.

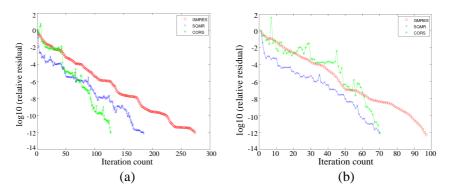| Solver	Type	Products by A	Products by A^H	Memory
CORS	non-Hermitian	2	–	$\text{matrix} + 14n$
GMRES	"	1	–	$\text{matrix} + (i+5)n$
SQMR	complex symmetric	1	–	$\text{matrix} + 10n$

Figure 4. Convergence histories on (a) example 4 and (b) example 5.

of the linear system not only enabled us to halve memory costs in the factorization, but also enhanced the iterative solution. The result on the Airbus aircraft is remarkable: the three solvers converged in less than one hundred iterations and the computation of a moderately sparse factorization took around ten minutes on one core. We were not able to achieve a comparable result with any other algebraic preconditioner that we tested. In our experiments, we selected a very low tolerance for the residual reduction in the stopping criterion, that is 10^{-12}. In practical electromagnetics calculations, a tolerance of $\mathcal{O}(10^{-3})$ may enable the correct reconstruction of the RCS of realistic aircrafts [11]. We can see from the convergence plots depicted in Figure 4, that in our run this level of residual reduction could be obtained after only few iterations of SQMR.

In combination with the multilevel ILU preconditioner, we tested the performance of different orderings available in ILUPACK [6], that are Approximate Minimum Degree (*amd*) [1], Metis Multilevel Nested Dissection by nodes (*metisn*) and edges (*metise*) [27], Reverse Cuthill-McKee (*rcm*) [16], Minimum Degree (*mmd*) [23], and the nonsymmetric ddPQ (*pq*) [36] orderings. Indeed we observed some

Table 4. Experiments with different orderings for the multilevel ILU factorization preconditioner.

ordering	Ex.1		Ex.2		Ex.3	
	mem	its	mem	its	mem	its
amd	6.0	132	4.6	155	4.2	168
mmd	5.6	213	4.4	158	4.1	178
rcm	7.2	69	4.1	175	4.6	168
metisn	6.2	163	3.7	170	3.9	167
metise	6.0	70	3.8	189	4.0	167
pq	6.0	132	4.6	155	4.2	168

minor difference of performance, and these are shown in Table 4, but they were problem-dependent. We cannot draw a sound conclusion on the best ordering for this problem class; in Table 2, we report the best results for each geometry.

In Table 5, we compare the multilevel ILU factorization preconditioner with the SPAI method, which computes the matrix M that minimizes, for $A = S$, $\|I - MA\|_F$ (or $\|I - AM\|_F$ for right preconditioning) subject to certain sparsity constraints. We imposed on M the sparse pattern of S. For fair comparison, in these experiments we set the maximum number of nonzeros in the triangular factors approximately equal to the density of S. In addition to the original SPAI algorithm with prescribed pattern, we consider a symmetric variant that computes only the lower triangular part of M, including the diagonal, and reflects the calculated nonzeros with respect to the diagonal. More precisely, in the computation of the kth 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:

$$\min \left\| \tilde{e}_j - \hat{A}\tilde{m}_{\bullet j} \right\|_2^2 \tag{9}$$

where $\hat{e}_j = e_j - \sum_{k<j} \hat{a}_{\bullet k} m_{kj}$, $\tilde{m} = \{m_{kj}\}_{k \geq j}$ and $\hat{A} = A(\bullet, k)$, for all $k < j$. We refer to this preconditioner as $M_{Sym\text{-}SPAI}$ in Table 5. The preconditioner built using this approach only computes $(m+n)/2$ nonzeros where we denote by m the number of nonzeros entries in S. However, it does not any more minimize any Frobenius norm. We notice again the remarkable robustness of the ILU preconditioner, that was much cheaper to construct compared to SPAI and delivered good

Table 5. Comparative results with the SPAI preconditioner. The parameter p denotes the maximum number of nonzeros computed in each column of the triangular factors L, U and of the SPAI preconditioner.

Ex.	p	MILUT ($10^{-2}, p$) + SQMR			SPAI + GMRES			$M_{Sym-SPAI}$ + SQMR		
		Setup	It	CPU	Setup	It	CPU	Setup	It	CPU
1	50	0.23	213	5.97	28.36	366	9.25	26.43	809	18.41
2	60	1.47	248	104.59	627.46	760	346.36	754.22	370	158.58
3	60	1.54	1586	627.45	705.45	2315	802.00	645.82	+3000	1050.24

Table 6. Experiments with shifted incomplete Cholesky algorithm with level of fill $IC(\ell)$.

Satellite				
level-of-fill	density	shift = 0	shift = 0.001	shift = 0.1
0	3.968	2295	2765	1592
1	14.809	1294	1764	+3000
2	28.567	62	64	270
cube7200				
level-of-fill	density	shift = 0	shift = 0.001	shift = 0.1
0	0.928	+3000	+3000	+3000
1	2.348	777	826	1050
2	4.127	637	637	1014
CYLINDER				
level-of-fill	density	shift = 0	shift = 0.001	shift = 0.1
0	1.114	+3000	+3000	+3000
1	3.152	+3000	+3000	+3000
2	5.141	+3000	+3000	+3000

rates of convergence except on example 3, which is tough and needed more nonzeros in the preconditioners.

In our experiments, the multilevel ILU algorithm was decidedly more robust than other one-level ILU factorization algorithms. In Table 6, we show experiments with an incomplete Cholesky factorization with sparsity pattern based on levels of fill, denoted as
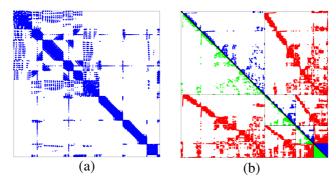
(a)   (b)

Figure 5. Sparsity patterns of S and of the multilevel ILU preconditioner (using four levels) for the satellite problem. (a) Pattern of S. (b) Pattern of ILU.

$IC(\ell)$. We also attempted to overcome the possible ill-conditioning of the triangular factors computed by $IC(\ell)$ by applying, before computing the factorization, a complex diagonal matrix perturbation to S. More specifically, we used

$$S_\tau = S + \mathbf{i}\tau h \Delta_r, \tag{10}$$

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

$$h = n^{-\frac{1}{d}} \quad \text{with} \quad d = 3 \text{ (the space dimension)}. \tag{11}$$

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 6, we report the number of SQMR iterations for different values of shift τ and various level of fill-in ℓ. In our set of problems, standard one-level algorithms were not competitive against the multilevel ILUPACK solver. Finally, in Figure 5 we display for the satellite problem the sparsity patterns of S and of the triangular factors computed by the ILUPACK preconditioner; we clearly see that the two patterns do not resemble very much, differently from the case of SPAI.

5. SOME CONCLUSIONS

We have shown by numerical experiments that the recently developed class of inverse-based multilevel incomplete LU factorization methods can effectively precondition highly indefinite boundary element matrices arising in electromagnetic scattering applications. They produce well-conditioned factorizations, have moderate construction

cost, are simple to combine with multipole techniques, and can exploit any symmetry of the integral formulation enabling the use of fast convergent symmetric Krylov methods. The multilevel nature of these algorithms has good potential for parallelism and offers a favourable trade-off between memory costs and performance for this problem class. Some algebraic preconditioners are nonsymmetric by construction and cannot be natively used with symmetric Krylov methods. And even in the case a symmetric preconditioner is available, due to round-off errors the multipole operator may sometimes loose the theoretical symmetry of the system [19]. However, if your matrix is exactly (complex), using a robust symmetric preconditioner may give a significant improvement in both performance and cost.

REFERENCES

1. Amestoy, P., T. A. Davis, and I. S. Duff, "Algorithm 837: AMD, an approximate minimum degree ordering algorithm," *ACM Transactions on Mathematical Software*, Vol. 30, No. 3, 381–388, 2004.

2. Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

3. Andriulli, F. P., K. Cools, H. Bagci, F. Olyslager, A. Buffa, S. Christiansen, and E. Michielssen, "A multiplicative calderon preconditioner for the electric field integral equation," *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 8, 2398–2412, Aug. 2008.

4. Bebendorf, M., "Approximation of boundary element matrices," *Numerische Mathematik*, Vol. 86, No. 4, 565–589, 2000.

5. Bollhöfer, M. and Y. Saad, "Multilevel preconditioners constructed from inverse-based ILUs," *SIAM J. Scientific Computing*, Vol. 27, No. 5, 1627–1650, 2006.

6. Bollhöfer, M., Y. Saad, and O. Schenk, ILUPACK — Preconditioning software package, Jun. 2011, http://ilupack.tu-bs.de/.Release 2.4.

7. Bollhöfer, M., Marcus J. Grote, and O. Schenk. "Algebraic multi-level preconditioner for the Helmholtz equation in heterogeneous media," *SIAM J. Scientific Computing*, Vol. 31, No. 5, 3781–3805, 2009.

8. Bruno, O., T. Elling, R. Paffenroth, and C. Turc, "Electro-

magnetic integral equations requiring small numbers of krylov-subspace iterations," *J. Comput. Phys.*, Vol. 228, 6169–6183, Sep. 2009.

9. Carpentieri, B., "Algebraic preconditioners for the fast multipole method in electromagnetic scattering analysis from large structures: Trends and problems," *Electronic Journal of Boundary Element*, Vol. 7, No. 1, 13–49, 2009.

10. Carpentieri, B., I. S. Duff, L. Giraud, and M. Magolu Monga Made, "Sparse symmetric preconditioners for dense linear systems in electromagnetism," *Numerical Linear Algebra with Applications*, Vol. 11, Nos. 8–9, 753–771, 2004.

11. Carpentieri, B., I. S. Duff, L. Giraud, and G. Sylvand, "Combining fast multipole techniques and an approximate inverse preconditioner for large electromagnetism calculations," *SIAM J. Scientific Computing*, Vol. 27, No. 3, 774–792, 2005.

12. Carpentieri, B., Y.-F. Jing, and T.-Z. Huang, "The BiCOR and CORS algorithms for solving nonsymmetric linear systems," *SIAM J. Scientific Computing*, Vol. 33, No. 5, 3020–3036, 2011.

13. Chen, K., *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, 2005.

14. Chew, W. C. and K. F. Warnick, "On the spectrum of the electric field integral equation and the convergence of the moment method," *Int J. Numerical Methods in Engineering*, Vol. 51, 475–489, 2001.

15. Cui, Z., Y. Han, and M. Li, "Solution of CFIE-JMCFIE using parallel MoM for scattering by dielectrically coated conducting bodies," *Journal of Electromagnetic Waves and Applications*, Vol. 25, No. 2–3, 211–222, 2011.

16. Cuthill, E. and J. McKee, "Reducing the bandwidth of sparse symmetric matrices," *Proc. 24th National Conference of the Association for Computing Machinery*, 157–172, Brandon Press, New Jersey, 1969.

17. Duff, I. S. and J. Koster, "The design and use of algorithms for permuting large entries to the diagonal of sparse matrices," *SIAM J. Matrix Analysis and Applications*, Vol. 20, No. 4, 889–901, 1999.

18. Duff, I. S. and S. Pralet, "Strategies for scaling and pivoting for sparse symmetric indefinite problems," *SIAM J. Matrix Analysis and Applications*, Vol. 27, No. 2, 313–340, 2005.

19. Durdos, R., "Krylov solvers for large symmetric dense complex linear systems in electromagnetism: Some numerical experiments," *Working Notes WN/PA/02/97*, CERFACS, Toulouse, France,

2002.

20. Ergül, Ö and L. Gürel, "Efficient solutions of metamaterial problems using a low-frequency Multilevel Fast Multipole Algorithm," *Progress In Electromagnetics Research*, Vol. 108, 81–99, 2010.

21. Ergül, Ö, T. Malas, and L. Gürel, "Solutions of large-scale electromagnetics problems using an iterative inner-outer scheme with ordinary and approximate Multilevel Fast Multipole Algorithms," *Progress In Electromagnetics Research*, Vol. 106, 203–223, 2010.

22. Freund, R. W., "Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices," *SIAM J. Scientific and Statistical Computing*, Vol. 13, No. 1, 425–448, 1992.

23. George, J. and J. W. H. Liu, "The evolution of the minimum degree ordering algorithm," *SIAM Review*, Vol. 31, 1–19, 1989.

24. Gürel, L. and T. Malas, "Iterative near-field preconditioner for the Multilevel Fast Multipole Algorithm," *SIAM J. Scientific Computing*, Vol. 32, 1929–1949, 2010.

25. Hackbush, W., "A sparse matrix arithmetic based on $\mathcal{H}$-matrices," *Computing*, Vol. 62, No. 2, 89–108, 1999.

26. Jing, Y.-F., B. Carpentieri, and T.-Z. Huang, "Experiments with Lanczos biconjugate a-orthonormalization methods for MoM discretizations of Maxwell's equations," *Progress In Electromagnetics Research*, Vol. 99, 427–451, 2009.

27. Karypis, G. and V. Kumar, "Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices version 4.0," http://glaros.dtc.umn.edu/gkhome/views/metis, University of Minnesota, Department of Computer Science/Army HPC Research Center Minneapolis, MN 55455, 1998.

28. Lai, B., H.-B. Yuan, and C.-H. Liang, "Analysis of Nurbs surfaces modeled geometries with higher-order MoM based AIM," *Journal of Electromagnetic Waves and Applications*, Vol. 25, No. 5–6, 683–691, 2011.

29. Lim, H. and N.-H. Myung, "A novel hybrid Aipo-MoM technique for jet engine modulation analysis," *Progress In Electromagnetics Research*, Vol. 104, 85–97, 2010.

30. Malas, T. and L. Gürel, "Incomplete LU preconditioning with Multilevel Fast Multipole Algorithm for electromagnetic scattering," *SIAM J. Scientific Computing*, Vol. 29, No. 4, 1476–1494, 2007.

31. Pan, X.-M., L. Cai, and X.-Q. Sheng, "An efficient high order Multilevel Fast Multipole Algorithm for electromagnetic scattering analysis," *Progress In Electromagnetics Research*, Vol. 126, 85–100, 2012.

32. Pan, X.-M., W.-C. Pi, and X.-Q. Sheng, "On OpenMP parallelization of the Multilevel Fast Multipole Algorithm," *Progress In Electromagnetics Research*, Vol. 112, 199–213, 2011.

33. Pan, X. M. and X. Q. Sheng, "An efficient parallel SAI preconditioner for multilevel fast multipole algorithm for scattering by extremely large complex targets," *Int. Conf. Microw. Millim. Wave. Tech.*, 407–410, 2009.

34. Rao, S. M., D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. on Antennas and Propagat.*, Vol. 30, 409–418, 1982.

35. Rokhlin, V., "Rapid solution of integral equations of scattering theory in two dimensions," *J. Comp. Phys.*, Vol. 86, No. 2, 414–439, 1990.

36. Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd Edition, SIAM Publications, 2003.

37. Saad, Y. and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Scientific and Statistical Computing*, Vol. 7, 856–869, 1986.

38. Su, J., X.-W. Xu, and B. Hu, "Hybrid PMM-MoM method for the analysis of finite periodic structures," *Journal of Electromagnetic Waves and Applications*, Vol. 25, No. 2–3, 267–282, 2011.

39. Wang, W. and N. Nishimura, "Calculation of shape derivatives with periodic Fast Multipole Method with application to shape optimization of metamaterials," *Progress In Electromagnetics Research*, Vol. 127, 49–64, 2012.

40. Yan, S., J.-M. Jin, and Z. Nie, "Calderon preconditioning techniques for integral equation based methods," *URSI International Symposium on Electromagnetic Theory (EMTS)*, 130–133, Aug. 2010.

41. Zhao, X.-W., Y. Zhang, H.-W. Zhang, D. Garcia-Donoro, S.-W. Ting, T. K. Sarkar, and C.-H. Liang, "Parallel MoM-PO method with out-of-core technique for analysis of complex arrays on electrically large platforms," *Progress In Electromagnetics Research*, Vol. 108, 1–21, 2010.