INSTITUTE for MATHEMATICS

(Graz University of Technology)

&

INSTITUTE for MATHEMATICS
and
SCIENTIFIC COMPUTING

(University of Graz)

B. Carpentieri

A matrix-free two-grid preconditioner for solving boundary integral equations in electromagnetism

Report No. 4/2005 November, 2005

Institute for Mathematics D,
Graz University of Technology
Steyrergasse 30
A-8010 Graz, Austria

Institute for Mathematics and
Scientific Computing,
University of Graz
Heinrichstrasse 36
A-8010 Graz, Austria

A matrix-free two-grid preconditioner for solving boundary integral equations in electromagnetism

B. Carpentieri[*]

Abstract

In this paper we describe a matrix-free iterative algorithm based on the GMRES method for solving electromagnetic scattering problems expressed in an integral formulation. Integral methods are an interesting alternative to differential equation solvers for this problem class since they do not require absorbing boundary conditions and they mesh only the surface of the radiating object giving rise to dense and smaller linear systems of equations. However, in realistic applications the discretized systems can be very large and for some integral formulations, like the popular Electric Field Integral Equation, they become ill-conditioned when the frequency increases. This means that iterative Krylov solvers have to be combined with fast methods for the matrix-vector products and robust preconditioning to be affordable in terms of CPU time. In this work we describe a matrix-free two-grid preconditioner for the GMRES solver combined with the Fast Multipole Method. The preconditioner is an algebraic two-grid cycle 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. Experiments on a set of linear systems arising from real radar cross section calculation in industry illustrate the potential of the proposed approach for solving large-scale problems in electromagnetism.

Keywords : Iterative methods, Frobenius-norm minimization method, spectral preconditioner, additive two-grid cycles, electromagnetic scattering applications.

AMS subject classification: 65F10, 65F50, 65N38, 65R20, 78A45, 78A50, 78-08

1 Introduction.

In recent years, considerable research efforts have been devoted to the efficient numerical solution of the Maxwell's equations on high-performance computers. This analysis is of relevant interest for the simulation of many industrial processes like the design of microcircuits, medical equipment and absorbing materials, the study of the electromagnetic compatibility of modern electronic devices, the radar cross section calculation of electrically large objects like aircrafts and many others. The physical issue consists in detecting the diffraction pattern of the outgoing radiation that is scattered back when an object is illuminated by an ingoing electromagnetic wave. Two main approaches are generally used for the numerical solution. One approach, based on differential equation methods (finite element, finite difference and finite volume methods), discretizes a finite volume surrounding the scatterer and gives rise to sparse and very large linear systems of equations. In recent years, alternative strategies based on integral equation methods have received a growing interest. Using the equivalence principle, the Maxwell's equations can be recast into a set of integral equations which relate the electromagnetic fields outside the object to the electric and magnetic currents that are induced on the surface of the object. Integral methods are particularly attractive because

[*]Karl-Franzens University, Institute for Mathematics and Scientific Computing
Heinrichstraße 36, A-8010 Graz, Austria.
Email : bruno.carpentieri@uni-graz.at

the boundary conditions are automatically satisfied and for homogeneous or layered homogeneous dielectric objects they require to mesh only the surface of the radiating object and the discontinuous interfaces between two different materials. They solve for the induced surface currents whereas differential equation methods solve for the electromagnetic scattered fields. The collocation or Galerkin discretization of integral equations give rise to dense matrices that have smaller size compared to those arising from finite element or finite difference discretizations. However, the resulting systems can be very large since their dimension n is of the order of $\lambda/10$, where λ denotes the wavelength. When the frequency increases, the memory cost of direct methods cannot be afforded even on high-performance computers and iterative solvers are mandatory to use. In this paper, we consider the design of robust parallel preconditioners for electromagnetic scattering problems expressed in the popular Electric Field Integral Equation formulation. We focus on this formulation because it gives rise to ill-conditioned systems especially for high frequencies and is routinely used in industry as it is applicable to general objects, including bodies with open surfaces, cavities and disconnected parts. After a brief description of the physical problem in Section 2, in Section 3 we introduce a matrix-free iterative algorithm combined with the FMM. We assess the performance of our method in Section 4 by reporting on experiments on a set of real-life industrial problems arising from radar cross section calculations. Finally, in Section 5 we draw some conclusions arising from the work.

2 The electromagnetic scattering problem

We consider the case of a dielectric body Ω_i with frontier Γ, and we denote by Ω_e the complementary of Ω_i. We indicate by ϵ_e (resp. ϵ_i) the electric permittivity in Ω_e (resp. Ω_i) and by μ_e (resp. μ_i) the magnetic permeability, and we denote ν the outgoing normal on Γ. The entire domain Ω is illuminated by an incident plane wave $(\vec{E}_{inc}, \vec{H}_{inc})$ of angular frequency $\omega = ck = 2\pi c/\lambda$, where the constant c is the speed of light, k is the wave number and $\lambda = c/f$ is the wave length (f is the frequency). The Maxwell's equations for the electromagnetic fields in Ω can be written

$$\left\{ \begin{array}{l} \vec{rot}\vec{E}_e - i\omega\mu_e\vec{H}_e = 0, \\ \vec{rot}\vec{H}_e + i\omega\epsilon_e\vec{E}_e = 0, \\ \lim\limits_{r\to+\infty} r \left| \sqrt{\epsilon_e}\vec{E}_e - \sqrt{\mu_e}\vec{H}_e \wedge \dfrac{\vec{r}}{r} \right| = 0 \end{array} \right. \qquad in\ \Omega_e \tag{1}$$

and

$$\left\{ \begin{array}{l} \vec{rot}\vec{E}_i - i\omega\mu_i\vec{H}_i = 0, \\ \vec{rot}\vec{H}_i + i\omega\epsilon_i\vec{E}_i = 0 \end{array} \right. \qquad in\ \Omega_i. \tag{2}$$

An integral formulation of this problem can be derived by considering the electric and magnetic surface currents $\vec{j}$ and $\vec{m}$ on Γ, that are defined as the tangential parts of the total fields $\vec{H}$ and $\vec{E}$

$$\left\{ \begin{array}{l} \vec{j} = \vec{\nu} \wedge \vec{H}_{tot}, \\ \vec{m} = \vec{\nu} \wedge \vec{E}_{tot}. \end{array} \right.$$

The value of the fields $\vec{H}$ and $\vec{E}$ at any point $y \notin \Gamma$ can be recovered from the only knowledge of $\vec{j}$ and $\vec{m}$ using the Maxwell representation theorem

$$\left\{ \begin{array}{l} \vec{E}(y) = \vec{Rm}(y) + i\omega\mu\vec{Sj}(y), \\ \vec{H}(y) = \vec{Rj}(y) - i\omega\epsilon\vec{Sm}(y), \end{array} \right. \tag{3}$$

where we denote by R and S the integral operators

$$\begin{cases} \vec{Rj}(y) = \displaystyle\int_\Gamma \vec{grad}_y G(|y-x|) \wedge \vec{j}(x) dx, \\ \vec{Sj}(y) = \displaystyle\int_\Gamma G(|y-x|)\vec{j}(x) + \frac{1}{k^2} \vec{grad}_y G(|y-x|) div_\Gamma \vec{j}(x) dx, \end{cases}$$

and $G(r) = \frac{e^{ikr}}{4\pi r}$ is the Green's function. Using equations (3) in Ω, we obtain

$$\begin{cases} (\vec{R_e m} + \vec{R_i m}) + i\omega(\mu_e \vec{S_e j} + \mu_i \vec{S_i j}) = -\vec{E}_{inc}, \\ (\vec{R_e j} + \vec{R_i j}) - i\omega(\epsilon_e \vec{S_e m} + \epsilon_i \vec{S_i m}) = -\vec{H}_{inc} \end{cases} \quad on\ \Gamma.$$

Finally, testing these equations with tangential test functions $(\vec{j}^t, \vec{m}^t)$, we derive the variational formulation:

find $(\vec{j}, \vec{m})$ such that for all $(\vec{j}^t, \vec{m}^t)$, we have

$$\begin{cases} \displaystyle\int_\Gamma (\vec{R_e m} + \vec{R_i m}).\vec{j}^t + i\omega \int_\Gamma (\mu_e \vec{S_e j} + \mu_i \vec{S_i j}).\vec{j}^t = -\int_\Gamma \vec{E}_{inc}.\vec{j}^t, \\ \displaystyle\int_\Gamma (\vec{R_e j} + \vec{R_i j}).\vec{m}^t - i\omega \int_\Gamma (\epsilon_e \vec{S_e m} + \epsilon_i \vec{S_i m}).\vec{m}^t = -\int_\Gamma \vec{H}_{inc}.\vec{m}^t. \end{cases} \tag{4}$$

For perfectly conducting objects, $\vec{m}$ and $\vec{m}^t$ are null and (4) simplifies

$$i\omega \int_\Gamma (\mu_e \vec{S_e j}).\vec{j}^t = -\int_\Gamma \vec{E}_{inc}.\vec{j}^t, \tag{5}$$

that is known as Electric Field Integral Equation (EFIE). This formulation can be used for arbitrary geometries, for instance those with cavities, disconnected parts, breaks on the surface. For closed and perfectly conducting targets, it is possible to derive the Magnetic Field Integral Equation (MFIE)

$$\int_\Gamma (\vec{R_e j} \wedge \vec{\nu}).\vec{j}^t + \frac{1}{2}\int_\Gamma \vec{j}.\vec{j}^t = -\int_\Gamma (\vec{H}_{inc} \wedge \vec{\nu}).\vec{j}^t. \tag{6}$$

This formulation suffers from interior cavity resonances and does not have unique solution. The problem of nonuniqueness of the solution can be solved by combining linearly (5) and (6). The resulting formulation is known as Combined Field Integral Equation (CFIE) and has the form $CFIE = \alpha EFIE + (1-\alpha)\frac{i}{k}MFIE$; its use is mandatory for closed targets. Although any value can be setup for the parameter α, the choice $\alpha = 0.2$ is usually recommended. A thorough presentation of integral equations in electromagnetism can be found in [40], from which we inherit the notations.

3 Matrix-free iterative solution of boundary integral equations

For homogeneous or layered homogeneous dielectric bodies, the Galerkin method discretizes equation (5) on the surface of the object and at the discontinuous interfaces between two different materials. The induced current is expanded into a set of basis functions, the Rao-Wilton-Glisson basis [34], and the surface of the radiating object is meshed using triangular facets like for standard finite element methods. The discretization gives rise to dense and complex linear systems of equations that are symmetric for the EFIE, unsymmetric for the CFIE and the MFIE. Each unknown of the linear system represents the vectorial flux across an edge in the mesh. The total number n of unknowns is given by the number of interior edges which is about one and a half times the number of triangular facets. The value of n grows linearly with the dimension of the object and, for a given target, quadratically with the frequency of the problem. In fact, when the frequency increases, the edge length has to decrease to be in the order of $\lambda/10$ for a correct representation of the oscillation of the field and the singularities of the scatterer. The coefficient matrix of the resulting linear system is assembled triangle by triangle and its generic entry is associated with the interaction of a pair of triangles in the mesh. The right-hand side depends on the frequency and the illuminating angle; in monostatic radar cross section (RCS) simulations, systems with one single right-hand side are solved, whereas the bistatic RCS calculation requires the solution for multiple right-hand sides, ranging over the complete set of directions between the transmitter and the receiver. For many years, Gaussian elimination has been the method of preference for solving large dense systems issued from boundary integral equations in industry, mainly because of its reliability and predictability in terms of cost [17]. Although efficient out-of-core direct solvers have been developed for this problem class [1, 13], the huge storage requirement can be a severe limit to the viability of integral equation methods for solving high-frequency problems in electromagnetism. Iterative solvers, namely modern Krylov methods, can solve the bottleneck of memory but have to be used in combination with fast methods for the matrix-vector products and robust preconditioners. The condition number of matrices arising from the discretization of EFIE can grow like $p^{1/2}$, where p denotes the size of the scatterer in terms of the wavelength, and linearly with the number of points per wavelength [14]. As a result, Krylov methods scale as $\mathcal{O}(n^{0.5})$ and preconditioning is crucial to accelerate convergence. In Figure 1 we display the eigenvalue distribution of the discretized matrix for a model problem that is representative of the general case. The presence of many isolated eigenvalues in the spectrum, some with large negative real part is unfavorable for a rapid convergence of Krylov solvers. The CFIE formulation is much better conditioned and unsymmetric Krylov solvers scale as $\mathcal{O}(n^{0.25})$. In this case basic preconditioners like block Jacobi or incomplete factorizations can be effective.

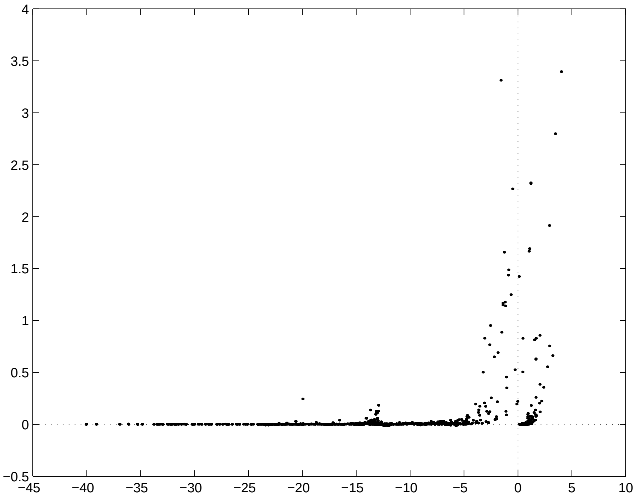


Figure 1: Eigenvalue distribution in the complex plane of the coefficient matrix for a model problem that is representative of the general trend. The problem is modeled using the EFIE formulation.

3.1 The design of the preconditioner

On EFIE many standard preconditioners can perform very poorly because of the indefiniteness of A. Basic preconditioners like the diagonal of A, diagonal blocks, or a band can be effective only when the coefficient matrix has some degree of diagonal dominance [39]. Block diagonal preconditioners are generally more robust than their point-wise counterparts, but may require matrix permutations or renumbering of the grid points to cluster the large entries close to the diagonal. Incomplete factorizations have been successfully used on nonsymmetric dense systems in [38] and hybrid integral formulations in [29], but on the EFIE the triangular factors computed by the factorization can be very ill-conditioned. We illustrate this numerical behavior on a model problem, that is a sphere of 1 meter length, illuminated at 300 MHz. The mesh is discretized with 2430 edges (Figure 2). More severe numerical difficulties can be expected for higher frequencies since the coefficient matrices are less diagonal dominant. In Table 1 we report on experiments with an incomplete Cholesky preconditioner constructed using the pattern of a sparse approximation $\tilde{A}$ of A. We consider increasing values of the density for $\tilde{A}$ as well as various levels of fill-in for the triangular factor L. The general trend is that increasing the fill-in generally produces a much more robust preconditioner than $IC(0)$ applied to a denser sparse approximation of the original matrix. Moreover, $IC(\ell)$ with $\ell \geq 1$ may deliver a good rate of convergence provided the coefficient matrix is not too sparse, as we get closer to LDL^T. However, on indefinite problems the numerical behavior of IC can be fairly chaotic. The factorization of a very sparse approximation (up to 2%) of the coefficient matrix can be stable and deliver a good rate of convergence, especially if at least one level of fill-in is retained. For higher values of density for the approximation of A, the factors may become very ill-conditioned, the triangular solves highly unstable and consequently the preconditioner is useless. As shown in the table, ill-conditioning of the factors is not related to ill-conditioning of A. This behavior has been already observed on sparse real indefinite systems, see for instance [15].

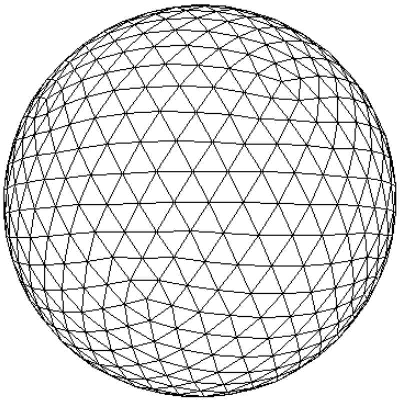


Figure 2: Model problem, a sphere of 1 meter length, illuminated at 300 MHz. The mesh is discretized with 2430 edges.

Approximate inverse methods are generally less prone to instabilities on indefinite systems, and several preconditioners of this type have been proposed in electromagnetism (see for instance [2, 10, 12, 28, 37, 42]). Approximate inverses can be computed in factorized or unfactorized form,

Density of $\tilde{A} = 2\%$				
IC(level)	Density of L	$\kappa_\infty(L)$	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 $\tilde{A} = 4\%$				
IC(level)	Density of L	$\kappa_\infty(L)$	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 $\tilde{A} = 6\%$				
IC(level)	Density of L	$\kappa_\infty(L)$	GMRES(30)	GMRES(50)
$IC(0)$	6.0%	$8 \cdot 10^{11}$	–	–
$IC(1)$	18.8%	$5 \cdot 10^{11}$	–	–
$IC(2)$	29.6%	$7 \cdot 10^4$	–	–

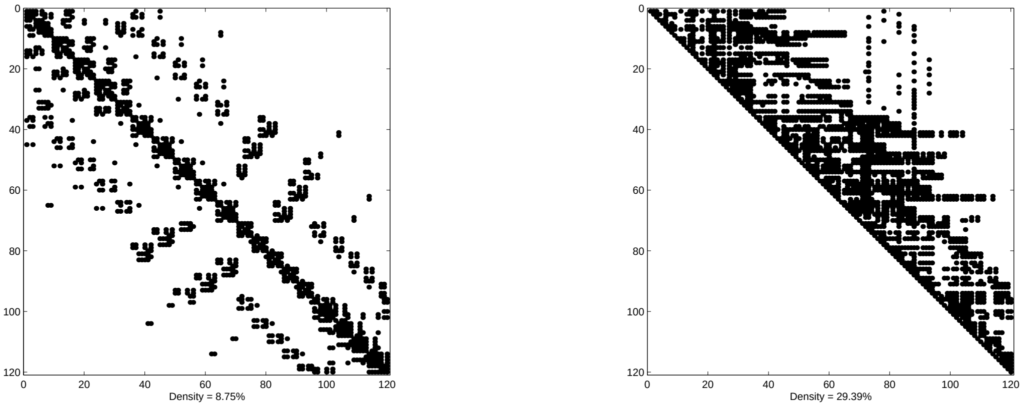
Table 1: Number of iterations of GMRES varying the sparsity level of $\tilde{A}$ and the level of fill-in on the model problem of Figure 2 ($n = 2430, \kappa_\infty(A) \approx \mathcal{O}(10^2)$). The iterations are stopped when initial residual is reduced by five orders of magnitude. The symbol '-' means that convergence was not obtained after 500 iterations.

depending on the fact that they are expressed as a single matrix or the product of two (or more) matrices. For a small sphere, we display in Figure 3 the sparsity pattern of A^{-1} (on the left) and L^{-1}, the inverse of its Cholesky factor (on the right), respectively, where all the entries smaller than 5.0×10^{-2} have been dropped after a symmetric scaling such that $\max_i |a_{ji}| = \max_i |\ell_{ji}| = 1$. The inverse factors can be totally unstructured as shown in Figure 3(b). In Figure 4 we plot the magnitude of the entries in the first column of A^{-1} (on the left) and L^{-1} (on the right), respectively, with respect to their row index. These plots indicate that for unfactorized methods any dropping strategy, either static or dynamic, may be very difficult to tune as it can easily discard relevant information and potentially lead to a very poor preconditioner. Selecting too small a threshold would retain too many entries and lead to a fairly dense preconditioner. Because of these issues, finding the appropriate threshold to enable a good trade-off between sparsity and numerical efficiency is challenging and very problem-dependent. As a result, the performance of the AINV [6] and FSAI [27] preconditioners are disappointing (see Table 2).

Density of M	GMRES(m)		
	m=50	m=110	m=∞
4%	–	–	–
6%	–	–	259
8%	–	–	229
10%	–	–	216

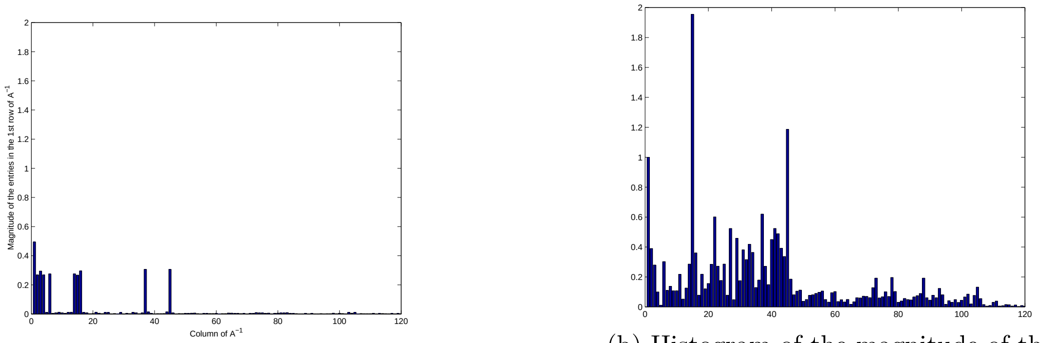
Table 2: Number of iterations of GMRES preconditioned by AINV to reduce the initial residual by five orders of magnitude. The symbol '-' means that convergence was not obtained after 500 iterations.

Owing to the rapid decay of the discrete Green's function, the location of the large entries in the inverse matrix exhibit some structure as shown in Figure 3(a). In addition, only a very small number of its entries have large magnitude compared to the others that are much



(a) Sparsity pattern of *sparsified*(A^{-1}) (b) Sparsity pattern of *sparsified*(L^{-1})

Figure 3: Sparsity patterns of the inverse of A (on the left) and of the inverse of its lower triangular factor (on the right), where all the entries whose relative magnitude is smaller than 5.0×10^{-2} are dropped. The test problem, representative of the general trend, is a small sphere.



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

(b) Histogram of the magnitude of the entries in the first column of the inverse of a factor of A

Figure 4: Histograms of the magnitude of the entries of the first column of A^{-1} and its lower triangular factor. A similar behavior has been observed for all the other columns. The test problem, representative of the general trend, is a small sphere.

smaller (Figure 4(a)). It means that a very sparse matrix is likely to retain the most relevant contributions to the exact inverse. The approximate inverse can be computed as the matrix M that minimizes the Frobenius-norm of the error matrix $\|I - AM\|_F$, subject to certain sparsity constraints. The Frobenius norm is chosen since it allows the decoupling of the constrained minimization problems into n independent linear least-squares problems, one for each column (resp. row) of M when preconditioning from the right (resp. left). The independence of these least-squares problems follows immediately from the identity:

$$\|I - AM\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_{\bullet j}\|_2^2, \tag{7}$$

where e_j is the jth canonical unit vector, $m_{\bullet j}$ is the column vector representing the jth column of M. Both the construction and the application of M are embarrassingly parallel. Two different approaches can be followed for the selection of the sparsity pattern of M: an adaptive technique that dynamically tries to identify the best structure for M and a static technique, where the pattern of M is prescribed *a priori* based on some heuristics. The idea is to keep M reasonably sparse while trying to capture the large magnitude entries of the inverse, which are expected to contribute the most to the quality of the preconditioner. A static approach that requires an *a priori* nonzero pattern for the preconditioner, introduces significant scope for parallelism and has the advantage that the memory storage requirements and the computational cost for the setup phase are known in advance. However, it can be very problem dependent. A dynamic approach is generally effective but very expensive. These methods usually start with a simple initial guess, like a diagonal matrix, and then improve the pattern until a criterion of the form $\|Am_j - e_j\|_2 < \varepsilon$ (for each j) is satisfied for a given $\varepsilon > 0$, e_j being the jth column of the identity matrix, or until a maximum number of nonzeros in the jth column m_j of M has been reached. In Table 3 we report on results on the model problem using the MI12 routine of the Harwell Subroutine Library [25] which implements the adaptive procedure described in [20]. Provided the preconditioner is dense enough, SPAI is quite effective in reducing the number of iterations. However, the construction of the preconditioner with a density of 10% takes more than one day using one processor on a SGI Origin 2000. On dense problems from electromagnetics, a good pattern for M can be prescribed *a priori* exploiting the rapid decay of the discrete Green's function which induces a similar structure in A and its inverse. The pattern can be computed in advance using graph information from the near-field part of the matrix that is explicitly computed and available in the Fast Multipole Method [2, 28], or graph information from the mesh, by selecting for the jth column of the approximate inverse edge j and its qth level nearest-neighbours [33]. In the next section we introduce a preconditioning algorithm constructed on top of a Frobenius-norm minimization method and aimed at enhancing its robustness. We present the main ideas behind the method and illustrate the results of numerical experiments for the solution of realistic industrial problems in combination with the Fast Multipole Method.

Density of M	GMRES(m)				
	m=10	m=30	m=50	m=80	m=110
2%	–	–	–	–	–
6%	–	–	194	122	93
10%	–	283	100	68	68

Table 3: Number of iterations required by different Krylov solvers preconditioned by SPAI to reduce the initial residual by five orders of magnitude. The symbol '-' means that convergence was not obtained after 500 iterations.

3.2 Algebraic two-grid spectral preconditioning

In this section we perform a study with a two-grid spectral preconditioning algorithm that is built on top of a given preconditioner M, meaning that we want to solve $MAx = Mb$ for left preconditioning, or $AMy = b$, $x = My$ for right preconditioning. We inherit the basic idea from standard two-grid cycles that are fully defined by the selection of the coarse grid, the choice of the smoother applied on the fine grid and of the grid transfer operators to move between fine and coarse grid. We use the preconditioner M to define a weighted stationary method that implements the smoother and the coarse space is defined using a Galerkin formula $A_c = R^H AP$, where R and P are the restriction and the prolongation operator, respectively. We consider an additive two-grid cycle, although schemes based on standard multiplicative cycles can be derived as well [11]. After μ smoothing steps, we project by means of the operator R the residual into the coarse subspace and we solve the coarse space error equation involving $A_c = R^H AP$. Finally, we prolongate back the error using the operator P in the original space and smooth again the new approximation.

These two contributions are summed together for the solution update. We define the grid-transfer operators algebraically, and select $P = V_\varepsilon$ to be the set of right eigenvectors associated with the set of eigenvalues λ_i of MA with $|\lambda_i| \leq \varepsilon$. For the restriction operator, a natural choice is to select $R = W$ orthogonal to V_ε (i.e. $W^H V_\varepsilon = I$). Following [19, 41], we define the filtering operators using the grid transfer operators as $(I - V_\varepsilon W^H)$. Thus we have $(I - V_\varepsilon W^H)V_\varepsilon = 0$ and the filtering operator is supposed to remove all the components of the preconditioned residual in the V_ε directions. The coarse grid correction computes components of the residual in the space spanned by the eigenvectors associated with the few selected small eigenvalues, while at the end of the smoothing step the preconditioned residual is filtered so that only components in the complementary subspace are retained. The preconditioner constructed using this scheme depends on A, M, μ, $\omega > 0$. For the sake of simplicity of exposure, we will simply denote it by $M_{Add}(A, M)$. We present a sketch of the algorithm in Figure 5. It takes as input a vector r that is the residual vector we want to precondition, and returns as output the preconditioned residual vector z.

```
set $z^1 = 0$

for k=1, $iter$ do

        1. Compute the residual
           $s^k = r - Az^k$
        2. Compute the high and low frequency corrections
           (a) High frequency correction
               $e_1^{k,0} = 0$
               for j=1, $\mu$ do
                 $e_1^{k,j} = e_1^{k,j-1} + \omega M(s^k - Ae_1^{k,j-1})$
               end for
               $c_1^k = (I - V_\varepsilon W^H)e_1^{k,\mu}$
           (b) Low frequency correction
               $c_2^k = V_\varepsilon A_c^{-1} W^H s^k$
        3. Update the preconditioned residual
           $z^{k+1} = z^k + c_1^k + c_2^k$

end for

$z = z^{iter}$
```

Figure 5: Additive two-grid spectral preconditioning. The algorithm.

Proposition 1 *Let W be such that $A_c = W^H A V_\varepsilon$ has full rank and satisfies $(I - V_\varepsilon W^H)V_\varepsilon = 0$, the preconditioning operation described in the algorithm of Figure 5 can be written in the form $z = M_{Add}\ r$. After one cycle ($iter = 1$), M_{Add} has the following expression:*

$$M_{Add} = V_\varepsilon A_c^{-1} W^H + (I - V_\varepsilon W^H)(I - (I - \omega M)^\mu)A^{-1}. \tag{8}$$

Proof

Hereby we remove the superscript k in the algorithm of Figure 5, as we analyze the case $iter = 1$. The statement easily follows from the observation that the smoothing step generates sequence of vectors of the form $e_1^j = (I - \omega MA)e_1^{j-1} + \omega Mr$, that can be written $e_1^\mu = [I - (I - \omega MA)^\mu]A^{-1}r$ because $e_1^0 = 0$.

■

In the following proposition, we examine the effect of M_{Add} on the spectrum of the preconditioned matrix. We analyse the case when MA has a complete set of eigenvectors, that is MA is diagonalizable in the form

$$MA = V\Lambda V^{-1}, \tag{9}$$

with $\Lambda = diag(\lambda_i)$, where $|\lambda_1| \leq \ldots \leq |\lambda_n|$ are the eigenvalues and $V = (v_i)$ the associated right eigenvectors. Let V_ε be the set of right eigenvectors associated with the set of eigenvalues λ_i with $|\lambda_i| \leq \varepsilon$.

Proposition 2 *The preconditioner M_{Add} defined by Proposition 1 is such that the preconditioned matrix $M_{Add}A$ has eigenvalues:*

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

<u>Proof</u>
Let $V = (V_\varepsilon, V_{\bar{\varepsilon}})$, where $V_{\bar{\varepsilon}}$ is the set of $(n-k)$ right eigenvectors associated with eigenvalues $|\lambda_i| > \varepsilon$ of MA. Let $D_\varepsilon = diag(\lambda_i)$ with $|\lambda_i| \leq \varepsilon$ and $D_{\bar{\varepsilon}} = diag(\lambda_j)$ with $|\lambda_j| > \varepsilon$. Then the following relations hold: $M_{Add}AV_\varepsilon = V_\varepsilon$, and $M_{Add}AV_{\bar{\varepsilon}} = V_{\bar{\varepsilon}}(I - (I - \omega D_{\bar{\varepsilon}}))^\mu + V_\varepsilon C_{Add}$ with $C_{Add} = A_c^{-1}W^H AV_{\bar{\varepsilon}} - W^H V_{\bar{\varepsilon}}(I - (I - \omega D_{\bar{\varepsilon}})^\mu$. This can be written:

$$M_{Add}AV = V \begin{pmatrix} I & C_{Add} \\ 0 & I - (I - \omega D_{\bar{\varepsilon}})^\mu \end{pmatrix}$$

■

The spectral properties of the preconditioner have been derived from the algorithm in Figure 5 where only one outer iteration is considered. For $\omega = 1$, implementing "*iter*" outer steps leads to a spectral transformation that is defined by the following proposition.

Proposition 3 *The preconditioner M_{Add} defined by the algorithm of Figure 5 with iter ≥ 1 and $\omega = 1$ is such that the preconditioned matrix $M_{Add}A$ has eigenvalues:*

$$\left\{ \begin{array}{lll} \eta_i = 1 & \text{if} & |\lambda_i| \leq \varepsilon, \\ \eta_i = 1 - (1 - \lambda_i)^{iter \times \mu} & \text{if} & |\lambda_i| > \varepsilon. \end{array} \right.$$

<u>Proof</u>

The assertion is true for *iter* $= 1$. Suppose that after *iter* cycles the eigenvalues are transformed as follows:

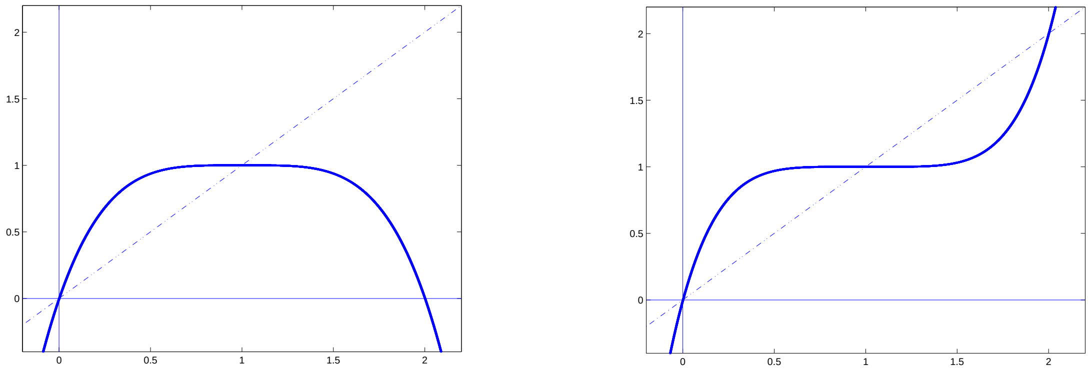
$$\left\{ \begin{array}{lll} \delta_i = 1 & \text{if} & |\lambda_i| \leq \varepsilon, \\ \delta_i = 1 - (1 - \lambda_i)^{iter \times \mu} & \text{if} & |\lambda_i| > \varepsilon. \end{array} \right.$$

One additional cycle with M_{Add} gives the following spectral transformation: $\eta_i = 1 - (1 - \delta_i)^\mu = = 1 - (1 - (1 - (1 - \lambda_i)^{iter \cdot \mu}))^\mu = 1 - ((1 - \lambda_i)^{iter \cdot \mu})^\mu = 1 - (1 - \lambda_i)^{(iter+1) \cdot \mu}$. Thus the assertion follows by induction on the number of outer steps.

■

Remark 1 *Let B and C be two square matrices, it is known that the eigenvalues of AB are the same as those of BA. Consequently using M_{Add} either as a left or as a right preconditioner will give the same spectrum for the preconditioned matrix.*

In Figure 6 we depict the spectral transformation for $\omega = 1.0$ in the case MA has real eigenvalues. We can observe that for $\mu > 1$ all the eigenvalues lying in the open disk of radius one centered in (1,0) are attracted toward (1,0); the clustering is stronger for larger μ. The eigenvalues out of this disk are spread away from (1,0) potentially damaging the convergence. We see that those having magnitude larger than two become negative for μ even, they stay positive for μ odd. 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. However, the most suitable choice of this parameter depends on the eigenvalue distribution and it is difficult to define in advance.



(a) μ even (=4)

(b) μ odd (=5)

Figure 6: Shape of the polynomial that governs the eigenvalues distribution of the two-grid spectral preconditioner for $\omega = 1$.

3.2.1 Discussions

The Frobenius-norm minimization preconditioner succeeds in clustering most of the eigenvalues far from the origin. This can be observed in Figure 7 where we see a big cluster near (1.0,0.0) in the spectrum of the preconditioned matrix for a small test problem that is representative of the general trend. This kind of distribution is highly desirable to get fast convergence of Krylov solvers. Nevertheless the eigenvalues nearest to zero potentially can slow down convergence. The small eigenvalues are difficult to remove even if we increase the number of nonzeros. This scenario is typical of many standard preconditioners and applications, where the preconditioned matrix results in a fairly well clustered spectrum close to one but often a few small eigenvalues are left close to the origin and are difficult to remove by tuning the parameters that control the preconditioner. This may happen in particular for indefinite problems when some eigenvalues, in their migration process towards the real axis under the action of a good preconditioner, can group close to the origin of the complex plane. It is known that the convergence of Krylov methods for solving the linear system depends to a large extent on the eigenvalue distribution. In exact arithmetic, the solution of $Ax = b$ lies in the space $K_m(A, b)$, where m is the degree of the minimal polynomial of A. If $\lambda_1, ..., \lambda_d$ are the distinct eigenvalues of A, we have $m = \sum_{j=1}^{d} m_j$ where m_j is the index of λ_j. The smaller the degree of the minimal polynomial, the faster the expected rate of convergence. If the eigenvalues are not distinct but the diameters of the clusters are small enough, each cluster behaves numerically like a single eigenvalue, and we may expect a few iterations of a Krylov method to converge to reasonably accurate approximations. Since Krylov methods build a polynomial expansion of A that is equal to one at zero and has to approximate zero on the set of eigenvalues, the presence of

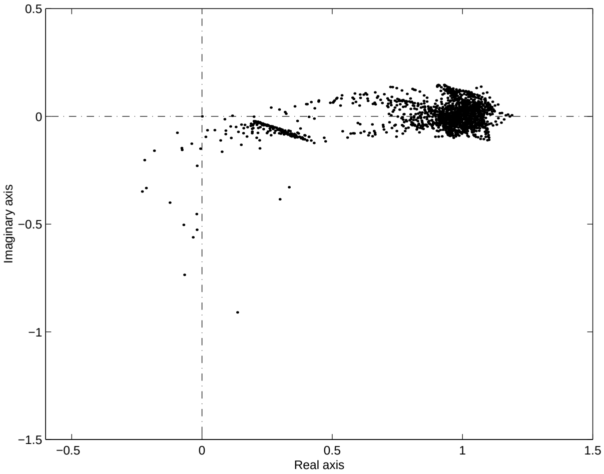


Figure 7: Eigenvalue distribution for the coefficient matrix preconditioned by the Frobenius-norm minimization method on a small test problem, that is representative of the general trend.

clusters close to zero is undesirable because a polynomial of low-degree cannot approximate zero at such many points. To cure this, several techniques have been proposed in the past few years, mainly to improve the convergence of GMRES. There are essentially two different approaches for exploiting information related to the smallest eigenvalues. The first idea is to compute a few, k say, approximate eigenvectors of MA corresponding to the k smallest eigenvalues in magnitude, and augment the Krylov subspace with those directions. This approach is referred to as the *augmented subspace approach* (see [7, 30, 31, 32, 43]). The approximate eigenvectors can be chosen to be Ritz vectors from the Arnoldi process. The standard implementation of the restarted GMRES algorithm is based on the Arnoldi process, and spectral information of MA might be recovered during the iterations. The second idea exploits spectral information gathered during the Arnoldi process to determine an approximation of an invariant subspace of A associated with the eigenvalues nearest the origin, and uses this information to construct a preconditioner or to adapt an existing one [3, 9, 18, 26, 36]. The novelty of the approach presented in this paper compared to other spectral preconditioning techniques like those described in [3, 9, 26] is that the proposed method induces a global deflation in the spectrum of the preconditioned matrix. The combined effect of the coarse grid correction that shifts the smallest eigenvalues to one, coupled with the action of the Richardson accelerator that clusters the rest of the spectrum can be beneficial in the presence of outliers or many isolated eigenvalues. The price to pay are additional matrix-vector products. Applying M_{Add} requires $(\mu - 1) \times iter$ matrix-vector products with M, and $(\mu \times iter - 1)$ products by A. The filtering operation costs $2nk$ operations and the low frequency correction costs $2nk + k^2$, where k is the size of the coarse space. As $\mu, iter, k$ are $\mathcal{O}(1)$, the overall cost of applying M_{Add} if $\mathcal{O}(n)$. Under the assumption that the convergence of Krylov solver is mainly governed by the eigenvalue distributions (i.e. $iter \times \mu$ given), from Proposition 3 follows that it is more efficient to use only one two-grid cycle with $(iter \times \mu)$ smoothing steps rather than $iter$ two-grid cycles with μ smoothing steps per cycle.

3.3 Implementation of the Fast Multipole Method

In the numerical experiments reported in this paper, the preconditioned Krylov method is combined with multipole techniques for the matrix-vector products. In the last twenty years, fertile research efforts have led to the development of fast methods for performing approximate matrix-vector products with boundary integral operators (see e.g. [4, 5, 8, 22, 23, 24]), including reliable implementations on distributed memory computers [21, 40]. In our experiments, we use a multilevel implementation of the Fast Multipole Method (FMM). The algorithm, introduced by Greengard and Rokhlin [22, 35], enables us to compute approximate matrix-vector products with dense matrices arising from the discretization of boundary integral operators in $\mathcal{O}(n \log n)$ arithmetic operations and $\mathcal{O}(n \log n)$ memory storage. The method is approximate in the sense that the relative error in the matrix-vector computation is of the order of $\epsilon = 10^{-3}$. The main idea is to compute interactions amongst degrees of freedom in the mesh at different levels of accuracy depending on their physical distance. In a first stage of the algorithm, the mesh of the discretized object is partitioned by recursive subdivision into disjoint aggregates of small size compared to the wavelength, each roughly formed by an equal number of separate triangles. The near-field interactions between triangles belonging to neighbouring boxes are computed explicitly using truncated series expansions of the Green's function. As the neighbourhood of a box is defined by the box itself and its 26 adjacent neighbours (eight in 2D), this computation reduces to a sparse matrix-vector product operation. The far-field interactions between far-away triangles are approximated using a separable expression of the discrete Green's function, which separates the Green's function into two sets of terms that depend only on the observation point x and the source point y, respectively. This expression can be used only if x and y are distant enough in the mesh and cannot be used for the near-field. The cumulative effect of the interaction corresponding to nodes belonging to each far-away box is concentrated in a single coefficient that is called *multipole coefficient*. Multipole coefficients of far-away boxes are summed together to compute local coefficients for the observation box, and the total effect of the far field on each triangle is evaluated from the local coefficients (see Figure 8 for a 2D illustration). Local and multipole coefficients are pre-computed, stored and reused many times in the matrix-vector calculation. The final cost of this operation is reduced to $\mathcal{O}(n^{3/2})$ in the basic one-level algorithm. In the numerical experiments, we consider the parallel multilevel implementation of the FMM described in [16, 40]. In the multilevel algorithm, the box-wise partitioning of the obstacle is carried out until the size of the smallest box is of the order of $\lambda/2$ and a tree-structured data is used at all levels to record

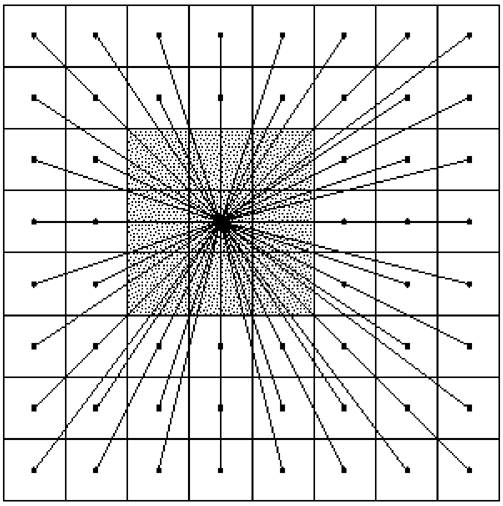


Figure 8: Interactions in the one-level FMM. For each observation box, the interactions with the gray neighbouring boxes are computed directly. The contribution of far away boxes are computed approximately. Multipole expansions of far away boxes are translated to local expansions for the observation box and the total field induced by far away boxes is evaluated from local expansions.

the non-empty boxes of the partitioning. The resulting tree is called the *oct-tree* (see Figure 9) and its leaves are generally referred to as the *leaf boxes*. Multipole coefficients are computed for all boxes starting from the lowest level of the oct-tree and recursively for each parent box by summing together multipole coefficients of their children. In the matrix-vector calculation, the tree is explored from the root to the leaves, and at each level the one level multipole method is used to treat interactions between non-neighbouring domains that have not yet been treated at an upper level. At the lowest level of the tree, that is for the leaf-boxes, the interactions are computed explicitly. In the multilevel algorithm, the computational cost is $\mathcal{O}(n \log n)$.

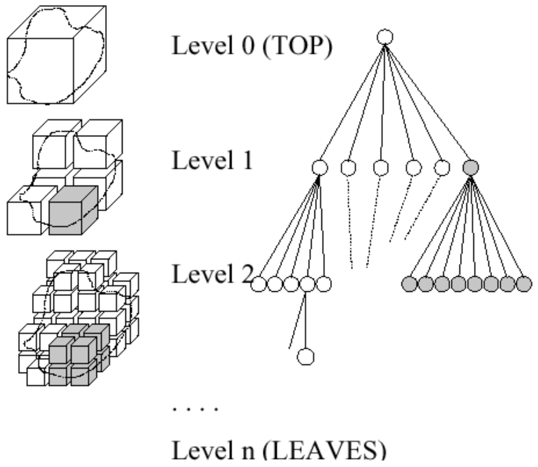


Figure 9: The oct-tree in the FMM algorithm. The maximum number of children is eight. The actual number corresponds to the subset of eight that intersect the object. Courtesy of EADS.

4 Numerical experiments.

We report on results of experiments on a set of real-life industrial problems provided by EADS. The two-grid preconditioning scheme described in the earlier section is applied on top of a sparse approximate inverse preconditioner based on Frobenius-norm minimization which plays the role of the smoother M. This class of approximate inverses is an interesting alternative to standard smoothers because it is inherently parallel, ordering independent and locally adaptive. A relevant issue is the selection of a good sparsity pattern for M. In this work, we compute the approximate inverse from the near-field part $\tilde{A}$ of A, meaning that we solve $\|I - \tilde{A}M\|_F^2$ for right preconditioning. We prescribe in advance the nonzero structure $J = \{i \in [1, n] \; s.t. \; m_{ij} = 0\}$ for the jth column of M using geometric information from the mesh. In the context of FMM, we adopt the following criterion: the nonzero structure of each column of the preconditioner is defined by retaining edges within the associated leaf box and one level of neighbouring boxes. The near-field matrix $\tilde{A}$ is constructed computing interactions of triangles within the same leaf box and up-to two levels of neighbouring boxes. The least-squares solution involves only the columns of $\tilde{A}$ indexed by J; we indicate this subset by $\tilde{A}(:, J)$. As $\tilde{A}$ is sparse, many rows in $\tilde{A}(:, J)$ are null and do not affect the solution of the least-squares problems (7). If we denote by I the set of indices corresponding to the nonzero rows in $\tilde{A}(:, J)$ and define by $\hat{A} = \tilde{A}(I, J)$, $\hat{m}_j = m_j(J)$, and $\hat{e}_j = e_j(J)$, the actual *reduced* least-squares problems to solve are

$$min \|\hat{e}_j - \hat{A}\hat{m}_j\|_2, j = 1, ..., n. \tag{10}$$

Problems (10) have much smaller size than problems (7) and can be effectively solved by a dense QR factorization. We observe that the least-square problems corresponding to edges within the same box are identical because they are defined using the same nonzero structure and the same set of entries of A. It means that the preconditioner has a block structure and we only have to compute one QR factorization per leaf box. Thus the construction cost of the approximate inverse scales as $\mathcal{O}(n)$, and its application is a sparse $\mathcal{O}(n)$ matrix-vector product operation. Details of the implementation of the approximate inverse in the EADS code as well as results on the parallel and numerical scalability are reported in [10]. In the numerical experiments, we consider a right preconditioned GMRES method; iterations start from the zero vector and are stopped when the normwise backward error $\frac{||r||}{||b||}$, where r denotes the residual and b the right-hand side of the linear system, is smaller that 10^{-3}. This tolerance is accurate for engineering purposes, as it enables the correct reconstruction of the radar cross section of the object and is in the order of accuracy of the approximate matrix-vector products when using FMM. We use highly accurate FMM for the standard matrix-vector product operation within GMRES, and less accurate FMM for the extra matrix-vector products involved in the smoother of the preconditioning operation. The price to pay for the accuracy is mainly computational time, high accuracy also means more time consuming to compute. The parallel runs have been performed in single precision arithmetic on sixteen processors of a HP-Compaq Alpha server, a cluster of Symmetric Multi-Processors. In the next sections we describe the set of test problems and the results of numerical experiments.

4.1 Test examples

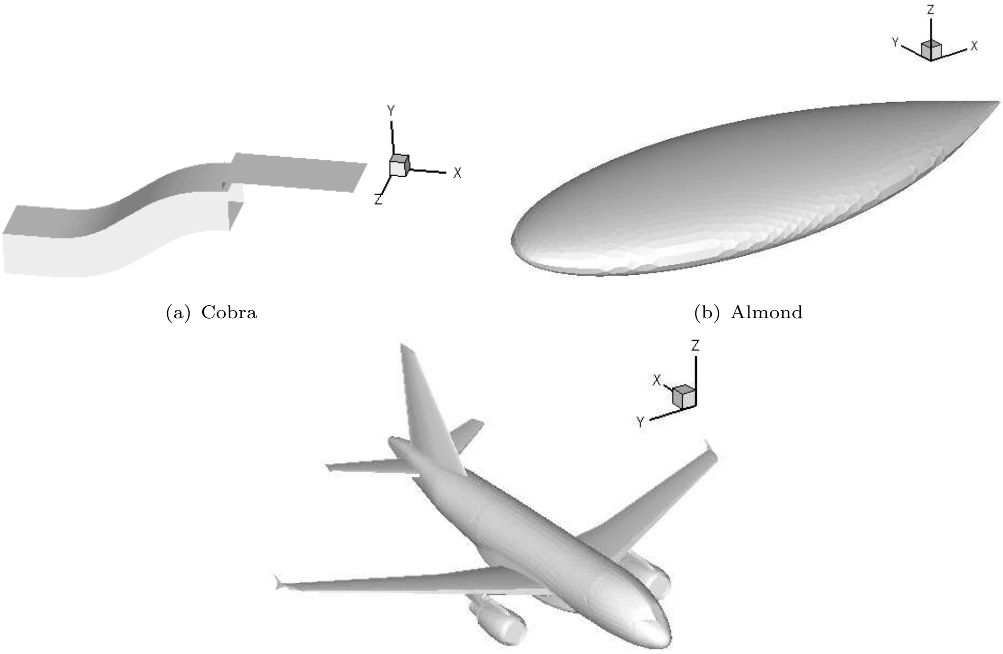
For the numerical experiments, we consider the following three test problems provided by EADS:

- an Airbus aircraft (Figure 10(c)) discretized with 23796 degrees of freedom. The size is 1.8 m × 1.9 m × 0.65 m. This mesh is used in realistic industrial simulations.
- the Almond problem (Figure 10(b)), discretized with 104793 edges. The size is 2.5 m. It was an official test case for the JINA 2002 conference (12th International workshop on Antenna design, Nice 2002).
- the Cobra problem (Figure 10(a)), a standard test case in the electromagnetic community. The geometry represents an air intake and has size 67.9 cm × 23.3 cm × 11 cm. As the surface is open, it can only modeled using the EFIE formulation. The mesh has 60695 degrees of freedom.

In Table 4 we report on the physical characteristics of each problem, that are the frequency of the illuminating wave and the size of the scatterer with respect to the wavelength measured as the diameter (in wavelengths) of the smallest sphere that encompasses the object. As the solution of Maxwells equations is highly oscillating, for an accurate representation the average length of the edges varies in the range between $\lambda/6$ and $\lambda/10$. Among all the possible right-hand sides we have selected those that are the most difficult to solve in order to better illustrate the robustness and the efficiency of our preconditioner. The condition number $\kappa_\infty(A)$ for the test problems is $\mathcal{O}(10^2)$ for CFIE and $\mathcal{O}(10^4)$ for EFIE. Although the test problems are not ill-conditioned, they are difficult to precondition as illustrated in Section 3.1

Geometry	Size	Frequency	λ	p
Aircraft	23676	2.3 GHz	13.0 cm	14
Almond	104793	2.6 GHz	11.5 cm	22
Cobra	60695	10.0 GHz	3.0 cm	24

Table 4: Physical characteristics of the test problems.



(a) Cobra

(b) Almond

(c) Airbus

Figure 10: Mesh associated with test examples. Courtesy of EADS.

4.2 Numerical results

In Tables 5-6 we report on the number of iterations and the elapsed-time for different sizes of the coarse space and an increasing number of smoothing steps. For the two tests examples we show results with restarted and full GMRES. For these experiments we setup $\omega = \frac{2}{3}\lambda_{max}(MA)$, that is an overall good choice. In this table we also display the number of iterations and the elapsed time of GMRES with only the sparse approximate inverse. The gain in time varies from fourteen to infinity with GMRES(10) and between two and three with GMRES(∞). From a numerical point of view we observe that the larger the coarse space, the better the preconditioner; the number of GMRES iterations decreases when the number of smoothing steps is increased. Furthermore, the gain is larger if restarted GMRES is considered than if full GMRES is used as solver. In particular on the Almond with GMRES(10) and less that fifty eigenvectors the only way to get convergence is to perform few step of smoothing; with fifty eigenvectors the gain introduced by the smoothing iterations is tremendous (i.e. larger than 21). On the Cobra problem, using fifteen eigenvectors the gain is far larger than two with GMRES(10), and close to two for full GMRES. Not only the number of iterations is significantly reduced but also the solution time. On the Almond problem, using fifty eigenvectors and GMRES(10) we gain a factor of twelve in elapsed-time when we increase the number of smoothing steps from one to three. We mention that on large electromagnetic problems (of size larger than 0.5 Million unknowns) the use of small restarts is recommended in order to save the heavy cost of reorthogonalization and reduce the final solution cost. The choice of a small restart is often dictated by memory constraints [10]. With full GMRES, the number of iterations is significantly decreased but the total solution cost is likely to only slightly decrase when the number of smoothing step is large as the preconditioner is expensive to apply. The optimal selection of the size of the coarse space and of the number of smoothing steps remains an open question, and the choice mainly depends on the clustering properties of the initial preconditioner. The coarse-grid correction and the smoothing mechanism are complementary components that balance each other. On the Cobra problem, for instance, using three smoothing steps and five eigenvectors we obtain better convergence rate but similar computational time than using one smoothing step but fifteen eigenvectors; on the Almond problem, the solution cost of GMRES(10) with three smoothing steps and thirty eigenvectors is similar to the time with two smoothing steps but fifteen eigenvectors. Balancing the reduction of the number of eigenvectors by using more smoothing steps is a desirable feature in the contexts where computing many eigenvalues can become very expensive which is for instance the case in this application for very large problems. 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 500 matrix-vector products and 15 minutes are required for computing 15 eigenvectors on the Cobra problem. This extra-cost is quickly amortized as many right-hand sides have usually to be solved to compute the so called radar cross section. Finally, in Table 7 we report on results of experiments on a real Airbus aircraft discretized with 23796 nodes. Although small, this test problem is difficult to solve for iterative methods. Convergence using GMRES and the Frobenius-norm minimization preconditioner can be achieved only with large values of restart [10]. It can be seen that the use of M_{Add} is very beneficial. It enables to obtain convergence for very small restart and the smoothing mechanism reduces significantly both the number of iterations and the CPU time.

	Cobra problem		

With M_{FROB}	GMRES(10) 2719 iterations (1^h 10^m)
	GMRES(∞) 378 iterations (18^m)		

	$\mu = 1$		
	Dimension of the coarse space		
	5	10	15
GMRES(10)	1458 (42^m)	594 (12^m)	517 (11^m)
GMRES(∞)	262 (9^m)	216 (7^m)	188 (6^m)
	$\mu = 2$		
	Dimension of the coarse space		
	5	10	15
GMRES(10)	471 (18^m)	209 (7^m)	201 (6^m)
GMRES(∞)	161 (8^m)	132 (6^m)	115 (5^m)
	$\mu = 3$		
	Dimension of the coarse space		
	5	10	15
GMRES(10)	281 (12^m)	132 (6^m)	124 (5^m)
GMRES(∞)	120 (7^m)	98 (6^m)	85 (5^m)

Table 5: Experiments with M_{Add} on the Cobra problem.

5 Conclusive remarks

In this work, we have presented an algebraic two-grid spectral preconditioner combined with multipole techniques for solving boundary integral equations in electromagnetism. We use a sparse approximate inverse as a smoother for an algebraic two-grid cycle where the inter-grids operators are based on spectral information from the preconditioned matrix. The preconditioner induces a global deflation of the eigenvalues of the preconditioned matrix which can result in a well clustered spectrum around one. The overall scheme is fairly robust, requires limited memory and is matrix-free as it does not require the explicit computation of all the entries of the coefficient of the linear

Almond problem			
With M_{FROB}	GMRES(10) +3000 iterations GMRES(∞) 242 iterations (14^m)		
$\mu = 1$			
	Dimension of the coarse space		
	10	30	50
GMRES(10)	+3000	+3000	1867 (1h12m)
GMRES(∞)	229 (13^m)	157 (8^m)	132 (6^m)
$\mu = 2$			
	Dimension of the coarse space		
	10	30	50
GMRES(10)	552 (29^m)	245 (14^m)	176 (9^m)
GMRES(∞)	134 (9^m)	92 (6^m)	77 (6^m)
$\mu = 3$			
	Dimension of the coarse space		
	10	30	50
GMRES(10)	216 (16^m)	116 (9^m)	87 (6^m)
GMRES(∞)	97 (9^m)	66 (6^m)	56 (6^m)

Table 6: Experiments with M_{Add} on the Almond problem.

Airbus aircraft problem		
With M_{FROB}	GMRES(10) +4000 iterations	
μ	GMRES(10)	CPU-time
1	1835	1^h07^m
2	807	36^m
3	368	22^m

Table 7: Experiments with M_{Add} on the Airbus aircraft problem.

system to solve. The sparse approximate inverse is computed from the near-field part of the dense coefficient matrix and the pattern is prescribed in advance by using physical information. The numerical results on small and medium size problems from radar cross section calculations illustrate the potential of this method for solving realistic electromagnetic applications. Future work has to assess the viability of the proposed scheme on larger problems especially in the context of multiple right-hand sides systems, where the spectral information can be profitably recovered from the Arnoldi factorization of previous system solving. Apart from electromagnetism, the proposed method can be potentially interesting to test in other application contexts where the coefficient matrix is not completely assembled, as in large least-squares calculations or for solving inner linear systems in nonlinear iterations.

Acknowledgement

This work has been developed using the computing facilities and thanks to the fertile working environment of the CERFACS institute. In particular, grateful thanks are deserved to Emeric Martin for his valuable assistance in the code development. The test problems for the numerical experiments have been provided by EADS.

References

[1] G. Alléon, S. Amram, N. Durante, P. Homsi, D. Pogarieloff, 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.* 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] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel. Adaptively preconditioned GMRES algorithms. *SIAM J. Scientific Computing*, 20(1):243–269, 1999.

[4] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.

[5] M. Bebendorf and S. Rjasanov. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.

[6] M. Benzi and M. Tůma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Scientific Computing*, 19:968–994, 1998.

[7] C. Le Calvez and B. Molina. Implicitly restarted and deflated GMRES. *Numerical Algorithms*, 21:261–285, 1999.

[8] F.X. Canning. The impedance matrix localization (IML) method for moment-method calculations. *IEEE Antennas and Propagation Magazine*, 1990.

[9] B. Carpentieri, I. S. Duff, and L. Giraud. A class of spectral two-level preconditioners. *SIAM J. Scientific Computing*, 25(2):749–765, 2003.

[10] B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand. Combining fast multipole techniques and an approximate inverse preconditioner for large parallel electromagnetics calculations. Technical Report TR/PA/03/77, CERFACS, Toulouse, France, 2003. Accepted for publication in SISC.

[11] B. Carpentieri, L. Giraud, and S. Gratton. Additive and multiplicative two-level spectral preconditioning for general linear systems. Technical Report TR/PA/04/38, CERFACS, Toulouse, France, 2004.

[12] K. Chen. An analysis of sparse approximate inverse preconditioners for boundary integral equations. *SIAM J. Matrix Analysis and Applications*, 22(3):1058–1078, 2001.

[13] 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.

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

[15] E. Chow and Y. Saad. Experimental study of ILU preconditioners for indefinite matrices. *Journal of Computational and Applied Mathematics*, 86:387–414, 1997.

[16] E. Darve. The fast multipole method: Numerical implementation. *J. Comp. Phys.*, 160(1):195–240, 2000.

[17] Alan Edelman. Records in dense linear algebra [online, cited 27 April 2005]. Available from: `http://www-math.mit.edu/~edelman/records.html`.

[18] J. Erhel, K. Burrage, and B. Pohl. Restarted GMRES preconditioned by deflation. *J. Comput. Appl. Math.*, 69:303–318, 1996.

[19] L. Fournier and S. Lanteri. Multiplicative and additive parallel multigrid algorithms for the acceleration of compressible flow computations on unstructured meshes. *Applied Numerical Mathematics*, 36:401–426, 2001.

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

[21] L. Greengard and W. Gropp. A parallel version of the fast multipole method. *Comput. Math. Appl.*, 20:63–71, 1990.

[22] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[23] W. Hackbush. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[24] W. Hackbush and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *NUMMATH*, 54(4):463–491, 1989.

[25] HSL. A collection of Fortran codes for large scale scientific computation, 2000. http://www.numerical.rl.ac.uk/hsl.

[26] S. A. Kharchenko and A. Yu. Yeremin. Eigenvalue translation based preconditioners for the GMRES(k) method. *Numerical Linear Algebra with Applications*, 2(1):51–77, 1995.

[27] L. Yu Kolotilina and A. Yu. Yeremin. Factorized sparse approximate inverse preconditionings. I: Theory. *SIAM J. Matrix Analysis and Applications*, 14:45–58, 1993.

[28] J. Lee, C.-C. Lu, and J. Zhang. Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics. Technical Report 363-02, Department of Computer Science, University of Kentucky, KY, 2002.

[29] 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.

[30] R. B. Morgan. Implicitely restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM J. Matrix Analysis and Applications*, 21(4):1112–1135.

[31] R. B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Analysis and Applications*, 16:1154–1171, 1995.

[32] R.B. Morgan. GMRES with deflated restarting. *SIAM J. Scientific Computing*, 24(1):20–37, 2002.

[33] 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.

[34] 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.

[35] Vladimir Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comp. Phys.*, 86(2):414–439, 1990.

[36] Y. Saad. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Automat. Contr.*, 33(3):290–297, 1988.

[37] 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.

[38] K. Sertel and J. L. Volakis. Incomplete LU preconditioner for FMM implementation.

[39] J. Song, C.-C. Lu, and W. C. Chew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects.

[40] G. Sylvand. *La Méthode Multipôle Rapide en Electromagnétisme : Performances, Parallélisation, Applications.* PhD thesis, Ecole Nationale des Ponts et Chaussées, 2002.

[41] R. S. Tuminaro. A highly parallel multigrid-like method for the solution of the euler equations. *SIAM J. Scientific and Statistical Computing*, 13:88–100, 1992.

[42] Stephen A. Vavasis. Preconditioning for boundary integral equations. *SIAM J. Matrix Analysis and Applications*, 13:905–925, 1992.

[43] Y.Saad. Analysis of augmented Krylov subspace techniques. *SIAM J. Scientific Computing*, 14:461–469, 1993.