

Algebraic preconditioners for the Fast Multipole Method in electromagnetic scattering analysis from large structures: trends and problems

Bruno Carpentieri

CRS4 Bioinformatics Laboratory,
Edificio 3, Loc. Piscinamanna, 09010 Pula (CA), Italy.

Abstract

The Fast Multipole Method was introduced by Greengard and Rokhlin in a seminal paper appeared in 1987 for studying large systems of particle interactions with reduced algorithmic and memory complexity [60]. Developments of the original idea are successfully applied to the analysis of many scientific and engineering problems of practical interest. In scattering analysis, multipole techniques may enable to reduce the computational complexity of iterative solution procedures involving dense matrices arising from the discretization of integral operators from $O(n^2)$ to $O(n \log n)$ arithmetic operations. In this paper we discuss recent algorithmic developments of algebraic preconditioning techniques for the Fast Multipole Method for 2D and 3D scattering problems. We focus on design aspects, implementation details, numerical scalability, parallel performance on emerging computer systems, and give some minor emphasis to theoretical aspects as well. Thanks to the use of iterative techniques and efficient parallel preconditioners, fast integral solvers involving tens of million unknowns are nowadays feasible and can be integrated in the design processes.

Keywords: algebraic preconditioners, Fast Multipole Method, Krylov solvers, electromagnetic scattering applications, Maxwell's equations.

1 Introduction

An efficient solution of the Maxwell's equations is a critical component of the simulation of many realistic industrial processes like the Radar Cross Section (RCS) calculation of arbitrarily shaped electrically large objects like aircrafts, the analysis of electromagnetic (EM) compatibility of electrical devices with their environment, the design of absorbing materials, radars, antennas and many others. Scattering problems address the physical issue of computing the diffraction pattern of the EM radiation that is propagated by a complex body illuminated by an incident wave. Objects of interest in real-life applications can have large electrical size compared to the wavelength so that the numerical solution becomes extremely demanding for large computer resources and fast numerical algorithms. Until the emergence of high-performance computing in the early eighties, scattering analysis was affordable only by using approximate

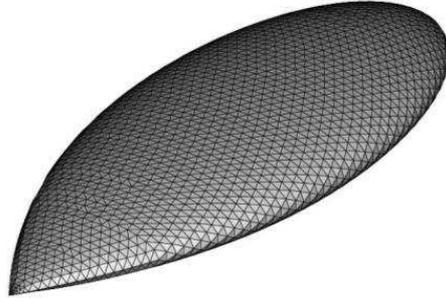


Figure 1: Example of surface discretization in an integral equation context. Each unknown of the problem is associated to an edge in the mesh. The geometry represents the ogive of a missile and has size 2.5 m. Courtesy of EADS-CCR Toulouse.

high frequency techniques such as the shooting and bouncing ray method (SBR) [87]. Basically, ray-based asymptotic methods like SBR and uniform theory of diffraction are based on the idea that EM scattering becomes a localized phenomenon as the size of the scatterer increases with respect to the wavelength. In the last few decades, thanks to the impressive advances in computer technology and the introduction of innovative algorithms with limited computational and memory requirement, a rigorous numerical solution has become possible for many practical applications. Differential equation solvers are a popular solution approach for EM problems because of the sparsity structure of the pertinent matrices arising from the discretization. Finite-difference (FD) [80, 119], finite-element (FE) [110, 124], or finite-volume (FV) methods [16, 17] can be used to discretize the Maxwell's equations into a volume surrounding the scatterer and generate a sparse linear system of equations. Prior to the discretization phase, the domain outside the object is truncated by imposing an artificial boundary that simulates an infinite volume, e.g. [12, 61, 93, 73]. An alternative approach based on integral equation methods solves for the electric and the magnetic currents induced on the surface of the object. Integral methods require a simple description of the surface of the target by means of triangular facets (see an example of discretization in Figure 1) so that a 3D problem is reduced to solving a 2D surface problem, simplifying considerably the mesh generation especially in the case of moving objects. Surface discretizations may limit the effect of grid dispersion errors which occur when a wave assumes a different phase velocity on the grid compared to the exact solution. Grid dispersion errors accumulate in space and may introduce spurious solutions over large 3D simulation regions [7, 74, 86]. For second-order accurate differential schemes, to alleviate the problem the grid density may grow up to $\mathcal{O}((kd)^3)$ unknowns in 2D and of $\mathcal{O}((kd)^{4.5})$ in 3D where k is the wavenumber and d is the approximate diameter of the simulation region, increasing considerably the overall solution cost also for practical (*i.e.* finite) values of wavenumber [36].

In an integral equation context, the standard EM scattering problem can be formulated in variational form as follows:

Find the surface current \vec{j} such that for all tangential test functions \vec{j}^t , we have

$$\int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{j}(x) \cdot \vec{j}^t(y) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{j}(x) \cdot \operatorname{div}_{\Gamma} \vec{j}^t(y) \right) dx dy = \frac{i}{kZ_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{j}^t(x) dx. \quad (1)$$

Eqn. (1) is called Electric Field Integral Equation (EFIE); we denote by $G(|y-x|) = \frac{e^{ik|y-x|}}{4\pi|y-x|}$ the Green's function of Helmholtz equation, Γ is the boundary of the object, k the wave number and $Z_0 = \sqrt{\mu_0/\epsilon_0}$ the characteristic impedance of vacuum (ϵ is the electric permittivity and μ the magnetic permeability). Given a continuously differentiable vector field $\vec{j}(x)$ represented in Cartesian coordinates on a 3D Euclidean space as $\vec{j}(x) = j_{x_1}(x_1, x_2, x_3)\vec{e}_{x_1} + j_{x_2}(x_1, x_2, x_3)\vec{e}_{x_2} + j_{x_3}(x_1, x_2, x_3)\vec{e}_{x_3}$, where $\vec{e}_{x_1}, \vec{e}_{x_2}, \vec{e}_{x_3}$ are the unit basis vectors of the Euclidean space, we denote by $\operatorname{div} \vec{j}(x)$ the divergence operator defined as

$$\operatorname{div} \vec{j}(x) = \frac{\partial j_{x_1}}{\partial x_1} + \frac{\partial j_{x_2}}{\partial x_2} + \frac{\partial j_{x_3}}{\partial x_3}.$$

The EFIE formulation can be applied to arbitrary geometries such as those with cavities, disconnected parts, breaks on the surface, and is very popular in industry, see e.g. [14, 88]. For closed targets, the Magnetic Field Integral Equation (MFIE) can be used, which reads

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

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 formulations suffer from interior resonances which make the numerical solution more problematic at some frequencies known as resonant frequencies, especially for large objects. The problem can be solved by combining linearly EFIE and MFIE. The resulting integral equation, known as Combined Field Integral Equation (CFIE), is the formulation of choice for closed targets.

On discretizing Eqn. (1) in space by 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 [100] is a popular choice), 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

$$\begin{aligned} \sum_{1 \leq i \leq n} \lambda_i \left[\int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{\varphi}_i(x) \cdot \vec{\varphi}_j(x) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{\varphi}_i(x) \cdot \operatorname{div}_{\Gamma} \vec{\varphi}_j(y) \right) dx dy \right] = \\ = \frac{i}{kZ_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{\varphi}_j(x) dx, \quad (2) \end{aligned}$$

for each $1 \leq i \leq n$. The set of equations (2) can be recast in matrix form as

$$A\lambda = b, \quad (3)$$

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

$$\begin{aligned} A_{ij} &= \int_{\Gamma} \int_{\Gamma} G(|y-x|) \left(\vec{\varphi}_i(x) \cdot \vec{\varphi}_j(y) - \frac{1}{k^2} \operatorname{div}_{\Gamma} \vec{\varphi}_i(x) \cdot \operatorname{div}_{\Gamma} \vec{\varphi}_j(y) \right) dx dy, \\ b_j &= \frac{i}{kZ_0} \int_{\Gamma} \vec{E}_{inc}(x) \cdot \vec{\varphi}_j(x) dx. \end{aligned}$$

The set of unknowns are associated with the vectorial flux across an edge in the mesh. The coefficient matrix A is dense complex symmetric for EFIE, unsymmetric for CFIE and MFIE. The right-hand side varies with the frequency and the direction of the illuminating wave.

Linear systems issued from boundary element discretizations can be large in applications, although their size n is typically much smaller compared to those arising from FE or FV formulations of the same problem. The number of unknowns grows linearly with the size of the scatterer and quadratically with the frequency of the incoming radiation [9]. A scattering simulation involving targets of size in the order of a few tens of wavelength, illuminated at approximately one GHz of frequency, may lead to meshes with a few million points [117]. Efficient out-of-core dense direct solvers based on variants of Gaussian elimination have been proposed for solving blocks of right-hand sides, see e.g. [1, 37]. However, the memory requirements of direct methods are not affordable for solving realistic applications even on modern parallel computers. Memory concerns are thoroughly addressed by Trefethen et al. [121]:

Evidently, in the course of forty-five years, the dimensions of tractable [dense] matrix problems have increased by a factor of 10^3 . This progress is impressive, but it pales beside the progress achieved by computer hardware in the same period - a speedup by a factor of 10^9 , from flops to gigaflops. In the fact that 10^9 is the cube of 10^3 , we see played out in history the $\mathcal{O}(m^3)$ bottleneck of direct matrix algorithms.

Iterative Krylov methods are based on matrix-vector (M-V) multiplications and can solve the problems of space. A significant amount of work has been devoted to design fast algorithms that can reduce the $\mathcal{O}(n^2)$ computational complexity for the M-V product with dense matrices arising from the discretization of boundary element equations, like the Fast Multipole Method (FMM) [60, 102], the panel clustering method [66], the \mathcal{H} -matrix approach [65], wavelet techniques [3, 15], the adaptive cross approximation method [8], the impedance matrix localization method [21], the multilevel

matrix decomposition algorithm [94] and others. The combination of Krylov solvers and FMM is a popular solution approach. For Helmholtz and Maxwell's equations, FMM algorithms enable to speedup M-V multiplications with boundary element matrices down to $\mathcal{O}(n \log n)$ algorithmic and memory complexity depending on the problem and on the specific implementation, see e.g. [48, 53, 111, 47, 35, 46, 120]. Two-level implementations of FMM can reduce the cost of the matrix-vector product operation from $\mathcal{O}(n^2)$ to $\mathcal{O}(n^{3/2})$, a three level algorithm down to $\mathcal{O}(n^{4/3})$ and the Multilevel Fast Multipole Algorithm (MLFMA) to $\mathcal{O}(n \log n)$.

Briefly, multipole techniques exploit the rapid decay of the Green's function and compute interactions amongst degrees of freedom in the mesh at different levels of accuracy depending on their physical distance. The 3D mesh of the object is partitioned recursively into boxes of roughly equal size until the size becomes small compared with the wavelength. The hierarchical partitioning of the object is typically represented using a tree-structured data called *oct-tree* (see Figure 2). Multipole coefficients are computed for all boxes starting from the smallest ones, that are the leaves, and recursively for each parent cube in the tree by summing together multipole coefficients of its children. Interactions of degrees of freedom within one observation box and its close neighboring boxes are computed exactly using MoM; depending on the frequency, they generate between 1% and 2% of the entries of A . Interactions with boxes that are not neighbors of the observation box but whose parent in the oct-tree is a neighbor of the box parent are computed using FMM (see Figure 3). All other interactions are computed hierarchically on a coarser level by traversing the oct-tree.

In the last ten years, significant research efforts have been devoted to implement multiple techniques efficiently on distributed memory parallel computers, resulting in competitive application codes provably scalable to several million discretization points, for instance the FISC code developed at University of Illinois [113, 112, 114], the INRIA/EADS integral equation code AS_ELFIP [117, 118], the Bilkent University code [54, 55] and others. Load imbalances and great demands for high bandwidth are challenging penalty factors for implementing hierarchical methods on modern petascale-class computer systems with tens (or hundreds) of thousands of processors. In this paper we address the other critical component of the iterative solution in this context that is the design of effective preconditioners for FMM.

2 Preconditioning the FMM

Krylov subspace methods solve a matrix problem in $\alpha \cdot N_{iter} \cdot \mathcal{O}(\text{M-V})$ flops where we denote by $\mathcal{O}(\text{M-V})$ the complexity of the M-V product operation that is related to the number of multipole levels. The constant α depends on the implementation of the specific iterative method; N_{iter} is the number of iterations necessary to achieve a certain accuracy depending on the integral operator as well as on object shape and material. Problems with cavities or open surfaces may require many more iterations than closed objects of the same physical size. Additionally, nonuniform meshes tend to produce ill-conditioned MoM matrices which make the approximate solution less accurate. Convergence can be accelerated by preconditioning, which transforms the original linear system into an equivalent one with better spectral properties, *i.e.* with most eigenvalues

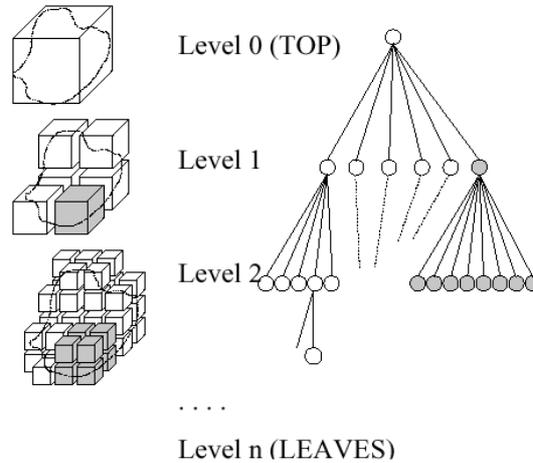


Figure 2: The *oct-tree* data structure representation in the FMM algorithm. Each cube has up to eight children and one parent box except for the largest cube which encloses the whole domain.

clustered close to one. The transformed system assumes the form $M^{-1}Ax = M^{-1}b$, or $AM^{-1}y = b$ with $x = M^{-1}y$, depending if one preconditions from the left or from the right; the matrix M is called *preconditioner*. When the number of unknowns n is related to the wavenumber, the iteration count of Krylov solvers may increase as $\mathcal{O}(n^{0.5})$ on EFIE, see for instance experiments reported in [112]. Thus preconditioning is a critical issue. On CFIE the number of iterations typically increases more slowly, as $\mathcal{O}(n^{0.25})$. Preconditioning is not required for solving smooth and compact Fredholm integral operators of the second kind, while it is critical for non-compact operators associated with singular integral equations [4, 72]. Some optimal analytic preconditioners have been proposed yielding formulations that require less iterations to converge, e.g. [39, 5, 115]. However, they are problem-dependent. In this paper we consider a purely algebraic approach where the preconditioner is computed using information contained in the coefficient matrix of the linear system and can be developed from existent public-domain software implementations (see [106, 33] for a review of methods). Although far from optimal for any specific problem, algebraic methods can be applied to different operators and to changes in the geometry only by tuning a few parameters. We overview techniques that maintain the $\mathcal{O}(n \log n)$ algorithmic and memory complexity of FMM in the construction and in the application phase, and can be combined with little implementation effort with the data structure of multipole codes. We address concerns of parallelization, scalability with respect to problem size, numerical stability; the latter aspect is especially relevant to solving indefinite integral formulations [104]. The substantial improvement on the speed of convergence of iterative methods motivates the research efforts put in the design of robust preconditioners. With the availability of a high quality preconditioner, the choice of the Krylov subspace accelerator is often not so critical, see e.g. [57, 106].

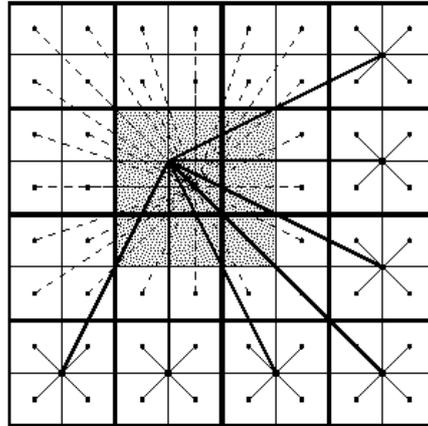


Figure 3: Interactions in the multilevel FMM algorithm. Interactions for the gray boxes are computed directly. We denote by dashed lines cubes that are not neighbors of the cube itself but whose parent is a neighbor of the cube's parent. These interactions are computed using the FMM. All other interactions are computed hierarchically on a coarser level.

For memory concerns, most preconditioning techniques for dense matrices proposed in the literature are computed decomposing the linear system in the form

$$(S + B)x = b \quad (4)$$

where S is a sparse matrix retaining the most relevant contributions to the singular integrals and is easy to invert, while B can be dense. For boundary integral equations of the Fredholm type, 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

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

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. [30] and [33, pp. 182-185].

The simplest approach to compute the local matrix S is to drop the small entries of A below a threshold [75, 123, 42]. However, all the entries of A are often not explicitly available and it becomes necessary to use other information, extracted from either the connectivity graph or the physical mesh of the problem. 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 S can be defined according to the concept of level k neighbours. 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. The large entries in A 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 approximation is likely to be constructed. Three levels can generally provide a good pattern for S [123, 97, 34]. Clearly, we can select the pattern for S also using physical information, that is: for each edge we select all those edges within a sufficiently large sphere that defines our geometric neighbourhood [22, 123]. Comparative experiments amongst different approaches suggest that there is little to choose. All three pattern selection strategies (*i.e.* matrix-, graph- and mesh-based approaches) can provide very good approximations to the dense coefficient matrix for very low sparsity ratio, between 1% and 2% [22]. The mesh-based approach is straightforward to implement in FMM codes that typically partition the object using geometric information. Multipole algorithms yield a matrix decomposition

$$A = A_{diag} + A_{near} + A_{far}, \quad (5)$$

where A_{diag} is the block diagonal part of A associated with interactions of basis functions belonging to the same box, A_{near} is the block near-diagonal part of A associated with interactions within one level of neighboring boxes (they are 8 in 2D and 26 in 3D), and A_{far} is the far-field part of A . Therefore, in a multipole setting a suitable choice for the local matrix may be $S = A_{diag} + A_{near}$.

To show the difficulty of the problem, in Table 1 we report on experiments with the GMRES solver and various algebraic preconditioners applied to a scattering problem from an open cylindrical surface illuminated at 200 MHz of frequency and modeled using EFIE. The mesh is depicted in Figure 4. In

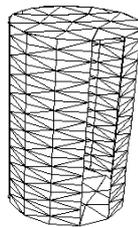


Figure 4: Test problem: an open cylindrical surface.

connection with GMRES, we consider preconditioners of either implicit type that approximately factorize S or of explicit type that approximately invert S , all having roughly the same number of nonzero entries. We adopt the following acronyms:

- *None*, means that no preconditioner is used;
- *Diag*, a simple diagonal scaling, *i.e.* M is the diagonal of S ;

- *SSOR*, the symmetric successive overrelaxation method $M = \frac{(D+\omega E)D^{-1}(D+\omega E^T)}{\omega(2-\omega)}$, where we denote by D the diagonal of S and E is the strict lower triangular part of S ;
- *ILU(0)* [106], the lower/upper incomplete LU factorization $M = \tilde{L}\tilde{U}$, $\tilde{L} \approx L$, $\tilde{U} \approx U$, $S = LU$, where the sparsity pattern of \tilde{L} (resp. \tilde{U}) is equal to that of the lower (resp. upper) triangular part of S ;
- *SPAI* [62], an approximate inverse preconditioner $M \approx S^{-1}$ computed by minimizing $\|I - SM\|_F$. The same pattern of S is imposed to M .
- *AINV* [10], a sparse approximate inverse computed in factorized form by applying an incomplete biconjugation process to S , and dropping small entries below a threshold in the inverse factors.

| Density of $S = 3.18\%$ - Density of $M = 1.99\%$. | | | |
|---|-----------|-----------|-------------------|
| Precond. | GMRES(30) | GMRES(80) | GMRES(∞) |
| <i>None</i> | - | - | 302 |
| <i>Diag</i> | - | - | 272 |
| <i>SSOR</i> | - | 717 | 184 |
| <i>ILU(0)</i> | - | 454 | 135 |
| <i>SPAI</i> | 308 | 70 | 70 |
| <i>AINV</i> | - | - | - |

Table 1: Number of iterations using GMRES and various preconditioners on a test problem, a cylinder with an open surface, discretized with $n = 1299$ edges. The tolerance is set to 10^{-8} . The symbol '-' means that no convergence was achieved after 1000 iterations. The results are for right preconditioning.

We clearly see the failures of many standard methods. Simple preconditioners like the diagonal of A , diagonal blocks, or a band are effective when the coefficient matrix has some degree of diagonal dominance [113]. Block diagonal (Jacobi) preconditioner is generally more robust than its pointwise counterpart and requires insignificant CPU time and storage in the setup phase. Each diagonal block is a LU factorization of the corresponding diagonal block of A_{diag} and is overwritten by the L and the U factors. For ill-conditioned and/or indefinite formulations like EFIE, block Jacobi is not robust and can be only partially improved by re-ordering the matrix prior to the factorization [43, 52], so that more robust methods are beginning to be investigated.

The quest for optimal dense matrix solvers to use in large-scale applications codes and the availability of efficient parallel MLFMA implementations have produced a vivid cross-fertilization of ideas, algorithms, benchmarking activities in the field of preconditioning rather than a specialization in specific technologies. 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 the next sections we identify some suitable class of algebraic methods for solving large dense linear

systems arising from the discretization of EM scattering applications expressed in an integral formulation and we discuss some relevant trends and problems.

3 Incomplete LU (ILU) factorization methods

ILLU-type algorithms compute an approximate triangular decomposition of the coefficient matrix by means of an incomplete Gaussian elimination process. The preconditioner has the form $M = \tilde{L}\tilde{U}$, $\tilde{L} \approx L$, $\tilde{U} \approx U$ where we denote by L , U the lower/upper triangular factors of S . Early experiments with ILU on dense matrices are found in [24,109]. Incomplete factorizations are applied successfully for solving integral equation models arising in electromagnetic scattering [91], bioelectric inverse problems [98], magnetoencephalography [99] as well as other applications. On CFIE, simple pattern selection strategies for the triangular factors are effective. Selecting the nonzero pattern of S for the triangular factors results in robust incomplete factorizations that may deliver rates of convergence similar to a complete *LU* factorization and are clearly more effective than block Jacobi preconditioners [91]. For solving indefinite systems arising from EFIE, it is often necessary to enlarge the pattern of the approximate factors to reduce both number of iterations and solution time. In this case the maximum number of entries per column is usually taken equal to the number of nonzeros of S in the attempt to maintain the algorithmic and memory complexity of MLFMA. This value is easy to determine after the initial setup of the problem. Small entries computed during the factorization are dropped below a threshold, to make storage and application of the preconditioner more economical [84,24,91].

On indefinite systems small pivots often appear during the factorization, producing ill-conditioned factors that may result in unstable triangular solves [24]. The same behavior is observed in the sparse case as well, see e.g. [38]. We illustrate this phenomenon in Table 2 where we show experiments with an *ILLU* preconditioner computed from S using different values of density on a sphere of 1 meter length illuminated at 300 MHz. The problem is modeled using EFIE; the mesh is discretized with 2430 edges (Figure 5). We use the pattern selection strategy based on levels of fill-in described in [106]. The set \mathcal{F} of fill-in entries to be kept for the approximate lower triangular factor \mathcal{L} is defined by

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

where the 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 computed as follows:

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

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

We denote the resulting preconditioner by *ILLU*(ℓ) [106]. Observe that the larger ℓ , the higher the density of the preconditioner. We report on experiments with various values for the density of the matrix S and levels of fill-in. We can see that

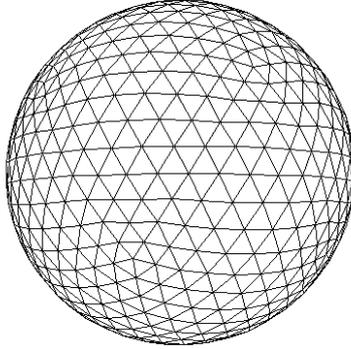


Figure 5: A model problem: a sphere of 1 meter length, illuminated at 300 MHz. The mesh is discretized with 2430 edges.

increasing the fill-in parameter may produce much more robust preconditioners than $ILU(0)$ applied to a denser sparse approximation of the original matrix; $ILU(1)$ may deliver a good rate of convergence provided the coefficient matrix is not too sparse. However, the factorization of a very sparse approximation (up to 2%) of the coefficient matrix can be stable and accelerate significantly the convergence, especially if at least one level of fill-in is retained. For higher values of the density of S , the factors may become progressively ill-conditioned, the triangular solves unstable and consequently the preconditioner is useless. The table also shows that ill-conditioning of the factors is not related to ill-conditioning of A . The problem can be detected before starting the iterations using condition estimators such as $\|(\mathcal{L} \cdot \mathcal{U})^{-1} e\|_{\infty}$ (we denote by e the vector of all ones) which provides an upper bound for $\|(\mathcal{L} \cdot \mathcal{U})^{-1}\|_{\infty}$ and can be applied at the cost of one forward and one backward substitution. If ill-conditioning is revealed, possible remedies may be shifting the diagonal entries, reordering the matrix elements prior to computing the factorization, or applying numerical pivoting during the factorization. A complex diagonal shift can help to compute a more stable preconditioner by shifting along the imaginary axis the eigenvalues close to zero in the spectrum of the coefficient matrix. However, it is not easy to tune *a priori* and its effect on the convergence is difficult to predict [24]. Experiments by Malas et al. show that pivoting can be a robust approach to overcome the problem [91]; in this case, the i th row of the factor is computed as soon as $permtol \times |s_{ij}| > |s_{ii}|$, where $permtol$ is the permutation tolerance and s_{ij} are the entries of S .

For sequential runs, incomplete factorization methods are very competitive. The implementation can be carried out block-wise reflecting the block structure of S . Block row operations may replace standard row operations and diagonal blocks are inverted as pivots instead of the diagonal entries, enabling to use higher level BLAS that exploits data locality in the cache memory. The

| Density of $S = 2\%$ | | | | |
|----------------------|--------------------------|------------------------------|-----------|-----------|
| $IC(\text{level})$ | Density of \mathcal{L} | $\kappa_\infty(\mathcal{L})$ | GMRES(30) | GMRES(50) |
| $IC(0)$ | 2.0% | $2 \cdot 10^3$ | 378 | 245 |
| $IC(1)$ | 5.1% | $1 \cdot 10^3$ | 79 | 68 |
| $IC(2)$ | 9.1% | $9 \cdot 10^2$ | 58 | 48 |
| Density of $S = 4\%$ | | | | |
| $IC(\text{level})$ | Density of \mathcal{L} | $\kappa_\infty(\mathcal{L})$ | GMRES(30) | GMRES(50) |
| $IC(0)$ | 4.0% | $6 \cdot 10^9$ | – | – |
| $IC(1)$ | 11.7% | $2 \cdot 10^5$ | – | – |
| $IC(2)$ | 19.0% | $7 \cdot 10^3$ | 40 | 38 |
| Density of $S = 6\%$ | | | | |
| $IC(\text{level})$ | Density of \mathcal{L} | $\kappa_\infty(\mathcal{L})$ | GMRES(30) | GMRES(50) |
| $IC(0)$ | 6.0% | $8 \cdot 10^{11}$ | – | – |
| $IC(1)$ | 18.8% | $5 \cdot 10^{11}$ | – | – |
| $IC(2)$ | 29.6% | $7 \cdot 10^4$ | – | – |

Table 2: Number of iterations of GMRES varying the sparsity level of S and the level of fill-in of the approximate factor \mathcal{L} on the model problem of Figure 5 ($n = 2430$, $\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty \approx \mathcal{O}(10^2)$). The symbol ‘-’ means that convergence was not obtained after 500 iterations.

block size is equal to the number of nodes per leaf-box. The parallelization of ILU-type algorithms is not straightforward but can be carried out using domain decomposition techniques based on the Schur complement at the cost of moderate computational overhead [89, 95, 106].

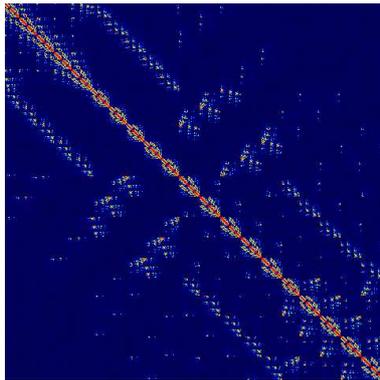
To date, at our best knowledge the largest experiment with ILU-type preconditioners on dense matrices has been reported by Malas et al. at Bilkent University on surface scattering problems on the order of $\mathcal{O}(10^5)$ unknowns [91]. At University of Kentucky, Lee et al. apply ILU for solving hybrid surface-volume integral equations on the order of $\mathcal{O}(10^5)$ unknowns as well [84]. Although inherently sequential, efficient parallel implementations of ILU are today available. The scalability of *ILU* preconditioners for solving very large dense complex problems is an open issue to explore.

4 Approximate inverse

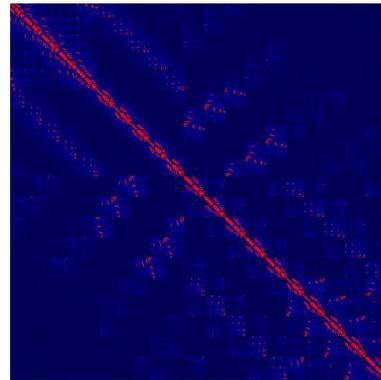
A substantial amount of work has been recently devoted to methods that approximate the inverse of the coefficient matrix explicitly, see for instance [97, 2, 31, 108]. The approximate inverse, denoted by $M \approx S^{-1}$, is applied as preconditioner at each step of an iterative solver by carrying out one or more sparse M-V products depending on the implementation. The approximate inverse may be computed in factorized form as $M = \tilde{G}\tilde{Z}$, where $\tilde{G} \approx U^{-1}$ and $\tilde{Z} \approx U^{-1}$ are approximation of the inverse triangular factors of S . This approach is clearly attractive for parallelism and may be numerically more stable than ILU-type algorithms which require triangular solves [51, 77, 6, 63].

The boundary element method discretizes integral equations on the surface

of the scattering object, 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. Therefore, a very sparse matrix can still retain the most relevant contributions from the singular integrals that give rise to dense matrices. Owing to the decay of the Green's function, the inverse of A may exhibit a very similar structure to A . In Figures 6 it is possible to observe the typical decay of the discrete Green's function of surface integral equations. The model problem is a small sphere, which is representative of the general trend. 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. In the density coloured plot, large to small magnitude entries are depicted in different colours, from red to green, yellow and blue. The discrete Green's function peaks at a point, then it decays rapidly, and far from the diagonal only a small set of entries have large magnitude. Clearly, sparse approximate inverses can capture effectively such rapid decay and may be a suitable approach for this problem class. A good pattern for the approximate inverse is likely to be the nonzero pattern of a sparse approximation to A , constructed by dropping all the entries lower than a prescribed global threshold (see Figure 7).



(a) Pattern of the large entries of A



(b) Pattern of the large entries of A^{-1}

Figure 6: Structure of the large entries of A (on the left) and of A^{-1} (on the right). Large to small entries are depicted in different colors, from red to green, yellow and blue. The test problem is a small sphere.

Several research groups have successfully applied approximate inverse preconditioners in combination with MLFMA, for instance at CERFACS [25] and at Bilkent University [90] for solving large surface scattering problems, and at University of Kentucky [85] for solving hybrid surface-volume integral problems. In these studies the pattern of the preconditioner is computed from the nonzero structure of the near-field multipole matrix S and is applied prior to computing the entries of M , simplifying the setup phase considerably. The actual entries of M may be computed by minimizing the error matrix $\|I - SM\|_F$ for right preconditioning ($\|I - MS\|_F$ resp. left preconditioning). The Frobenius-norm allows to decouple the constrained minimization problem into n independent linear least-squares problems, one for each column (resp. row) of

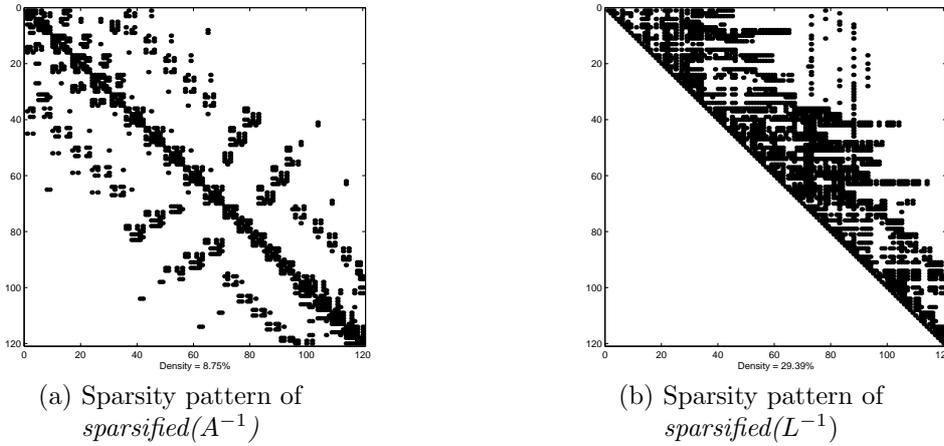


Figure 7: Sparsity patterns of a A^{-1} (on the left) and of L^{-1} (on the right) after dropping all the entries of relative magnitude smaller than 5.0×10^{-2} . The test problem is a small sphere.

M when preconditioning from the right (resp. from the left). The independence of the least-squares problems follows immediately from the identity

$$\|I - SM\|_F^2 = \sum_{j=1}^n \|e_j - Sm_{\bullet j}\|_2^2, \quad (6)$$

where e_j is the j th canonical unit vector and $m_{\bullet j}$ is the column vector representing the j th column of M . In the case of right preconditioning, the analogous relation

$$\|I - MS\|_F^2 = \|I - S^T M^T\|_F^2 = \sum_{j=1}^n \|e_j - S^T m_{j\bullet}\|_2^2 \quad (7)$$

holds, where $m_{j\bullet}$ is the column vector representing the j th row of M . Note that the preconditioner is not guaranteed to be nonsingular and in general it does not preserve any symmetry of A ; some possible remedies are discussed in [62, 24].

If the nonzero pattern of M

$$\mathcal{P} = \{ (i, j) \in [1, n]^2 \text{ s.t. } m_{ij} \neq 0 \}$$

is applied in advance, the structure for the j th column of M is automatically determined, and defined as

$$\mathcal{C}_j = \{ i \in [1, n] \text{ s.t. } (i, j) \in \mathcal{P} \}.$$

The least-squares solution involves only those columns of S indexed by \mathcal{C}_j ; we indicate this subset by $S(:, \mathcal{C}_j)$. Because S is sparse, many rows in $S(:, \mathcal{C}_j)$ are usually null, not affecting the solution of the least-squares problems (6). Thus denoting by \mathcal{R}_j the set of indices corresponding to the nonzero rows in $S(:, \mathcal{C}_j)$,

by $\widehat{S} = S(\mathcal{R}_j, \mathcal{C}_j)$, by $\widehat{m}_j = m_j(\mathcal{C}_j)$, and by $\widehat{e}_j = e_j(\mathcal{C}_j)$, the actual “reduced” least-squares problems to solve are

$$\min \|\widehat{e}_j - \widehat{S}\widehat{m}_j\|_2, \quad j = 1, \dots, n. \quad (8)$$

Usually problems (8) have much smaller size than problems (6) and can be efficiently solved by dense QR factorization.

The preconditioner can be combined with the box-wise data structure of multipole codes illustrated in Figure 2 using a block strategy. All the columns of M that are associated with edges belonging to the same box have an identical nonzero pattern. This means that only one block dense QR factorization per leaf box in the oct-tree needs to be computed. The least-squares can be solved independently on different processors. The use of BLAS-3 operations enables to exploit data locality in the cache memory and may reduce the setup cost of the approximate inverse on the order of $\mathcal{O}(n)$ arithmetic operations [25].

In Table 3 we show experiments extracted from [25] to illustrate the numerical behavior of the Frobenius-norm minimization method combined with parallel MLFMA for solving Eqn (1). We refer to this preconditioner as SPAI (SParse Approximate Inverse). The sparsity pattern is taken equal to that of the near-field matrix $S = A_{diag} + A_{near}$. The test problems, presented in Figure 8, arise from realistic RCS calculations in industry. The results reported in Table 3 refer to EFIE. Large problem size is obtained by illuminating the same obstacle at increasing frequency; the value of the frequency is in the range between 12 and 42 GHz for the Cetaf problem (*i.e.* $24-84\lambda$ of physical size), between 2.3 and 11.4 GHz for the aircraft problem ($14-73\lambda$) and around 10 GHz (approximately 30λ) for the Cobra problem. The experiments are run in single precision complex arithmetic on eight processors of a Compaq Alpha server. The Compaq Alpha server is a cluster of Symmetric Multi-Processors. Each node consists of four DEC Alpha processors (EV 6, 1.3 GFlops peak) that share 512 MB of memory. On that computer, the temporary disk space that can be used by the out-of-core solver is around 189 GB. We report on the number of M-V products and solution time for GMRES; in the table, “d” means day, “h” hour and “m” minute. The number of multipole levels in MLFMA is selected adaptively by the algorithm; more precisely, the object is partitioned until the size of the leaf-boxes in the oct-tree is half of a wavelength [117]. The tolerance for GMRES is fixed equal to 10^{-3} on the normwise backward error $\frac{\|r\|}{\|b\|}$, where r denotes the residual and b the right-hand side of the linear system. This level of accuracy is on the same order of magnitude as the relative error of the M-V product operation in our multipole code, estimated comparing with the level 2 BLAS routine GEMV on a small setup. It can be considered accurate enough for engineering purposes as it enables the correct reconstruction of the RCS of the object.

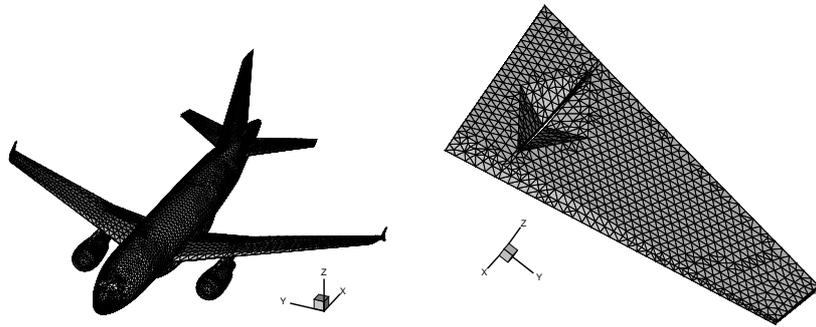
We see that the iterative solver enables to solve fairly large problems, although it does not scale very well with the number of unknowns. For GMRES(∞) the increase in the iteration count is less significant, even though convergence cannot be obtained on the largest systems because we either exceed the memory limits or the time limit allocated to a single run. On the Cetaf geometry, the solution time for the GMRES method increases superlinearly for small and medium problems, but nearly quadratically for large problems. On the largest test case, discretized with one million unknowns, unrestarted

GMRES does not converge after 750 iterations requiring more than nine hours of computation on 32 processors. The Airbus aircraft is very difficult to solve because the mesh has many surface details and the discretization matrices become ill-conditioned. On small and medium problems, the number of GMRES iterations increases with the problem size, and the solution time increases superlinearly. On the largest test case, discretized with one million unknowns, full GMRES exceeds the memory limit on 64 processors. In this case, the use of large restarts (120 in this table) does not enable convergence within 2000 iterations except on a small mesh of size 94704. From a timing point of view, the solution of full GMRES is strongly affected by the orthogonalisation involved in the Arnoldi procedure. On the Cetaf problem discretized with 531900 points, the number of iterations of GMRES(120) is twice as large with respect to full GMRES, but GMRES(120) is about twice as cheap. Provided we get convergence, the use of a large restart often reduces the solution time even though it significantly deteriorates the convergence. We mention that the solution of the problem associated with the aircraft with 213084 degrees of freedom modeled using the CFIE is much simpler; it requires only 129 iterations of unpreconditioned full GMRES and 22 iterations of preconditioned full GMRES. Furthermore, preconditioned full GMRES converges in 24 iterations on the aircraft with more than a million degrees of freedom. A sequential direct solver would require 37 years of computation and 8 Tbytes of storage!

In Table 4 we illustrate the parallel scalability of the approximate inverse. Problems of increasing size are solved on a larger number of processors, keeping the number of unknowns per processor constant. We observe the very good parallel scalability of the construction and of the application of the approximate inverse that makes it a suitable candidate for massively parallel implementation.

The main focus of the paper is to discuss recent developments on optimal matrix solvers for EFIE, which is tough to solve using iterative methods. The advantages of this formulation are numerous; in particular, it does not require any hypothesis on the geometry of the object. We should nevertheless mention that for closed geometries the CFIE can also be used. The linear systems arising from the CFIE formulation are much easier to solve. For instance the solution of the problem associated with the aircraft with 213084 degrees of freedom requires only 129 iterations of unpreconditioned full GMRES, and 22 iterations of preconditioned full GMRES. Furthermore, preconditioned full GMRES converges in 24 iterations on the aircraft with more than a million degrees of freedom. Because the linear systems arising from the CFIE are not challenging from a linear algebra point of view, we do not consider them further in this study.

Approximate inverse techniques are typically more stable than incomplete factorization methods which may suffer from ill-conditioning of the triangular factors especially in the indefinite case as discussed in Section 3. However, a potential source of instability may arise in the presence of clusters of small or isolated eigenvalues in the spectrum of the preconditioned matrix. The Frobenius-norm minimization method succeeds in grouping most of the eigenvalues close to point one of the spectrum but it often leaves a few very small eigenvalues close to the origin. This can be observed in Figure 9, where we depict the eigenvalue distribution of the preconditioned matrix for the cylinder problem of Figure 4 that is fairly representative of the general case. It is known



(a) The aircraft problem. The physical size is $1.8 \text{ m} \times 1.9 \text{ m} \times 0.65 \text{ m}$. (b) The Cetaf problem. The geometry represents a wing with a slit. The physical size is $50 \text{ cm} \times 30 \text{ cm} \times 5 \text{ cm}$.



(c) The Cobra problem, a complex geometry that represents an air intake. Its size is $67.9 \text{ cm} \times 23.3 \text{ cm} \times 11 \text{ cm}$.

Figure 8: Geometries considered for the numerical experiments. Courtesy of EADS-CCR Toulouse.

that the convergence of Krylov methods for solving $Ax = b$ can be described in terms of polynomial approximations. Fast convergence is achieved if a low order polynomial can be found which has value 1 at the origin and approximately 0 on the set of the eigenvalues of A , e.g. [107, 59]. Clearly, the presence of spread or very small eigenvalues makes this difficult to achieve. Deflating algorithms have been used as an attempt of a possible remedy [23]. They apply low-rank matrix updates on top of an existing preconditioner and have the effect of shifting to one a small cluster of small eigenvalues of the preconditioned matrix, resulting in more favorable spectral properties and consequently faster convergence of Krylov methods. Basically, deflation proceeds by projecting the preconditioned system in the coarse (*i.e.* low-dimensional) space associated with approximate eigenvectors corresponding to its smallest eigenvalues and solving it exactly in this coarse space; the information is then used to update the preconditioned residual. In the formulation introduced in [23], the spectrally updated preconditioner assumes the form $\widehat{M} = M + M_{upd}$, where we denote by $M_{upd} = V_\epsilon \widehat{A}^{-1} U_\epsilon^H M$, by V_ϵ (resp. U_ϵ) the set of right (resp. left) eigenvectors

| Cetaf | | | | |
|----------|--------------|-----------|--------------------------------|---------------|
| Size | Density SPAI | Time SPAI | GMRES(∞) | GMRES(120) |
| | | | Iter / Time | Iter / Time |
| 86256 | 0.18 | 4m | 656 / 1h 25m | 1546 / 1h 44m |
| 134775 | 0.11 | 6m | 618 / 1h 45m | 1125 / 1h 55m |
| 264156 | 0.06 | 13m | 710 / 9h | 1373 / 4h 46m |
| 531900 | 0.03 | 20m | 844 / 1d 18m | 1717 / 14h 8m |
| 1056636 | 0.01 | 37m | T.L.E. / >9h ⁽³²⁾ | - / > 1d |
| Aircraft | | | | |
| Size | Density SPAI | Time SPAI | GMRES(∞) | GMRES(120) |
| | | | Iter / Time | Iter / Time |
| 94704 | 0.28 | 11m | 746 / 2h 9m | 1956 / 3h 13m |
| 213084 | 0.13 | 31m | 973 / 7h 19m | - / 7h 56m |
| 591900 | 0.09 | 1h 30m | 1461 / 16h 42m ⁽⁶⁴⁾ | - / 1d 57m |
| 1160124 | 0.02 | 3h 24m | M.L.E. ⁽⁶⁴⁾ / N.A. | - / > 4d |
| Cobra | | | | |
| Size | Density SPAI | Time SPAI | GMRES(∞) | GMRES(120) |
| | | | Iter / Time | Iter / Time |
| 60695 | 0.24 | 2m | 369 / 26m | 516 / 23m |
| 179460 | 0.09 | 7m | 353 / 1h 11m | 406 / 1h 2m |

Table 3: Number of matrix-vector products and elapsed time required to converge on two problems on 8 processors of the Compaq machine, except those marked with ^(k), that were run on k processors. Tolerance for the iterative solution was 10^{-3} . The symbol '-' means that no convergence was achieved in 2000 iterations. Acronyms: N.A. \equiv not available. M.L.E. \equiv memory limits exceeded, T.L.E. \equiv CPU time limits exceeded.

of MA associated with the set of eigenvalues λ_i with $|\lambda_i| \leq \varepsilon$ and $\widehat{A} = U_\varepsilon^H M A V_\varepsilon$ is the projection of MA on the coarse space. Observe that M_{upd} is a low-rank matrix, hence it is computationally cheap to apply. The updated preconditioned matrix $\widehat{M}A$ preserves the eigenvectors of MA and has eigenvalues η_i given by

$$\begin{cases} \eta_i = \lambda_i & \text{if } |\lambda_i| > \varepsilon, \\ \eta_i = 1 + \lambda_i & \text{if } |\lambda_i| \leq \varepsilon. \end{cases}$$

Similar formulations hold for right preconditioning. Weaker formulations, that do not preserve the eigenvectors of MA but require only the computation of V_ε are also possible to derive and are discussed in [23].

In Table 5 we illustrate the effect of applying spectral updates to the Frobenius-norm minimization preconditioner for the Almond problem, a standard electromagnetic test case representing the ogive of a missile. The geometry is depicted in Figure 1. We see a noticeable enhancement of the performance of the iterative solver. By selecting up to ten eigenpairs the number of iterations decreases by nearly a factor of two. The gain in terms of iterations is strongly related to the magnitude of the shifted eigenvalues [23]. A speedup in convergence is obtained when a full cluster of small eigenvalues

| Parallel scalability results | | | | |
|------------------------------|----------|-------------------------|---------------------------------|----------------------------|
| Problem size | Nb procs | Construction time (sec) | Elapsed time precondition (sec) | Elapsed time mat-vec (sec) |
| 112908 | 8 | 513 | 0.39 | 1.77 |
| 221952 | 16 | 497 | 0.43 | 2.15 |
| 342732 | 24 | 523 | 0.47 | 3.10 |
| 451632 | 32 | 509 | 0.48 | 2.80 |
| 900912 | 64 | 514 | 0.60 | 3.80 |

Table 4: Parallel scalability of the implementation on the aircraft problem.

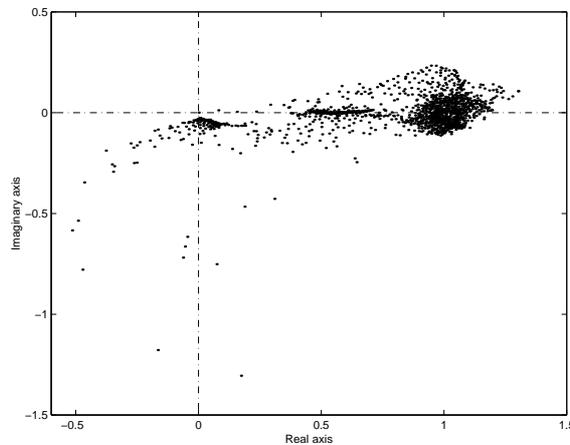


Figure 9: Eigenvalue distribution in the complex plane of the coefficient matrix for a model problem that is representative of the general trend. The problem is the cylinder of Figure 4 modeled using the EFIE formulation.

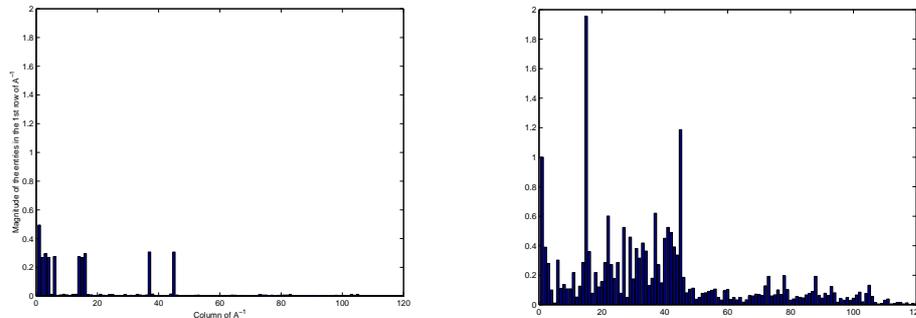
is completely removed from the spectrum. The application of the correction update at each iteration step costs $2nk + k^2$ where k is the size of the coarse space. In the reported experiments, the eigenvectors are computed in forward mode by ARPACK in a preprocessing phase. To give an idea of this cost, the eigencomputation of 30 eigenvectors on the Almond problem takes 1100 matrix-vector products and 1h of time on 32 processors (while constructing the approximate inverse, which is 0.19 dense, takes 6m on 8 processors). This extra-cost is quickly amortized as many right-hand sides have usually to be solved to compute the so called radar cross section where linear systems with the same coefficient matrix and up to hundreds of different right-hand sides are solved, ranging over the complete set of directions between the transmitter and the receiver. Extensive experiments with deflation techniques for solving electromagnetic scattering problems are found in [92, 23].

Finally we mention that Carpentieri et al. report on disappointing experiments with algorithms that compute the approximate inverse methods in factorized form [24]. The reason of failure is that for many integral formulations like EFIE and CFIE, the inverse factors may be totally unstructured as shown

| Almond problem, $n = 104793$ - $frequency = 2.6$ GHz | | | | |
|--|-------------------------------|----------------|----------------|----------------|
| | Dimension of the coarse space | | | |
| | 0 | 10 | 30 | 50 |
| GMRES(50) | 1524 (1^h 17^m) | 883 (45^m) | 368 (20^m) | 284 (15^m) |
| GMRES(∞) | 242 (14^m) | 134 (9^m) | 92 (6^m) | 77 (6^m) |

Table 5: Experiments with sparse approximate inverses and spectral deflation. We show the number of iterations of GMRES and the solution time using an increasing number of deflated eigenvalues. The scattering problem is modeled using EFIE.

in Figure 7(b). In this case, selecting a priori a sparse pattern for the factors can be extremely hard as no real structures are revealed, preventing the use of techniques like FSAI [75, 77, 76, 79, 78] which require to provide such pattern in advance. In Figure 10 we plot the magnitude of the entries in the first column of A^{-1} and L^{-1} with respect to their row index (very similar behavior has been observed for all the other columns). Those plots indicate that dynamic pattern selection strategies, that drop small entries below a user-defined threshold during the computation like in the AINV preconditioner [10, 11], may be very difficult to tune as they can easily discard relevant information and 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.



(a) Histogram of the magnitude of the entries of the first column of A^{-1} (b) Histogram of the magnitude of the entries of the first column of L^{-1}

Figure 10: Histograms of the magnitude of the entries of one column of A^{-1} and its lower triangular factor on a small sphere. A similar behavior has been observed for all the other columns.

5 MLFMA-based preconditioners

Known bottlenecks of parallel solvers are lack of scalability and needs of large memory and interconnection bandwidth. Ideally, one wants to achieve constant solution time when problem size and number of processors increase proportionately. In the experiments reported in Table 3, the solution time for GMRES increases nearly quadratically for large problems. Scalability is difficult to achieve especially at high frequency due to the standard implementation choice of relating the size of the smallest boxes of the oct-tree to the wavelength. The density of the preconditioner decreases for increasing problem size because the pattern of the sparse approximate inverse is computed from the box-wise partitioning of the object. When the near-field matrix S is very sparse and each degree of freedom is coupled only to one level of neighboring cubes, any information associated with the far-field may be totally lost.

Multipole approximations of an integral operator compute the near-field mesh interactions exactly and yield low-rank representations of the far-field interactions that are natural to incorporate in the design of the preconditioner [116]. Carpentieri et al. use multiple expansions of the integral operator not only for the M-V product operation, but also for the preconditioning operation that is performed at each iteration of a Krylov method. As preconditioner, they carry out a few steps of an inner Krylov method. The outer iterative solver must allow to use variable preconditioning [25], for instance FGMRES [105] and GMRES* [49, p. 91] may be used; the latter reduces to GMRESR [50] when the inner solver is GMRES. A sketch of the algorithm is presented in Figure 11. Iterations are stopped after a fixed number of steps. The inner solver is preconditioned to decrease rapidly the residual, typically in a few number of steps, and the matrix-vector products in the inner and outer solvers are carried out using MLFMA with different levels of accuracy. Highly accurate MLFMA is applied in the outer iterations which govern the final accuracy; less accurate MLFMA is applied in the inner iterations that only attempt to give a rough approximation of the solution. We remark that different accuracy for the M-V calculation can be implemented by tuning various parameters in MLFMA such as the size of the smallest box, the number of multipole levels and of integration points [117]. In early experiments by Grama et al. with inner-outer schemes combined with MLFMA, no preconditioner was used in the inner iterations and the complexity was higher than for standard GMRES [58].

We report some results to show the beneficial effect of using a two-level scheme especially on large systems in Tables 6-7. On the same geometries considered in the previous section, we apply FGMRES as outer solver and GMRES for the inner iteration preconditioned by the Frobenius-norm minimization method described in Section 4. We show the number of inner and total outer M-V products and the elapsed time needed to achieve convergence using a tolerance of 10^{-3} on eight processors of the Compaq machine. The comparison with the results reported in Table 3 is fair because GMRES has exactly the same storage requirements as the combination FGMRES/GMRES. For the same restart value, the storage requirement for the FGMRES algorithm is twice as large as for standard GMRES because it requires also to store the preconditioned vectors of the Krylov basis.

The increase in the number of outer iterations is fairly modest except on the

```

Outer solver → FGMRES, GMRES★
Do k = 1, 2, ...
  • Matrix-vector product: FMM with high accuracy
  • Preconditioning: Inner solver (GMRES, TFQMR, ...)
    Do i = 1, 2, ..., MAX_INNER_ITER
      • Matrix-vector product: FMM with low accuracy
      • Preconditioning: SPAI
    End Do
  End Do
End Do

```

Figure 11: Inner-outer solution schemes in the FMM context. Sketch of the algorithm. Iterations are stopped after a fixed number of step (denoted by `MAX_INNER_ITER`).

largest and difficult aircraft test cases. However, note that the scheme enables to solve this challenging industrial problem while classical restarted GMRES does not converge and full GMRES exceeds the memory limits. Similarly, on the Cetaf discretized with one million points the embedded scheme enables convergence in 22 outer iterations whereas GMRES(120) does not converge in 2000 iterations. The savings in time is also noticeable, with a gain ranging from two to four depending on the geometry. On the Cobra test cases, the embedded solver reduces not only the solution time but also the memory used, as FGMRES(15)/GMRES(30) is faster than GMRES(120). The gain is especially visible on the aircraft test example with 213084 unknowns. In that example, the embedded scheme is about 2 hours faster than standard GMRES; up to 973 basis vectors are to orthogonalize while in the inner-outer scheme only a basis up to 60, leading to a significant saving in the orthogonalization procedure that is extremely time consuming. Malas et al. have proposed a modified version of Algorithm 11 that stops the inner iterates when the residual error is below a threshold [90]. Their code is reported to solve a square patch problem discretized using EFIE with approximately 22 million nodes, in 9 outer iterations and less than 7 hours on a moderate number of processors.

| Cetaf | | | | |
|----------|------------|--------|---------------|-------------------------|
| Size | GMRES(120) | | FGMRES(30,60) | |
| | Iter | Time | Iter | Time |
| 86256 | 1546 | 1h 44m | 17+ 960 | 55m |
| 134775 | 1125 | 1h 55m | 15+ 840 | 1h 19m |
| 264156 | 1373 | 4h 46m | 17+ 960 | 2h 22m |
| 531900 | 1717 | 14h 8m | 19+1080 | 6h |
| 1056636 | - | > 1d | 22+1260 | 14h |
| Aircraft | | | | |
| Size | GMRES(120) | | FGMRES(30,60) | |
| | Iter | Time | Iter | Time |
| 94704 | 1956 | 3h 13m | 27+1560 | 2h 14m |
| 213084 | - | 7h 56m | 34+1920 | 5h |
| 591900 | - | 1d 57m | 57+3300 | 1d 9h 45m |
| 1160124 | - | > 4d | 51+2940 | 16h 41m ⁽⁶⁴⁾ |
| Cobra | | | | |
| Size | GMRES(120) | | FGMRES(30,60) | |
| | Iter | Time | Iter | Time |
| 60695 | 708 | 29m | 24+660 | 18m |
| 179460 | 433 | 48m | 20+540 | 42m |

Table 6: Number of matrix-vector products and elapsed time required to converge on eight processors of the Compaq machine. The tests were run on eight processors of the Compaq machine, except those marked with ^(k), which were run on *k* processors. The symbol '-' means that no convergence was achieved in 2000 iterations. The values of *m* and *n* in the symbol FGMRES(*m,n*) refer to the restart of the outer and of the inner solver. For FGMRES we give the number of inner plus total outer iterations needed to achieve convergence. The tolerance for the iterative solution was 10^{-3} .

| Cetaf | | | | |
|----------|------------------------|-------------------------|-----------------------|--------------------------|
| Size | GMRES(∞) | | FGMRES($\infty,60$) | |
| | Iter | Time | Iter | Time |
| 86256 | 656 | 1h 25m | 17+ 960 | 55m |
| 134775 | 618 | 1h 45m | 15+ 840 | 1h 19m |
| 264156 | 710 | 9h | 17+ 960 | 2h 22m |
| 531900 | 844 | 1d 18m | 19+1080 | 6h |
| 1056636 | T.L.E. | $> 9h^{(32)}$ | 22+1260 | 14h |
| Aircraft | | | | |
| Size | GMRES(∞) | | FGMRES($\infty,60$) | |
| | Iter | Time | Iter | Time |
| 94704 | 746 | 2h 9m | 23+1320 | 2h 30m |
| 213084 | 973 | 7h 19m | 30+1740 | 6h 11m |
| 591900 | 1461 | 16h 42m ⁽⁶⁴⁾ | 43+2520 | 12h ⁽³²⁾ |
| 1160124 | M.L.E. ⁽⁶⁴⁾ | $> 1d$ | 43+2520 | 14 h 28m ⁽⁶⁴⁾ |
| Cobra | | | | |
| Size | GMRES(∞) | | FGMRES($\infty,60$) | |
| | Iter | Time | Iter | Time |
| 60695 | 369 | 26m | 21+600 | 17m |
| 179460 | 353 | 1h 11m | 18+510 | 38m |

Table 7: Number of matrix-vector products and elapsed time required to converge on eight processors of the Compaq machine. The tests were run on eight processors of the Compaq machine, except those marked with ^(k), that were run on k processors. The values of m and n in the symbol FGMRES(m,n) refer to the restart of the outer and of the inner solver. For FGMRES we give the number of inner plus total outer iterations needed to achieve convergence. The tolerance for the iterative solution was 10^{-3} . Acronyms: M.L.E. \equiv memory limits exceeded, T.L.E. \equiv CPU time limits exceeded

6 Other preconditioners

Multigrid methods are provably optimal algorithms for solving various classes of partial differential equations. Basically, multigrid-based preconditioners solve a sequence of problems on a hierarchy of grids of different size. Geometric multigrids require a hierarchy of nested meshes to setup the principal components of the algorithm, *i.e.* a coarsening strategy to decrease the number of unknowns, grid transfer operators to move from a grid to another one, coarse grid operators and smoothing procedure, see e.g. [64]. They can show very good scalability properties on some problems class but are difficult to implement on unstructured meshes when a hierarchy of grids is not available and it may be complicate to construct. Attempts to apply these techniques to dense linear systems have obtained mixed success. Early experiments on boundary element equations are reported with geometric versions on model problems, typically the hypersingular and single-layer potential integral operators arising from the Laplace equation [101, 96, 19]. Algebraic multigrid algorithms use only single grid information extracted from either the graph or the entries of the coefficient matrix and are nearly as effective as geometric algorithms in reducing the number of iterations, see e.g [18, 20, 122, 103]. Langer et al. propose to apply an auxiliary sparse matrix reflecting the local topology of the mesh on the fine grid to setup all the components of the multigrid algorithm in a purely algebraic setting [83]. This *gray-box* approach is fairly robust on model problems and maintains the algorithmic and memory complexity of the M-V product operation [82], thus it is well suited to be combined with MLFMA. In [26], an additive two-grid cycle is proposed for multiple right-hand side systems. The algorithm is built on top of a sparse approximate inverse that is used as smoother, while the grid transfer operators are defined using spectral information of the preconditioned matrix. More precisely, the coarse space is defined using a standard Galerkin formula $A_c = RAP$ where the prolongation operator P is selected to be the set of right eigenvectors associated with the set of eigenvalues λ_i of MA with $|\lambda_i| \leq \varepsilon$ and the restriction operator R is selected orthogonal to P . A sketch of the algorithm is presented in Figure 12. After μ smoothing steps, the residual is projected into the coarse subspace by means of the operator R and the coarse space error equation involving A_c is solved exactly. Finally, the error is prolonged back in the original space using the operator P and the new approximation is smoothed again. These two contributions are summed together for the solution update. The use of a damping parameter ω can ensure better clustering of the right part of the spectrum in this case. Selecting $\omega = \frac{\lambda_{max}}{\alpha}$, the spectrum of MA is contracted so that the largest eigenvalue has modulus α . Although in the contraction some small eigenvalues will approach the origin, with an appropriate choice of α the right part of the spectrum is likely to cluster more effectively around one. The two-grid cycle induces a global deflation of eigenvalues in the spectrum of MA . The preconditioned matrix has eigenvalues η_i defined as

$$\begin{cases} \eta_i = 1 & \text{if } |\lambda_i| \leq \varepsilon, \\ \eta_i = 1 - (1 - \omega\lambda_i)^\mu & \text{if } |\lambda_i| > \varepsilon. \end{cases}$$

In Table 8 we show results of one experiment on the Airbus aircraft problem discretized with 94704 nodes. Although small, this problem is difficult enough to solve by iterative methods. Convergence of GMRES preconditioned by the

Frobenius-norm minimization approximate inverse is achieved only with large values of restart. Using only ten eigenvectors, the two-grid cycle enables to obtain convergence for very small restart, therefore with very low memory. The cost of applying the preconditioner is $\mathcal{O}(n)$.

```

1. High frequency correction:
    $e_1^0 = 0$ 
   for  $j=1, \mu$  do
      $e_1^j = e_1^{j-1} + \omega M(r - A_{FMM} e_1^{j-1})$ 
   end for
    $c_1 = (I - PR)e_1^\mu$ 

2. Low frequency correction:
    $c_2 = PA_c^{-1}Rr, A_c = RAP$ 

3. Solution update:
    $z = c_1 + c_2$ 

```

Figure 12: Additive two-grid spectral preconditioning. The algorithm.

| Airbus aircraft problem (size 94704) | | |
|--------------------------------------|----------------------------|------------|
| With M_{SPAI} | GMRES(10) +4000 iterations | |
| μ | GMRES(10) | CPU-time |
| 1 | 1835 | $1^h 07^m$ |
| 2 | 807 | 36^m |
| 3 | 368 | 22^m |

Table 8: Experiments with additive two-grid spectral preconditioning on the Airbus aircraft problem. For these experiments we use $\omega = \frac{2}{3} \lambda_{max}(MA)$ in Algorithm 12

Preconditioners based on wavelet techniques are also receiving considerable interest. The wavelet compression of integral operators with smooth kernels yields nearly sparse matrices with at most $\mathcal{O}(n \log^a n)$ nonzero entries, where a is a small constant that depends on the operator and the wavelet used [45, 13, 68, 81, 71]. The compressed matrix is spectrally equivalent to the original matrix and preconditioning is often needed, see e.g. [29, 70, 27, 28, 32, 56, 69]. Very efficient algorithms have been proposed by Chen et al. at University of Liverpool, based on reordering the wavelet basis so to produce matrices with banded or nearly banded structures that are better suited to design preconditioners with banded [32] or bordered block structure [56, 70]. Other approaches are based on multi-level preconditioners [28] and sparse approximate inverses. The inverse of dense matrices arising from the discretization of boundary integral equations with smooth Green's function can be effectively

compressed in wavelet basis [41, 27, 29]. Denoting by W the level L discrete wavelet transform and by $\widetilde{\cdot}$ the compressed matrices, we may write

$$\widetilde{A}^{-1} = W^T A^{-1} W = (W^T A W)^{-1} = \widetilde{A}^{-1}$$

and the approximate inverse $\widetilde{M} \simeq \widetilde{A}^{-1}$ may be computed by minimizing $\|\widetilde{A}\widetilde{M} - I\|_F$. The inverse of many wavelet matrices have a finger pattern; however, including such pattern is prohibitively expensive [40] so that Chan et al. propose to take a block diagonal pattern [29]. Hawkins et al. propose to compute the approximate inverse directly from A using the relations

$$\|\widetilde{A}\widetilde{M} - I\|_F = \|W^T A M W - I\|_F = \|W W^T A M W - W\|_F = \|A\widehat{M} - W\|_F,$$

where we denote by $\widetilde{M} = W^T M W$ and by $\widehat{M} = M W = W \widetilde{M}$. Since \widetilde{M} has a predictable sparsity pattern, then \widehat{M} has a predictable sparsity pattern that is similar to the pattern of W [69]. Including all scales interactions, the sparse approximate inverse and the discrete wavelet transform have linear complexity in n and can be effectively combined with an operator splitting approach such as (4). Most experiments with wavelet preconditioners are reported for model problems, e.g. Calderon-Zygmund type matrix, single and double layer potentials, the hyper-singular operator. For oscillatory kernels the compressed matrix may be fairly dense and wavelet techniques are less useful. For Helmholtz problems, wavelet Galerkin schemes yield matrices with approximately $\mathcal{O}(kn)$ (k is the wavenumber) which becomes $\mathcal{O}(n^2)$ when the number of unknowns is related to k . If the number of unknowns is independent of k , the wavelet Galerkin scheme may have optimal complexity $\mathcal{O}(n)$ and therefore wavelet techniques are more appealing to use [67, 44, 71].

7 Concluding remarks

Thanks to the use of iterative methods and suitable preconditioners, fast integral solvers involving tens of million unknowns are nowadays feasible and can be integrated in the design processes making boundary element techniques particularly attractive to use in realistic applications. Many undergoing projects around the world are currently targeting petascale-class computer systems with hundreds of thousands of CPU cores. The availability of such large computer resources poses tremendous scientific challenges. It is not sufficient to port existing codes to novel supercomputer architectures but it is necessary to identify optimal high-performance numerical algorithms for better performance. From a linear algebra viewpoint, large-scale simulations are extremely demanding for scalable preconditioners efficiently combined with hierarchical techniques like MLFMA. We have reviewed important advances for the class of incomplete factorizations, sparse approximate inverses and multilevel algorithms. Further investigation is necessary to identify the best class of method for the given problem and the selected computer hardware. The use of more powerful (but also more complex) computing facilities should help in the search for additional speed, but it will also mean that there will be even more factors that need to be considered when attempting to identify the optimal approach in the future.

8 Acknowledgements

The numerical results reported in this paper are extracted from a joint research between CERFACS, INRIA-CERMICS and EADS-CCR Toulouse. The author is grateful to Guillaume Alleon from EADS-CCR Toulouse for providing us with some test examples and to Guillaume Sylvand for providing us with the AS_ELFIP codes. The author is also grateful to Stuart C. Hawkins for insightful discussions on wavelets preconditioners. Thanks are deserved to Gundolf Haase for bringing papers [82, 83] to our attention. Finally, we would like to thank the anonymous referees for their valuable remarks and comments that greatly improved the quality of the paper.

References

- [1] G. Alléon, S. Amram, N. Durante, P. Homsy, D. Pogarielloff, and C. Farhat. Massively parallel processing boosts the solution of industrial electromagnetic problems: High performance out-of-core solution of complex dense systems. In M. Heath, V. Torczon, G. Astfalk, P. E. Bjrstad, A. H. Karp, C. H. Koebel, V. Kumar, R. F. Lucas, L. T. Watson, and Editors D. E. Womble, editors, *Proceedings of the Eighth SIAM Conference on Parallel Computing*. SIAM Book, Philadelphia, 1997. Conference held in Minneapolis, Minnesota, USA.
- [2] 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.
- [3] B.K. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Scientific and Statistical Computing*, 14:159–184, 1993.
- [4] S. Amini and Ke Chen. Conjugate gradient method for second kind integral equations - applications to the exterior acoustic problem. *Engineering Analysis with Boundary Elements*, 6(2):72–77, 1993.
- [5] X. Antoine, A. Bendali, and M. Darbas. Analytic preconditioners for the electric field integral equation. *International Journal for Numerical Methods in Engineering*, 61(8):1310–1331, 2004.
- [6] S. T. Barnard, L. M. Bernardo, and H. D. Simon. An MPI implementation of the SPAI preconditioner on the T3E. *International Journal of High Performance Computing Application*, 13(2):107–123, 1999.
- [7] A. Bayliss, C. I. Goldstein, and E. Turkel. On accuracy conditions for the numerical computation of waves. *J. Comp. Phys.*, 59:396–404, 1985.
- [8] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.
- [9] A. Bendali. *Approximation par elements finis de surface de problemes de diffraction des ondes electro-magnetiques*. PhD thesis, Université Paris VI, 1984.

- [10] M. Benzi, C.D. Meyer, and M. Tũma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Scientific Computing*, 17:1135–1149, 1996.
- [11] M. Benzi and M. Tũma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Scientific Computing*, 19:968–994, 1998.
- [12] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comp. Phys.*, 114:185–200, 1994.
- [13] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Comm. Pure Appl. Math.*, 44:141–183, 1991.
- [14] F. Bilotti and C. Vegni. MoM entire domain basis functions for convex polygonal patches. *Journal of Electromagnetic Waves and Applications*, 17(11):1519–1538, 2003.
- [15] D.M. Bond and S.A. Vavasis. Fast wavelet transforms for matrices arising from boundary element methods. Technical Report 94-174, Cornell University Ithaca, NY, USA, 1994.
- [16] P. Bonnet, X. Ferrieres, J. Grando, J.C. Alliot, and J. Fontaine. Frequency-domain finite volume method for electromagnetic scattering. *Antennas and Propagation Society International Symposium, 1998. IEEE*, 1(21-26):252–255, 1998.
- [17] M.M. Botha. Solving the volume integral equations of electromagnetic scattering. *J. Comput. Phys.*, 218(1):141–158, 2006.
- [18] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55(4):379–393, 1995.
- [19] J.H. Bramble, Z. Leyk, and J.E. Pasciak. The analysis of multigrid algorithms for pseudodifferential operators of order minus one. *Math. Comput.*, 63(208):461–478, 1994.
- [20] A. Brandt. General highly accurate algebraic coarsening. *Electronic Trans. Num. Anal.*, 10:1–20, 1999.
- [21] F.X. Canning. The impedance matrix localization (IML) method for moment-method calculations. *IEEE Antennas and Propagation Magazine*, 1990.
- [22] 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.
- [23] B. Carpentieri, I.S. Duff, and L. Giraud. A class of spectral two-level preconditioners. *SIAM J. Scientific Computing*, 25(2):749–765, 2003.
- [24] B. Carpentieri, I.S. Duff, L. Giraud, and M. Magolu monga Made. Sparse symmetric preconditioners for dense linear systems in electromagnetism. *Numerical Linear Algebra with Applications*, 11(8-9):753–771, 2004.

- [25] B. Carpentieri, 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*, 27(3):774–792, 2005.
- [26] B. Carpentieri, L. Giraud, and S. Gratton. Additive and multiplicative two-level spectral preconditioning for general linear systems. *SIAM J. Scientific Computing*, 29(4):1593–1612, 2007. Accepted for publication.
- [27] T.F. Chan and Ke Chen. Two-stage preconditioners using wavelet band splitting and sparse approximation. UCLA CAM report CAM00-26, Dept of Mathematics, UCLA, USA, 2000.
- [28] T.F. Chan and Ke Chen. On two variants of an algebraic wavelet preconditioner. *SIAM Journal on Scientific Computing*, 24(1):260–283, 2002.
- [29] T.F. Chan, W.-P. Tang, and W.L. Wan. Fast wavelet based sparse approximate inverse preconditioner. *BIT*, 37(3):644–660, 1997.
- [30] Ke Chen. Efficient iterative solution of linear systems from discretizing singular integral equations. *Elec. Trans. Numer. Anal.*, 2:7691, 1994.
- [31] Ke Chen. On a class of preconditioning methods for dense linear systems from boundary elements. *SIAM J. Scientific Computing*, 20(2):684–698, 1998.
- [32] Ke Chen. Discrete wavelet transforms accelerated sparse preconditioners for dense boundary element systems. *Electronic Transactions on Numerical Analysis*, 8:138–153, 1999.
- [33] Ke Chen. *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, 2005.
- [34] Ke Chen and P.J. Harris. Efficient preconditioners for iterative solution of the boundary element equations for the three-dimensional Helmholtz equation. *Applied Numerical Mathematics*, 36(4):475–489, 2001.
- [35] H. Cheng, W.Y. Crutchfield, Z. Gimbutas, L.F. Greengard, J.F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, and J. Zhao. A wideband fast multipole method for the Helmholtz equation in three dimensions. *J. Comput. Phys.*, 216(1):300–325, 2006.
- [36] W. C. Chew, J. M. Jin, C. C. Lu, E. Michielssen, and J. M. Song. Fast solution methods in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 45(3):533–543, 1997.
- [37] W.C. Chew and Y.M. Wang. A recursive T-matrix approach for the solution of electromagnetic scattering by many spheres. *IEEE Transactions on Antennas and Propagation*, 41(12):1633–1639, 1993.
- [38] E. Chow and Y. Saad. Experimental study of ILU preconditioners for indefinite matrices. *Journal of Computational and Applied Mathematics*, 86:387–414, 1997.

- [39] S.H. Christiansen and J.-C. Nédélec. A preconditioner for the electric field integral equation based on Calderon formulas. *SIAM J. Numer. Anal.*, 40(3):1100–1135, 2002.
- [40] A. Cohen and R. Masson. Wavelet methods for second order elliptic problems, preconditioning and adaptivity. *SIAM J. Scientific Computing*, 21:1006–1026, 1999.
- [41] A. Cohen and R. Masson. Wavelet adaptive method for second order elliptic problems: boundary conditions and domain decomposition. *Numerische Mathematik*, 86(2):193–238, August 2000.
- [42] A. Cosnau. Etude d’un préconditionneur pour les matrices complexes dense symmetric issues des équations de Maxwell en formulation intégrale. Note technique ONERA, ONERA, 1996. 142328.96/DI/MT.
- [43] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proc. 24th National Conference of the Association for Computing Machinery, Brandon Press, New Jersey*, pages 157–172. Brandon Press, New Jersey, 1969.
- [44] W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for boundary integral equations—asymptotically optimal complexity estimates. *SIAM J. Numer. Anal.*, 43(6):2251–2271, 2006.
- [45] W. Dahmen, S. Prössdorf, and R. Schneider. Wavelet approximation methods for pseudodifferential equations ii: Matrix compression and fast solution. *Advances in Computational Mathematics*, 1(3):259–335, 1993.
- [46] E. Darrigrand. Coupling of fast multipole method and microlocal discretization for the 3D Helmholtz equation. *J. Comput. Phys.*, 181(1):126–154, 2002.
- [47] E. Darve and P. Havé. Efficient fast multipole method for low-frequency scattering. *J. Comput. Phys.*, 197(1):341–363, 2004.
- [48] B. Dembart and M.A. Epton. A 3D fast multipole method for electromagnetics with multiple levels. Tech. Rep. ISSTECH-97-004, The Boeing Company, Seattle, WA, 1994.
- [49] H.A. Van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge University Press, Cambridge, UK, 2003.
- [50] H.A. Van der Vorst and C. Vuik. GMRESR: A family of nested GMRES methods. *Numerical Linear Algebra with Applications*, 1:369–386, 1994.
- [51] J.C. Dfaz and T. Mansfield. Connection machine implementation of an approximate inverse preconditioning with minimum residual iteration. In *Proc. Symp. Applied Computing*, pages 223–228, 1990.
- [52] I.S. Duff and J. Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J. Matrix Anal. Appl.*, 22(4):973–996, 2000.

- [53] N. Engheta, W.D. Murphy, V. Rokhlin, and M.S. Vassiliou. The fast multipole method (FMM) for electromagnetic scattering problems. *IEEE Transactions on Antennas and Propagation*, 40(6):634–641, 1992.
- [54] Ö. Ergül and L. Gürel. Fast and accurate solutions of extremely large integral-equation problems discretized with tens of millions of unknowns. *Electron. Lett.*, 43(9):499–500, 2007.
- [55] Ö. Ergül and L. Gürel. Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems. *IEEE Transactions on Antennas and Propagation*, 56(8):2335–2345, 2008.
- [56] J. Ford and Ke Chen. Wavelet-based preconditioners for dense matrices with non-smooth local features. *BIT*, 41(2):282–307, 2001.
- [57] G.H. Golub and H.A. van der Vorst. Closer to the solution: iterative linear solvers. Technical report, In I.S. Duff and G.A. Watson editors, 1997. The State of the Art in Numerical Analysis.
- [58] A. Grama, V. Kumar, and A. Sameh. Parallel hierarchical solvers and preconditioners for boundary element methods. *SIAM J. Scientific Computing*, 20(1):337–358, 1999.
- [59] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Frontiers in Applied Mathematics, No. 17. SIAM, Philadelphia, 1997.
- [60] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.
- [61] M.J. Grote. Nonreflecting boundary conditions for electromagnetic scattering. *Int. J. Numer. Model.*, 13:397–416, 2000.
- [62] M.J. Grote and T. Huckle. Parallel preconditionings with sparse approximate inverses. *SIAM J. Scientific Computing*, 18:838–853, 1997.
- [63] M.J. Grote and H. D. Simon. Parallel preconditioning and approximate inverses on the Connection machine. In L.R. Petzold, R.F. Sincovec, D.E. Keyes, M.R. Leuze, and D.A. Reed, editors, *Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing*, pages 519–553, Norfolk, VI, 1993. SIAM Press.
- [64] W. Hackbusch. *Multigrid Methods and Applications*. Springer-Verlag, Berlin, 1985.
- [65] W. Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [66] W. Hackbusch and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.
- [67] H. Harbrecht. *Wavelet Galerkin schemes for the boundary element method in three dimensions*. PhD thesis, Technische Universität Chemnitz, 2001.

- [68] H. Harbrecht and R. Schneider. Wavelet based fast solution of boundary integral equations. In N.M. Chuong et al., editor, *Abstract and Applied Analysis*, pages 139–162. World Scientific Publishing Company, 2004. Proceedings of the International Conference in Hanoi, 2002.
- [69] S.C. Hawkins and Ke Chen. An implicit wavelet sparse approximate inverse preconditioner. *SIAM J. Scientific Computing*, 27(2):667–686, 2005.
- [70] S.C. Hawkins, Ke Chen, and P.J. Harris. An operator splitting preconditioner for matrices arising from a wavelet boundary element method for the Helmholtz equation. *International Journal of Wavelets, Multiresolution and Information Processing*, 3 (4):601–620, 2005.
- [71] S.C. Hawkins, Ke Chen, and P.J. Harris. On the influence of the wavenumber on compression in a wavelet boundary element method for the Helmholtz equation. *International Journal of Numerical Analysis and Modeling*, 4(1):48–63, 2007.
- [72] P.M. Hemker and H.Schippers. Multigrid methods for the solution of Fredholm integral equations of the second kind. *Mathematics of Computation*, 36:215–232, 1981.
- [73] J.M. Jin and V.V. Liepa. A note on hybrid finite element method for solving scattering problems. *IEEE Trans. Ant. Prop.*, 36(10):1486–1489, 1988.
- [74] W.R. Scott Jr. Errors due to spatial discretization and numerical precision in the finite-element method. *IEEE Trans. Ant. Prop.*, 42(11):1565–1569, 1994.
- [75] 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*.
- [76] L. Yu Kolotilina and A. Yu Yeremin. Factorized sparse approximate inverse preconditionings. I: Theory. *SIAM J. Matrix Analysis and Applications*, 14:45–58, 1993.
- [77] 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.
- [78] 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.
- [79] 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.

- [80] K.S. Kunz and R.J. Luebbers. *The Finite Difference Time Domain Method for Electromagnetics*. CRC Press, Boca Raton, 1993.
- [81] C. Lage and C. Schwab. Wavelet Galerkin algorithms for boundary integral equations. *SIAM J. Scientific Computing*, 20(6):2195–2222, 1999.
- [82] U. Langer and D. Pusch. Data-sparse algebraic multigrid methods for large scale boundary element equations. *Applied Numerical Mathematics*, 54(3-4):406–424, 2005.
- [83] U. Langer, D. Pusch, and S. Reitzinger. Efficient preconditioners for boundary element matrices based on grey-box algebraic multigrid methods. *International Journal for Numerical Methods in Engineering*, 58(13):1937–1953, 2003.
- [84] J. Lee, C.-C. Lu, and J. Zhang. Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems. *J. Comp. Phys.*, 185:158–175, 2003.
- [85] J. Lee, C.-C. Lu, and J. Zhang. Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 52(9):2277–2287, 2004.
- [86] R. Lee and A.C. Cangellaris. A study of discretization error in the finite element approximation of wave solution. *IEEE Trans. Ant. Prop.*, 40(5):542–549, 1992.
- [87] S.W. Lee, H. Ling, and R.C. Chou. Ray tube integration in shooting and bouncing ray method. *Micro. Opt. Tech. Lett.*, 1:285–289, 1988.
- [88] J.Y. Li, L.W. Li, and Y.B. Gan. Method of Moments analysis of waveguide slot antennas using the EFIE. *Journal of Electromagnetic Waves and Applications*, 19(13):1729–1748, 2005.
- [89] Z. Li, Y. Saad, and M. Sosonkina. pARMS: a parallel version of the algebraic recursive multilevel solver. Technical Report umsi-2001-100, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2001.
- [90] T. Malas, Ö. Ergül, and L. Gürel. Sequential and parallel preconditioners for large-scale integral-equation problems. In *Computational Electromagnetics Workshop, August 30-31, 2007, Izmir, Turkey*, pages 35–43, 2007.
- [91] T. Malas and L. Gürel. Incomplete LU preconditioning with multilevel fast multipole algorithm for electromagnetic scattering. *SIAM J. Scientific Computing*, 29(4):1476–1494, 2007.
- [92] E. Martin. *Spectral two-level preconditioners for sequences of linear systems*. PhD thesis, INPT, 2005.
- [93] B. McDonald and A. Wexler. Finite element solution of unbounded field problem. *IEEE Trans. Microwave Theory Tech.*, 20:841–847, 1972.

- [94] E. Michielssen and A. Boag. A multilevel matrix decomposition algorithm for analyzing scattering from large structures. *IEEE Transactions on Antennas and Propagation*, 44(8):1086–1093, 1996.
- [95] M. Magolu munga Made and H.A. van der Vorst. ParIC: A family of parallel incomplete Cholesky preconditioners. In M. Bubak, H. Afsarmanesh, R. Williams, and Eds. B. Hertzberger, editors, *High Performance Computing and Networking. Proceedings of the HPCN Europe 2000 Conference, Amsterdam. Lecture Notes in Computer Science, 1823*, pages 89–98. Springer-Verlag, Berlin, 2000.
- [96] T. von Petersdorff and E.P. Stephan. Multigrid solvers and preconditioners for first kind integral equations. *Numer. Meth. for PDE*, 8:443–450, 1992.
- [97] 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.
- [98] J. Rahola and S. Tissari. Iterative solution of dense linear systems arising from boundary element formulations of the biomagnetic inverse problem. Technical Report TR/PA/98/40, CERFACS, Toulouse, France, 1998.
- [99] J. Rahola and S. Tissari. Iterative solution of dense linear systems arising from the electrostatic integral equation in MEG. *Physics in Medicine and Biology*, 47(6):961–975, 2002.
- [100] S.M. Rao, D.R. Wilton, and A.W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. Antennas Propagat.*, AP-30:409–418, 1982.
- [101] S. Rjasanow. Zweigittermethode für eine modellaufgabe bei bemdiskretisierung. *Wiss. Z. Tech. Univ. Karl-Marx-Stadt*, 29(2):230–235, 1987.
- [102] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comp. Phys.*, 86(2):414–439, 1990.
- [103] J.W. Ruge and K. Stüben. *Multigrid Methods, Frontiers in Applied Mathematics 3*, chapter Algebraic multigrid (AMG), pages 73–130. S.F. McCormick ed., SIAM, Philadelphia, PA, 1987.
- [104] Y. Saad. Preconditioning techniques for nonsymmetric and indefinite linear systems. *Journal of Computational and Applied Mathematics*, 24:89–105, 1988.
- [105] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Scientific and Statistical Computing*, 14:461–469, 1993.
- [106] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, New York, 1996.
- [107] 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.

- [108] A.R. Samant, E. Michielssen, and P. Saylor. Approximate inverse based preconditioners for 2D dense matrix problems. Technical Report CCEM-11-96, University of Illinois, 1996.
- [109] K. Sertel and J.L. Volakis. Incomplete LU preconditioner for FMM implementation. *Micro. Opt. Tech. Lett.*, 26(7):265–267, 2000.
- [110] P.P. Silvester and R.L. Ferrari. *Finite Elements for Electrical Engineers*. Cambridge University Press, Cambridge, 1990.
- [111] J.M. Song and W.C. Chew. Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering. *Mico. Opt. Tech. Lett.*, 10(1), 1995.
- [112] J.M. Song and W.C. Chew. The Fast Illinois Solver Code: Requirements and scaling properties. *IEEE Computational Science & Engineering*, 5(3):19–23, 1998.
- [113] J.M. Song, C.-C. Lu, and W.C. Chew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects. *IEEE Transactions on Antennas and Propagation*, 45(10):1488–1493, 1997.
- [114] J.M. Song, C.C. Lu, W.C. Chew, and S.W. Lee. Fast illinois solver code (FISC). *IEEE Antennas and Propagation Magazine*, 40(3):27–34, 1998.
- [115] O. Steinbach and W.L. Wendland. The construction of some efficient preconditioners in the boundary element method. *Adv. Comput. Math.*, 9(1-2):191–216, 1998.
- [116] X. Sun and N.P. Pitsianis. A matrix version of the fast multipole method. *SIAM Rev.*, 43(2):289–300, 2001.
- [117] G. Sylvand. *La Méthode Multipôle Rapide en Electromagnétisme : Performances, Parallélisation, Applications*. PhD thesis, Ecole Nationale des Ponts et Chaussées, 2002.
- [118] G. Sylvand. Complex industrial computations in electromagnetism using the fast multipole method. In P. Joly P. Neittaanmaki G.C. Cohen, E. Heikkola, editor, *Mathematical and Numerical Aspects of Wave Propagation*, pages 657–662. Springer, 2003. Proceedings of Waves 2003.
- [119] A. Taflove. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, Boston, 1995.
- [120] J. Tausch. The variable order fast multipole method for boundary integral equations of the second kind. *Computing*, 72(3-4):267–291, 2004.
- [121] L.N. Trefethen and David Bau , III. *Numerical Linear Algebra*. SIAM Book, Philadelphia, 1997.
- [122] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.

- [123] S.A. Vavasis. Preconditioning for boundary integral equations. *SIAM J. Matrix Analysis and Applications*, 13:905–925, 1992.
- [124] J.L. Volakis, A. Chatterjee, and L.C. Kempel. *Finite element methods for electromagnetics*. IEEE Press, Piscataway, NJ, 1998.