Return

Efficient parallel iterative solvers for the solution of large dense linear systems arising from the boundary element method in electromagnetism

G. Alléon
EADS-CCR
Av. D. Daurat
31 700 Blagnac
guillaume.alleon@eads.net

B. Carpentieri
CERFACS
42 Av. G. Coriolis
31057 Toulouse
carpenti@cerfacs.fr

I.S. Duff
CERFACS
42 Av. G. Coriolis
31057 Toulouse
duff@cerfacs.fr

L. Giraud
42 Av. G. Coriolis
31057 Toulouse
giraud@cerfacs.fr

J. Langou
42 Av. G. Coriolis
31057 Toulouse
langou@cerfacs.fr

E. Martin
42 Av. G. Coriolis
31057 Toulouse
emartin@cerfacs.fr

G. Sylvand
2004 route des lucioles
BP 93 06 902 Sophia-Antipolis
guillaume.sylvand@inria.fr

Abstract

The boundary element method has become a popular tool for the solution of Maxwell's equations in electromagnetism. It discretizes only the surface of the radiating object and gives rise to linear systems that are smaller in size compared to those arising from finite element or finite difference discretizations. However, these systems are prohibitevely demanding in terms of memory for direct methods and challenging to solve by iterative methods. In this paper we address the iterative solution via preconditioned Krylov methods of electromagnetic scattering problems expressed in an integral formulation, with main focus on the design of the preconditioner. We consider an approximate inverse method based on the Frobenius-norm minimization with a pattern prescribed in advance. The preconditioner is constructed from a sparse approximation of the dense coefficient matrix, and the patterns both for the preconditioner and for the coefficient matrix are computed *a priori* using geometric information from the mesh. We describe the implementation of the approximate inverse in an out-of-core parallel code that uses multipole techniques for the matrix-vector products, and show results on the numerical scalability of our method on systems of size up to one million unknowns. We propose an embedded iterative scheme based on the GMRES method and combined with multipole techniques, aimed at improving the robustness of the approximate inverse for large problems. We prove by numerical experiments that the proposed scheme enables the solution of very large and difficult problems efficiently at reduced computational and memory cost. Finally we perform a preliminary study on a spectral two-level preconditioner to enhance the robustness of our method. This numerical technique exploits spectral information of the preconditioned systems to build a low rank-update of the preconditioner.

Return

1 Introduction

The analysis of wave propagation phenomena is gaining an increasing interest in recent years in the simulation of many challenging industrial processes, ranging from the prediction of the Radar Cross Section (RCS) of arbitrarily shaped 3D objects like aircrafts, the study of electromagnetic compatibility of electrical devices with their environment, the design of antennas and absorbing materials, and many others. All these simulations are very demanding in terms of computer resources, and require fast and efficient numerical methods to compute an approximate solution of Maxwell's equations. Standard discretization schemes like the finite-element or the finite-difference method can be used to discretize Maxwell's equations and give rise to very large and sparse linear systems to solve. However, the discretization of large 3D domains requires the use of effective absorbing boundary conditions to truncate the computational domain previous to the discretization phase, and may suffer from grid dispersion errors which occur when a wave has a different phase velocity on the grid compared to the exact solution. Alternatively, using the equivalence principle Maxwell's equations can be recast in the form of four integral equations that relate the electric and magnetic fields to the equivalent electric and magnetic currents on the surface of the object. For homogeneous or layered homogeneous dielectric bodies, integral equations are discretized on the surface of the object and at the discontinuous interfaces between two different materials. The resulting linear systems are much smaller than those generated by differential equation methods. However, a global coupling of the induced currents in the problem results in dense matrices, and the solution cost associated with these dense matrices has precluded until a recent past the popularity of integral solution methods in electromagnetism. Nowadays, owing to the impressive development in computer technology and to the introduction of fast methods which require less computational cost and memory resources, a *rigorous* numerical solution of many of these applications has become possible.

In this paper, we consider the solution of scattering problems expressed in an integral formulation. Amongst integral formulations, we concentrate on the electric-field integral equation (EFIE) that is the most general for electromagnetic scattering problems as it can handle fairly general geometries, and thus is widely used in industrial simulations. It expresses the electric field $\mathbf{E}$ outside the object in terms of the induced current $\mathbf{J}$ on the surface of the object. In the case of harmonic time dependency it reads

$$\mathbf{E}(x) = -\int_\Gamma \nabla G(x,x')\rho(x')d^3x' - \frac{ik}{c}\int_\Gamma G(x,x')\mathbf{J}(x')d^3x' + \mathbf{E}^E(x) \tag{1}$$

where $\mathbf{E}^E$ is the electric field due to external sources, and G is the Green's function for scattering problems:

$$G(x,x') = \frac{e^{-ik|x-x'|}}{|x-x'|}. \tag{2}$$

The EFIE provides a first-kind integral equation which is well known to be ill-conditioned. It can be converted into matrix equations by the Method of Moments (MoM) [20]. The surface of the object is modelled by a triangular faceted mesh, and the unknown current $\mathbf{J}$ on the surface is expanded into a set of basis functions $\mathbf{B}_i, i = 1, 2, ..., N$ which have local support [26]. The current expansion is introduced in (1), and the discretized equation is applied to a set of test functions to obtain a linear system

$$Ax = b \tag{3}$$

Return

whose unknowns are the coefficients of the expansion, representative of the vectorial flux across each edge in the triangular mesh. The entries of A are expressed in terms of surface integrals and when discretized using m-point Gauss quadrature formulae they assume the form

$$A_{kl} = \sum_{i=1}^{m} \sum_{j=1}^{m} \omega_i \omega_j G(x_{k_i}, y_{l_j}) \mathbf{B}_k(x_{k_i}) \cdot \mathbf{B}_l(y_{l_j}). \tag{4}$$

The coefficient matrix A is dense, complex, symmetric, non-hermitian for EFIE.

Direct solution methods are often the method of choice for solving systems (3) in an industrial environment because they are reliable and predictable both in terms of accuracy and cost. The factorization can be performed once and then is reused to compute a solution for all excitations. In industrial simulations objects are illuminated at several, slightly different incidence directions, and hundred of thousands of systems with the same coefficient matrix and a different right-hand side have often to be solved for the same application. For the solution of large-scale problems, direct methods are unfeasible even on large parallel platforms because they require storage of n^2 single or double precision complex entries of the coefficient matrix and $\mathcal{O}(n^3)$ floating-point operations to compute the factorization. Some direct solvers with reduced computational complexity have been introduced for the case when the solution is sought for blocks of right-hand sides (see e.g. [1, 11]) but the computational cost remains a bottleneck for large-scale applications.

The use of preconditioned Krylov solvers can be an alternative to direct solution methods, provided we have fast matrix-vector products and robust preconditioners. Active research efforts have been recently devoted to fast methods to perform fast matrix-vector products with $\mathcal{O}(n \log(n))$ computational complexity, including strategies for parallel distributed memory implementations. These methods, generally referred to as *hierarchical methods*, were introduced originally in the context of the study of particle simulations and can be effectively used on boundary element applications. However, it is now established that iterative solvers have to be used with some form of preconditioning to be effective on challenging problems, like those arising in industry. In this paper, we focus on the design of the preconditioner, and consider an approximate inverse preconditioner based on Frobenius-norm minimization with a pattern prescribed in advance. In Section 2 we describe the implementation of the preconditioner within an out-of-core parallel code that uses the Fast Multipole Method for the matrix-vector product, and we report on results on the numerical scalability of the preconditioner on a large industrial model problem. In Section 3 we propose an embedded iterative scheme based on the GMRES method aimed at improving the robustness of the preconditioner on large applications, and illustrate the efficiency and the cost of the proposed scheme on systems up to one million unknowns. In Section 4 we perform a preliminary study on a spectral two-level preconditioner to enhance the robustness of our preconditioner. This numerical technique exploits spectral information of the preconditioned systems to build a low rank-update of the preconditioner. Finally, in Section 5 we drop some conclusions from this work.

2 Preconditioning Boundary Integral Equations

The design of robust preconditioners for boundary integral equations can be challenging. Simple 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 depending on the integral

Return

formulation [31]. Block diagonal preconditioners are generally more robust than their pointwise 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 [30] and hybrid integral formulations in [24], but on the EFIE the triangular factors computed by the factorization are often very ill-conditioned due to the indefiniteness of A [6]. 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, 33, 29, 10, 23]). Owing to the rapid decay of the discrete Green's function, the location of the large entries in the inverse matrix exhibit some structure. In addition, only a very small number of its entries have large magnitude compared to the others that are much smaller. It means that a very sparse matrix is likely to retain the most relevant contributions to the exact inverse. This desirable property can be effectively exploited in the design of robust approximate inverses in electromagnetism.

2.1 Frobenius-norm minimization preconditioner

In this section we describe an approximate inverse preconditioner based on Frobenius-norm minimization. The original idea, due to Benson and Frederickson, is to compute the sparse approximate inverse as the matrix M which minimizes $\|I - MA\|_F$ (or $\|I - AM\|_F$ for right preconditioning) subject to certain sparsity constraints [3, 16]. The Frobenius norm is usually chosen since it allows the decoupling of the constrained minimization problem into n independent linear least-squares problems, one for each column of M (when preconditioning from the right) or row of M (when preconditioning from the left). The independence of these least-squares problems follows immediately from the identity:

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

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

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

holds, where $m_{\bullet j}$ is the column vector representing the jth column of M. Clearly, there is considerable scope for parallelism in this approach. The main issue is the selection of the nonzero pattern of M, that is the set of indices

$$S \ = \{ \ (i,j) \ \subseteq \ [1,n]^2 \mid m_{ij} = 0 \ \}. \tag{7}$$

The idea is to keep M reasonably sparse while trying to capture the "large" entries of the inverse, which are expected to contribute the most to the quality of the preconditioner. Two different approaches can be followed for this purpose, that is 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. Adaptive strategies can solve fairly general or hard problems but tend to be very expensive. The use of effective static pattern selection strategies can greatly reduce the amount of work in terms of CPU-time, and improve substantially the overall setup process, introducing significant scope for parallelism. On boundary integral equations the

Return

discrete Green's function (2) decays rapidly far from the diagonal, and the inverse of A may have a very similar structure to that of A. 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 this case a good pattern for the preconditioner can be computed in advance using graph information from $\tilde{A}$, a sparse approximation of the coefficient matrix constructed by dropping all the entries lower than a prescribed global threshold [21, 2, 5]. When fast methods are used for the matrix-vector products, all the entries of A are not available in memory and the pattern can be formed exploiting the near-field part of the matrix that is explicitely computed and available in the FMM [23].

Since we work in an integral equation context, relevant information for the construction of the pattern of M can be extracted from the mesh. When the object geometries are smooth, only the neighbouring edges in the mesh can have a strong interaction with each other, while far-away connections are generally much weaker. Thus an effective pattern for the jth column of the approximate inverse can be computed by selecting in the mesh edge j and its qth level nearest-neighbours. Three levels can generally provide a good pattern for constructing an effective sparse approximate inverse. Using more levels increases the computational cost but does not improve substantially the quality of the preconditioner [5]. When the object geometries are not smooth or have disconnected parts, far-away edges in the mesh can have a strong interaction with each other and be strongly coupled in the inverse matrix. In this case a more robust pattern for the preconditioner can be computed using physical information, that is selecting for each edge all those edges within a sufficiently large geometric neighbourhood. In [5] we compared pattern selection strategies based both on algebraic and mesh information on a large set of problems, and found that those exploiting geometric information are the most effective to capture the large entries of the inverse. We refer to the Frobenius-norm minimization method described in this section and computed using a static pattern strategy based on geometric information as the FROB preconditioner.

The construction of FROB costs $\mathcal{O}(n^2)$ arithmetic operations. This cost can be significantly reduced if the preconditioner is computed using as input a sparse approximation $\tilde{A}$ of the dense coefficient matrix A. On general problems, this approach can cause a severe deterioration of the quality of the preconditioner. In an integral equation context it is likely to be more effective because the boundary element method generally introduces a very localized strong coupling among the edges in the underlying mesh. It means that a very sparse matrix can still retain the most relevant contributions from the singular integrals that give rise to dense matrices. If the sparsity pattern of M is known in advance, the nonzero structure for the jth column of M is automatically determined, and defined as

$$J \; = \; \{i \; \in \; [1, n] \; s.t. \; (i, j) \; \in \; S\}.$$

The least-squares solution involves only the columns of $\tilde{A}$ indexed by J; we indicate this subset by $\tilde{A}(:, J)$. When $\tilde{A}$ is sparse, many rows in $\tilde{A}(:, J)$ are usually null, not affecting the solution of the least-squares problems (5). Thus if I is the set of indices corresponding to the nonzero rows in $\tilde{A}(:, J)$, and if we define by $\hat{A} = \tilde{A}(I, J)$, by $\hat{m}_j = m_j(J)$, and by $\hat{e}_j = e_j(J)$, the actual "reduced" least-squares problems to solve are

$$min\|\hat{e}_j - \hat{A}\hat{m}_j\|_2, j = 1, ..., n. \tag{8}$$

Usually problems (8) have much smaller size than problems (5) and can be effectively solved by dense QR factorization. In [2] the same nonzero sparsity pattern is selected both for A and M; in that case, especially when the pattern is very sparse, the computed preconditioner may be

Return

poor on some geometries. Selecting more entries in M than in $\tilde{A}$ can enable to compute a more robust preconditioner, and the additional cost in terms of CPU time is negligible because of the complexity of the QR factorization used to solve the least-squares problems [5]. Increasing the number of rows, that is the number of entries of $\tilde{A}$, is much cheaper in terms of overall CPU time than increasing the density of the preconditioner, that is the number of columns in the least-squares problems.

2.2 Implementation of the preconditioner in the FMM context

The Fast Multipole Method (FMM), introduced by Greengard and Rokhlin in [19], provides an algorithm for computing approximate matrix-vector products for electromagnetic scattering problems. The method is fast in the sense that the computation of one matrix-vector product costs $\mathcal{O}(n \log n)$ arithmetic operations instead of the usual $\mathcal{O}(n^2)$ operations, and is approximate in the sense that the relative error with respect to the exact computation is around 10^{-3} [12, 32]. The storage is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. Owing to these desirable properties its use in combination with iterative solvers is mandatory for the solution of large problems.

The basic idea of the algorithm is to compute interactions amongst degrees of freedom in the mesh at different levels of accuracy depending on their physical distance. Single and multilevel variants of the FMM exist and, for the multilevel algorithm, there are adaptive variants to handle efficiently inhomogeneous discretizations. In the hierarchical multilevel algorithm, the 3D obstacle is entirely enclosed in a large rectangular domain, and the domain is divided into eight boxes (four in 2D). Each box is recursively divided until the size of the smallest box is generally half of a wavelength. A tree-structured data is used at all levels. In particular, only non-empty cubes are indexed and recorded in the data structure. The resulting tree is called the *oct-tree* and its leaves are generally referred to as the *leaf-boxes*. The oct-tree provides a hierarchical representation of the computational domain partitioned by boxes: each box has one parent in the oct-tree, except for the largest cube which encloses the whole domain, and up to eight children. Obviously, the leaf-boxes have no children. The near-field interactions, that is those amongst degrees of freedom within neighbouring boxes, are computed from (4) using the regular expression of the discrete Green's function. The neighbourhood of a box is defined by the box itself and its 26 adjacent neighbours (eight in 2D). The far-field contribution from far away cubes is computed approximately. More precisely, for each far away box multipole coefficients are computed from (4) using truncated series expansion of the Green's function

$$G(x, y) = \sum_{p=1}^{P} \psi_p(x) \phi_p(y), \tag{9}$$

which separates the Green's function into two sets of terms, ψ_i and ϕ_j, that depend on the observation point x and the source point y, respectively. In (9) the integer P is generally very small, and the origin of the expansion is near the source point y while the observation point x is far away. Multipole coefficients are computed for all cubes starting from the lowest level of the oct-tree, that is from the leaf-boxes, and then recursively for each parent cubes by summing together multipole coefficients of their children. For each observation cube, an interaction list is defined which consists of those cubes that are not neighbours of the cube itself but whose parent is a neighbour of the cube's parent. Multipole coefficients of far-away boxes are sommed together to compute local coefficients for the observation cube, and the total effect of the far field from cubes that are in the interaction list are computed from the local coefficients. All the other interactions are computed hierarchically on a coarser level traversing the oct-tree. Both

Return

the computational cost and the memory requirement of the algorithm are of order $\mathcal{O}(n \log n)$. Further information on the algorithmic steps and recent theoretical investigations of the FMM can be found in [12, 32], and also in [18, 34] for discussions on parallel implementation issues.

The box-wise decomposition of the domain naturally leads to an *a priori* pattern selection strategy for M and $\tilde{A}$ in the FMM based on geometric information, that is on the spatial distribution of its degrees of freedom. We will adopt the following criterion: the nonzero structure of each column of the preconditioner is defined by retaining all the edges within a given leaf-box and those in one level of neighbouring boxes, and the structure for the sparse approximation of the dense coefficient matrix is defined by retaining the entries associated with edges included in the given leaf-box as well as those belonging to the two levels of neighbours. The approximate inverse has a sparse block structure; each block is dense and is associated with one leaf-box. Indeed the least-squares 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 we only have to compute one QR factorization per leaf-box. In our implementation we use a different partitioning to assemble the approximate inverse and the approximate multipole coefficient matrix. The size of the smallest boxes in the partitioning associated with the preconditioner is a user-defined parameter that can be tuned to control the number of nonzeros computed per row, that is the density of the preconditioner. According to our criterion, the larger the size of the leaf-boxes, the larger the geometric neighbourhood that determines the sparsity structure of the columns of the preconditioner. Parallelism can be exploited by assigning disjoint subsets of leaf-boxes to different processors and performing the least-squares solutions independently on each processor. We refer to [32] for a complete description of the parallel code that we used.

2.3 Numerical scalability of the preconditioner

In this section we are interested to study the numerical scalability of the Frobenius-norm minimization preconditioner. The optimal behaviour would be to get constant solution time when the problem size and the number of processors increase proportionally. It means that the amount of computation on each processor and the number of iterations should remain constant for increasing problem size. We show results on an Airbus aircraft (see Figure 1), a real life model problem in an industrial context.

In the numerical experiments, the surface of the object is always discretized using ten points per wavelength; larger discretizations are obtained by increasing the frequency of the illuminating wave. In Table 1, we report on the number of matrix-vector products required by the GMRES method to converge to an accuracy of 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 tolerance is accurate for engineering purposes, as it enables to detect correctly the radar cross section of the object. The symbol '–' in the tables means no convergence after 2000 iterations. We also report on the parallel elapsed time to build the preconditioner and to solve the linear system. All the runs have been performed in single precision on eight processors of a Compaq Alpha server. The Compaq Alpha server is a cluster of Symmetric Multi-Processors. Each node consists of four Alpha processors 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. The Airbus aircraft is very difficult to solve because the mesh has many surface details and the discretization matrices can become ill-conditioned. On small and medium problems, the number of GMRES iterations in Table 1 increases linearly with the problem size, the solution time nearly quadratically. On the

Return

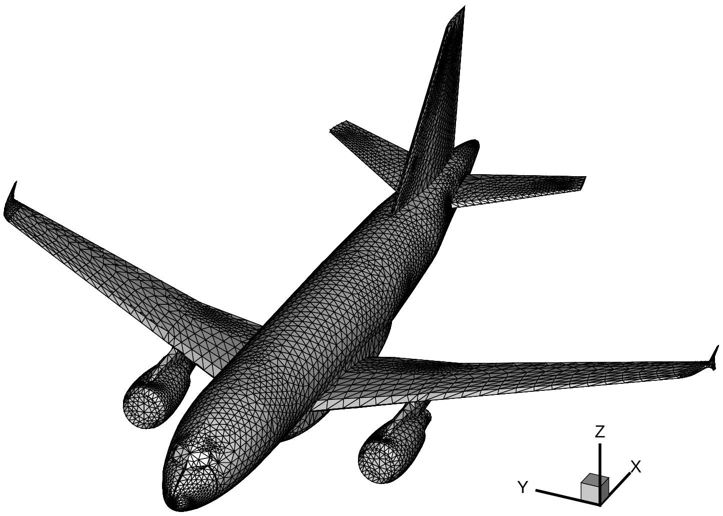


Figure 1: Mesh associated with an Airbus aircraft discretized with 15784 triangles

largest test case, discretized with one million unknowns, GMRES exceeds the memory limit on 64 processors. In this case the use of large restarts does not enable to get convergence within 2000 iterations except on a small mesh of size 94704.

Size	Density FROB	Time FROB	GMRES(∞)		GMRES(120)	
			Iter	Time	Iter	Time
94704	0.28	11m	746	2h 9m	1956	3h 13m
213084	0.13	31m	973	7h 19m	+2000	7h 56m
591900	0.09	1h 30m	1461	16h 42m*	+2000	1d 57m
1160124	0.02	3h 24m	M.L.E.*	N.A.	+2000	$>$ 4d*

Table 1: Number of matrix-vector products and elapsed time in seconds required to converge on an aircraft Airbus on 8 procs on the Compaq machine, except those marked with $*$, that have been run on 64 procs. Illuminating direction $(\phi, \theta) = (30^o, 30^o)$. Tolerance for the iterative solution $= 10^{-3}$. Acronyms: N.A. $\equiv$ not available, M.L.E. $\equiv$ memory limits exceeded.

3 Improving the preconditioner robustness using embedded iterations

The numerical results shown in the previous section indicate that the FROB preconditioner tends to become less effective when the problem size increases, especially on difficult systems. By its nature the sparse approximate inverse is inherently local because each degree of freedom is coupled to only a very few neighbours. When the exact inverse is dense the compact support

Return

```
**Outer solver** ⟶ FGMRES, FQMR

Do k=1,2, ...
    • M-V product: FMM with high accuracy
    • Preconditioning : **Inner solver** (GMRES, TFQMR, ...)
      Do i=1,2, ...
          • M-V product: FMM with low accuracy
          • Preconditioning : $M_{Frob}$
      End Do
End Do
```

Figure 2: Inner-outer solution schemes in the FMM context. Sketch of the algorithm.

used to define the preconditioner may not allow an exchange of global information and on large problems the lack of global approximation may have a severe impact on the convergence. In our implementation the overall number of computed nonzeros decreases for increasing values of the frequency. When the preconditioner become very sparse, information related to the far-field are completely lost. In this case some suitable mechanism has to be introduced to recover global information on the numerical behaviour of the discrete Green's function.

In this section we describe an embedded iterative scheme, combined with multipole techniques, that is designed to meet the goals of robustness, scalability and parallelism of the iterative solver. The basic idea is to carry out a few steps of an inner Krylov method for the preconditioning operation. The overall algorithm results in an inner-outer scheme (see Figure 2), and its efficiency relies on two main factors, that is: the inner solver has to be preconditioned so that the residual in the inner iterations can be significantly reduced in a few number of steps, and the matrix-vector products within the inner and the outer solvers are carried out at different accuracy. The motivation that naturally leads us to consider inner-outer schemes is to try to balance the locality of the preconditioner with the use of the multipole matrix. Experiments conducted in [17] with inner-outer schemes combined with multipole techniques on the potential equation were unsuccesful. In that case no preconditioner was used in the inner solver. The desirable feature of using different accuracy for the matrix-vector products is enabled by the use of the FMM. In our scheme, highly accurate FMM is used within the outer solver that is used to actually solve the linear system, and a lower accurate FMM within the inner solver that is used as preconditioner for the outer scheme. More precisely, the FMM accuracy is "high" for the FGMRES iteration (the relative error in the matrix-vector computation is around $5 \cdot 10^{-4}$ compared to the exact computation) and "medium" (the relative error is around 10^{-3}). In fact, we solve a nearby system for the preconditioning operation. This enables us to save considerable computational effort during the iterative process.

3.1 Numerical results

In this section we carry out experiments using the FGMRES method [27] as the outer solver with an inner GMRES iteration [28] preconditioned with the Frobenius-norm minimization method

Return

described in Section 2. For the GMRES and FGMRES methods, we consider the implementations described in [15] and [14], respectively. The analysis of the convergence history of GMRES gives us some clues to the numerical behaviour of the proposed scheme. The residual of GMRES tends to decrease very rapidly in the first few iterations independently of the restarts, then decreases much more slowly, and finally tends to stagnate to a value that depends on the restart; the larger the restart, the lower the stagnation value. It suggests that a few steps in the inner solver can be very effective for obtaining a significant reduction of the initial residual. A different numerical behaviour has been observed for other Krylov methods as inner solver, as the TFQMR solver [4]. The residual in the beginning of the convergence is nearly constant or decreases very slowly. The use of this method as an inner solver is ineffective. Large restarts of GMRES do not enable a further reduction of the normwise backward error in the beginning of convergence. Thus small restarts should be preferred in the inner GMRES iterations. Amongst the various possibilities, we select FGMRES(30) and GMRES(60) on the Airbus aircraft, that seem to give the optimal trade-off.

We show the results of experiments on the Airbus aircraft in Table 2. We report on the number of inner and outer matrix-vector products and the elapsed time needed to achieve convergence using a tolerance of 10^{-3} on eight pocessors on the Compaq machine. For comparison, in the tables we also show results obtained with the restarted GMRES method. The comparison in the tables is fair because GMRES has exactly the same storage requirements as the combination FGMRES/GMRES. In fact, for the same restart value, the storage requirement for the FGMRES algorithm is twice that for the standard GMRES algorithm, as it stores the preconditioned vectors of the Krylov basis.

The combination FGMRES/GMRES remarkably enhances the robustness of the preconditioner especially on large problems. It can be seen that GMRES(120) does not converge after 2000 iterations even on quite small problem.

Size	Density FROB	Time FROB	GMRES(120)		FGMRES(30,60) *precfmm(high,medium)*	
			Iter	Time	Iter	Time
94704	0.28	11m	1956	3h 13m	23+1320	2h 30m
213084	0.13	31m	+2000	N.A.	30+1740	6h 10m
591900	0.09	1h 30m	+2000	N.A.	57+3300	1d 9h 45m
1160124	0.02	3h 24m	+2000	N.A.	51+2940	16h 41m*

Table 2: Number of matrix-vector products and elapsed time in seconds required to converge on an aircraft Airbus. The tests have been run on 8 procs on the Compaq machine, except those marked with $*$, that have been run on 64 procs. Illuminating direction $(\phi, \theta) = (30^o, 30^o)$. Tolerance for the iterative solution $= 10^{-3}$. Acronyms: N.A. $\equiv$ not available.

4 Spectral low-rank updates

It is well known that the convergence of Krylov methods for solving the linear system often depends to a large extent on the eigenvalue distribution. In many cases, it is observed that “removing” the smallest eigenvalues can greatly improve the convergence. Many of the preconditioners proposed in the literature succeed in clustering most of the eigenvalues of the preconditioned matrix MA (for left preconditioning) far from the origin. Such a distribution is

Return

highly desirable to get fast convergence of Krylov solvers. However, a few eigenvalues can be left close to zero and they potentially can significantly degrade the convergence. In order to tackle this difficulty we propose a refinement technique based on the introduction of low-rank corrections computed from spectral information associated with the smallest eigenvalues of MA. Roughly speaking, the proposed technique consists in solving exactly the preconditioned system in the low dimensional space spanned by the eigenvectors associated with the eigenvalues closest to the origin. This is then used to update the preconditioned residual. We consider the solution of the linear system

$$Ax = b, \tag{10}$$

where A is a $n \times n$ unsymmetric complex nonsingular matrix, and x and b are vectors of size n. The linear system is solved using a preconditioned Krylov solver and we denote by M_1 the left preconditioner, meaning that we solve

$$M_1 A x = M_1 b. \tag{11}$$

We assume that the preconditioned matrix $M_1 A$ is diagonalizable, that is:

$$M_1 A = V \Lambda V^{-1}, \tag{12}$$

with $\Lambda = diag(\lambda_i)$, where $|\lambda_1| \leq \ldots \leq |\lambda_n|$ are the eigenvalues and $V = (v_i)$ the associated right eigenvectors.

Proposition 1 *Let W be such that $\tilde{A}_c = W^H A M_1 V_\varepsilon$ has full rank, $\tilde{M}_c = M_1 V_\varepsilon \tilde{A}_c^{-1} W^H$ and $\tilde{M} = M_1 + \tilde{M}_c$. Then $A\tilde{M}$ is similar to a matrix whose eigenvalues are*

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

For more details on this technique and its application to left preconditioning or symmetric definite situation, we refer to [8].

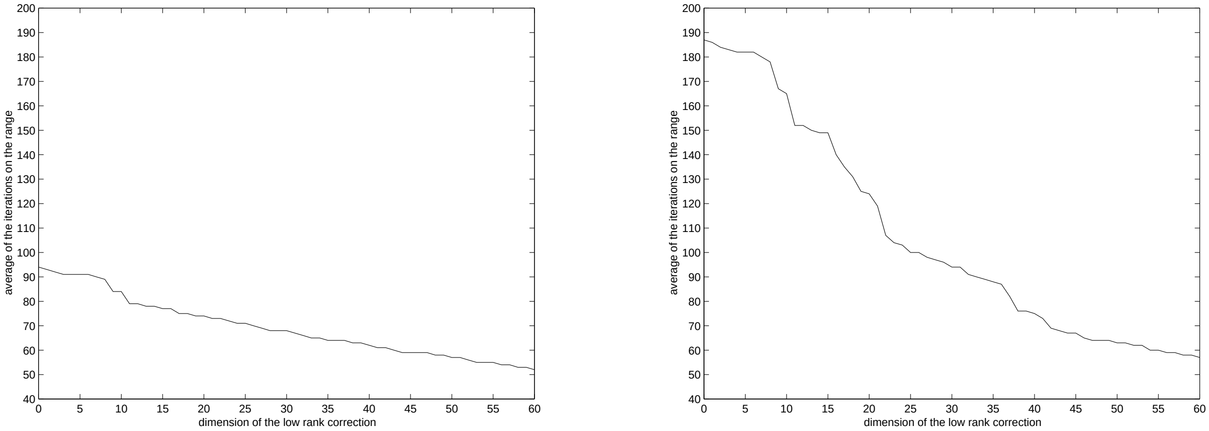
In the sequel, we illustrate the benefit of using this approach when computing a complete monostatic radar cross section. In this framework, many linear systems having the same coefficient matrix but different right-hand sides have to be solved. For the Airbus calculation 181 linear systems have to be solved. In Figure 3 we depict the convergence history of full GMRES when the number of shifted eigenvalues is varied for two specific right-hand sides. On the left, we consider a right-hand side that is easily solved by GMRES; on the right a right-hand side that is more difficult to solve. In these graphs, FROB denotes the Frobenius norm minimization preconditioner and $SpU(k)$ the FROB preconditioner with a rank-k spectral update. In both cases, it can be observed that the more eigenvalues are shifted, the better the convergence is.

The gain in iteration number is even larger if a restart is used for GMRES. This behaviour is illustrated in Figure 4.

Using these technique, for a complete monostatic radar cross section calculation enables us to save a large number of iterations. In Figure 5, we depict the number of full GMRES iterations for each of the right-hand side associated with the various angles of the illuminating wave. It can be observed that the saving is significative.

In the experiments reported in this paper the eigenvalues calculation were performed in a preprocessing phase using Arpack [25]. The extra cost of this preprocessing is negligible. The complete solution time on 32 processor Compaq Alpha server (EV 6.8, 1 Ghz) took 46 hours using the FROB preconditioner and reduced to 20 hours when using the spectral low

Return



(a) “easy” right-hand side

(b) “difficult” right-hand side

Figure 3: Convergence history of full GMRES varying the number of shifted eigenvalues

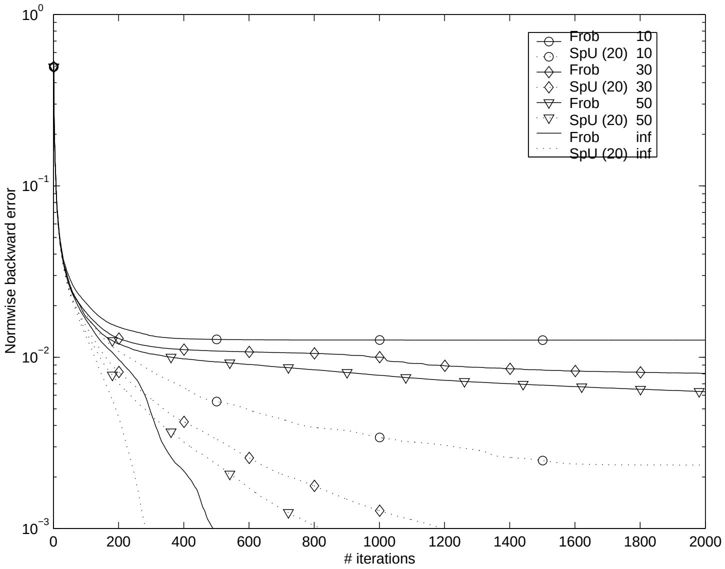


Figure 4: Convergence history varying the restart on a “difficult” right-hand side

rank correction (including the eigencalculation). Similar gain have been observed on other test problems [13].

We shall mention that those preconditioning techniques are still effective when used for accelerating the convergence of Krylov solvers designed for handling multiple rigth-hand sides. Such Krylov solvers are for instance Block-GMRES or Seed-GMRES, we refer to [22] for details and illustration of this claim.

5 Concluding remarks

In this paper, we present the parallel scalability and the numerical scalability of the solver for solving large problems with up-to a few million unknowns. We discuss two techniques to intro-

Return

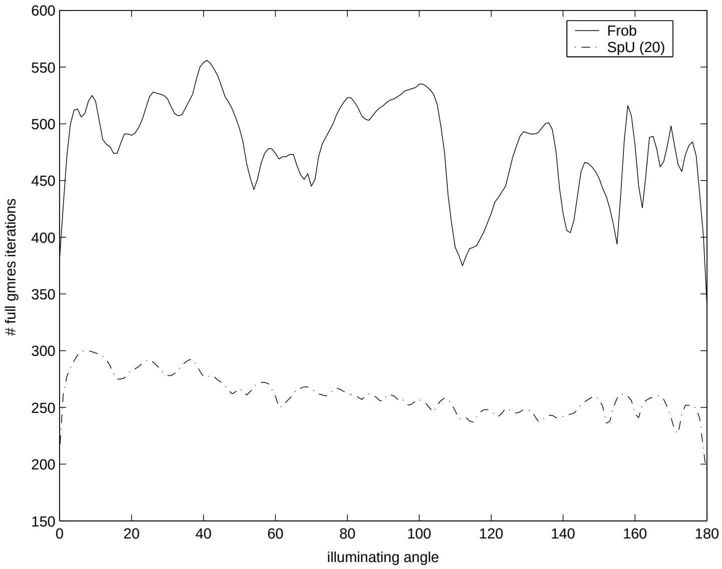


Figure 5: # iterations for the 181 right-hand sides

duce multi-level mechanisms for improving its robustness and its scalability. The first approach is based on *inner-outer* iterations with variable accuracy in the fast multipole calculation [7]. The inner Krylov iterations, preconditioned with the approximate inverse technique, implement a less accurate but faster multipole approximation and enable some global information about the overall solution to be recovered in few iterations. This approximate solution is then used to precondition the outer FGMRES iterations. The robustness of the resulting solver is demonstrated on large problems arising both from academic and from industrial applications such as those involved in aircraft design. In particular, on objects discretized using more than a million unknowns, this new scheme reduces the elapsed time to solve a problem on a sphere by almost one order of magnitude, and enables us to compute the solution on an aircraft while classical approaches fail. The second approach is based on low-rank update of the preconditioner [8] similar in structure to the one studied in domain decomposition [9]. Here the idea is to exploit information related to the smallest eigenvalues to make a low-rank update of the preconditioner. This update enables us to shift the smallest eigenvalues of the original preconditioned system close to one and results in faster convergence of the Krylov solvers. The resulting preconditioner has been successfully applied to significantly speed-up various Krylov solvers. For solving symmetric non-Hermitian systems, such as those involved in certain BEM formulations in electromagnetism, we also derive a symmetric version that is particularly efficient on our problems. We show that the extra computation for this spectral information can become quickly unimportant if many linear systems with the same coefficient matrix but different right-hand sides have to be solved [4, 13]. We have shown numerical experiments on parallel distributed memory computers to highlight the efficiency of the implementation as well as the efficiency of the resulting numerical schemes for solving large industrial test problems.

Finally we mention that more details can be found in some recent PhD manuscript [4, 22, 32].

Return

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

[4] B. Carpentieri. *Sparse preconditioners for dense linear systems from electromagnetic applications.* PhD thesis, CERFACS, Toulouse, France, 2002.

[5] B. Carpentieri, I. S. Duff, and L. Giraud. Some sparse pattern selection strategies for robust frobenius norm minimization preconditioners in electromagnetism. Technical Report RAL-TR-2000-009, Rutherford Appleton Laboratory, Chilton, England, 2000. To Appear in Numerical Linear Algebra with Applications.

[6] B. Carpentieri, I. S. Duff, L. Giraud, and M. Magolu monga Made. Sparse symmetric preconditioners for dense linear systems in electromagnetism. Technical Report TR/PA/01/35, CERFACS, Toulouse, France, 2001. Also Technical Report RAL-TR-2002-016. To appear in NLAA.

[7] B. Carpentieri, I. S. Duff, L. Giraud, and G. Sylvand. Combining fast multipole techniques and an approximate inverse preconditioner for large parallel electromagnetics calculations. Technical Report in preparation, CERFACS, Toulouse, France.

[8] B. Carpentieri, I.S. Duff, and L. Giraud. A class of spectral two-level preconditioners. *SIAM J. Scientific Computing*, To appear.

[9] L. M. Carvalho, L. Giraud, , and P. Le Tallec. Algebraic two-level preconditioners for the schur complement method. *SIAM J. Scientific Computing*, 22(6):1987–2005, 2001.

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

[11] 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.

[12] E. Darve. The fast multipole method (i) : Error analysis and asymptotic complexity. *SIAM J. Numerical Analysis*, 38(1):98–128, 2000.

[13] I. S. Duff, L. Giraud, J. Langou, and E. Martin. Using spectral information for accelerating the convergence of GMRES for large electromagnetics calculations. Technical Report in preparation, CERFACS, Toulouse, France.

[14] V. Frayssé, L. Giraud, and S. Gratton. A set of Flexible-GMRES routines for real and complex arithmetics. Technical Report TR/PA/98/20, CERFACS, Toulouse, France, 1998.

[15] V. Frayssé, L. Giraud, S. Gratton, and J. Langou. A set of GMRES routines for real and complex arithmetics on high performance computers. Technical Report TR/PA/03/3, CERFACS, Toulouse, France, 2003.

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

[17] 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.

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

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

[20] R. Harrington. Origin and development of the method of moments for field computation. *IEEE Antennas and Propagation Magazine*, 1990.

[21] 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.

[22] J. Langou. *Solving large linear systems with multiple right-hand sides.* PhD thesis, Instutut National des Sciences Appliquées de Toulouse, 2003.

[23] 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.

[24] 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.

[25] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM, Philadelphia, 1998.

[26] 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.

[27] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Scientific and Statistical Computing*, 14:461–469, 1993.

[28] 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.

[29] 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.

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

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

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

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

[34] F. Zhao and S. L. Johnsson. The parallel multipole method on the connection machine. *SIAM J. Scientific and Statistical Computing*, 12:1420–1437, 1991.