Fast Direct Solution of Method of Moments Linear System

Alex Heldring, Juan. M. Rius, *Member, IEEE*, José Maria Tamayo, Josep Parrón, and Eduard Úbeda

***Abstract*—A novel algorithm, the compressed block decomposition (CBD), is presented for highly accelerated direct (non-iterative) method of moments (MoM) solution of electromagnetic scattering and radiation problems. The algorithm is based on a block-wise subdivision of the MoM impedance matrix. Impedance matrix subblocks corresponding to distant subregions of the problem geometry are not calculated directly, but approximated in a compressed form. Subsequently, the matrix is decomposed preserving the compression. Examples are presented of typical problems in the range of 5000 to 70 000 unknowns. The total execution time for the largest problem is about 1 h and 20 min for a single excitation vector. The main strength of the method is for problems with multiple excitation vectors (monostatic RCS computations) due to the negligible extra cost for each new excitation. For radiation and scattering problems in free space, the numerical complexity of the algorithm is shown to be N^2 and the storage requirements scale with $N^{3/2}$.**

***Index Terms*—Fast solvers, impedance matrix compression, method of moments (MoM), numerical simulation.**

I. Introduction

THE METHOD OF moments (MoM) [1] is a widely used technique for numerical simulation of radiation and scattering problems. Its main drawback, the costly construction, storage, and solution of a dense linear system, has led to the development of several fast algorithms such as AIM [2], MLFMA [3], and Multilevel MDA [4]–[6]. Most of these algorithms are based on some approximative compressed representation of the linear system matrix, or impedance matrix, that needs much less storage and highly accelerates the matrix-vector multiplications.

These efficient matrix-vector multiplication algorithms are then used inside an iterative solution method such as the bi-conjugate gradient (BiCG) or the generalized minimum residual (GMRES), often with an appropriate preconditioner [7]. These new algorithms have enormously extended the range of problems that MoM can manage. The maximum affordable number of unknowns used to be limited to a few thousands. Now problems with hundreds of thousands or even millions of unknowns are within reach, depending on the available computational resources.

Based on the previous, one might be tempted to conclude that direct solution of the MoM matrix equation, through Gaussian elimination or LU decomposition, has become obsolete. This is not entirely the case, for the following number of reasons.

- First of all, the fast algorithms are very efficient for electrically large structures, while the direct solution is faster for small and medium size problems, depending on the computational resources and on the specifics of the problem. The turning point may be of the order of a few thousand unknowns.
- Furthermore, iterative solution methods for matrix equations yield the solution to the linear system for only one independent vector (excitation vector) at a time. Consequently, the computational effort is proportional to the number of independent vectors. By contrast, the bulk of the effort in LU decomposition, which is the generation of the L and U factors of the impedance matrix, needs to be done only once for as many independent vectors as needed.
- Finally, the convergence rate of iterative methods can vary in an unpredictable way. It is related to the matrix condition number, which is notoriously bad for the electric field integral equation (EFIE) in large problems [8]. The only remedy is the use of a good preconditioner with a relatively large number of non-zero elements, but the construction of such a preconditioner becomes the bottleneck of the computation [9]–[11].

In this paper, we propose a direct solution method for the MoM linear system, the compressed block decomposition (CBD). It is directly applicable to almost any MoM formulation and substantially reduces the storage and computational cost with respect to straightforward LU decomposition. The method is based on a blockwise compression of the impedance matrix (see Section II), by the same technique as used in the matrix decomposition algorithm (MDA) [4]–[6]. The compressed matrix is then decomposed using an adapted version of the block LU algorithm previously published in [9] and [10], which allows to retain the original compression rate (see Section III-A). As in the MDA algorithm, the compression rate depends on the problem characteristics. In Section III-C, the algorithm is shown to scale asymptotically with the number of unknowns as N^2. The storage requirement scales with $N^{3/2}$. Several examples of typical 3-D scattering and radiation problems, concerning perfectly conducting objects in free space and modeled with the electric field integral equation (EFIE), are given in Section IV.

Manuscript received September 18, 2006; revised July 9, 2007. This work was supported by the Spanish "Comisión Interministerial de Ciencia y Tecnología (CICYT)" under the "Ramón y Cajal" Programme and projects TIC 2003-09317-C03-02 and TIC 2003-09317-C03-03.

A. Heldring, J. M. Rius, J. M. Tamayo, and E. Úbeda are with the Antenna Laboratory, Department of Signal Processing and Telecommunications, Universitat Politecnica de Catalunya, 08034 Barcelona, Spain (e-mail: heldring@tsc.upc.edu).

J. Parrón is with the Department of Telecommunications and Systems Engineering, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain.

Digital Object Identifier 10.1109/TAP.2007.908804

II. Impedance Matrix Compression

A. MDA Compression

The MDA compression algorithm is based on the observation that subblocks of the impedance matrix representing interactions between mutually distant groups of elementary scatterers (basis functions), are typically low rank. In MDA, this feature is exploited as follows: The problem geometry is divided into subdomains containing groups of subscatterers. Each pair of source-observer subdomains corresponds to an impedance matrix block. Blocks representing self interactions, or interactions between adjacent groups cannot be compressed through MDA, but all other blocks can.

The details of the MDA can be found in [4] and [5]. Here, we show the form of the resulting decomposition. Consider a block that represents the action of a group labeled ν, of n elementary scatterers upon another and distant group labeled μ of m elementary scatterers. The corresponding submatrix $Z_{[m][n]}$ is then replaced with[1]

$$Z_{[m][n]} \longrightarrow Z_{[m][p]} Z^{-1}_{[q][p]} Z_{[q][n]} \tag{1}$$

where the matrix dimensions are, respectively, $m \times n$, $m \times p$, $p \times q$, and $q \times n$. The sets $[p]$ and $[q]$ denote sets of equivalent elementary scatterers defined in groups ν and μ, respectively, using the techniques described in [4] and [5]. Since $p \ll m$ and $q \ll n$, the storage size and matrix-vector computation time are much smaller for the decomposed matrix than for the original one, $mp+pq+qn \ll mn$. Of course, the compression implies a loss of information, which can be controlled through the choice of p and q, compromising between compression rate and desired accuracy.

B. SVD Post Compression

An important further compression, with little additional loss of accuracy, can be obtained by applying the following sequence of transformations to the resulting factors:

$$Z_{[m][p]} \longrightarrow Q_{[m][p]} R_{[p][p]} \tag{2}$$

$$Z_{[q][n]} \longrightarrow R'_{[q][q]} Q'_{[q][n]} \tag{3}$$

$$R_{[p][p]} Z_{[p][q]} R'_{[q][q]} \longrightarrow U_{[p][t]} S_{[t][t]} V'_{[t][q]}. \tag{4}$$

Equation (2) represents a QR decomposition, (3) an adjoined QR decomposition (the prime denotes the adjoined matrix), and (4) an SVD decomposition with a given compression threshold τ on the singular values. Denoted as SVD(τ), this amounts to eliminating all rows and/or columns in the respective matrices that correspond to a given singular value that is smaller than $\tau\sigma_1$, where σ_1 is the largest singular value. This procedure reduces the inner dimensions of the inner matrix from $p \times q$ to $t \times t$.

The previous transformations inserted in (1) yield

$$Z_{[m][n]} \longrightarrow \widetilde{Z}_{[m][t]} S_{[t][t]} \widetilde{Z}_{[t][n]} \tag{5}$$

with

$$\widetilde{Z}_{[m][t]} = Q_{[m][p]} U_{[p][t]}$$

$$\widetilde{Z}_{[t][n]} = V'_{[t][q]} Q'_{[q][q]}. \tag{6}$$

Since the outer matrices $\widetilde{Z}_{[m][t]}$ and $\widetilde{Z}_{[t][n]}$ are orthonormal and $S_{[t][t]}$ is a diagonal matrix, (5) corresponds to a singular value decomposition (SVD) of $Z_{[m][n]}$ [12], leading to the optimum compression with threshold τ.

While SVD post-compression is optional in iterative MDA, it is necessary for the CBD algorithm, as will be explained in Section III. The QR and SVD algorithms are computationally expensive, but due to the initial MDA compression they are only applied to blocks of highly reduced size with respect to the original impedance matrix blocks ($p \ll m$ and $q \ll n$).

Blocks representing interactions between adjacent source and field boxes cannot be precompressed with MDA, since MDA requires a minimum distance between the source and the observer subdomains. However, using an octal tree space subdivision such blocks can be subdivided into their children blocks, and MDA can be used to compress all the children blocks that represent nonadjacent interactions. Those smaller blocks that do represent touching boxes can in turn be subdivided and treated in the same way, recursively down to a minimum box-size, which is typically of the order of half a wavelength. Of course, all these compressed submatrices must be recombined to form the compressed representation of the original block. Although this involves concatenation of compressed matrices, as depicted in Fig. 1(b) and (c), and reorthonormalization and recompression of the resulting matrices, it is still much more efficient than direct SVD(τ) compression.

At the lowest level, blocks representing touching boxes cannot be compressed with MDA so they must be directly compressed with SVD(τ). The latter SVD(τ) compression would be computationally expensive if applied directly to the submatrices. Instead, an adaptive division strategy is used: a strip of columns with a width of, for instance, $m' = 5\%$ of the total number of columns is picked off the left of the block $Z_{[m][n]}$. This strip is QR factorized and the R-factor, which is of dimensions $m' \times m'$, is compressed with SVD(τ). If the compression is small (compression rate larger than 25%), the width of the next strip to be picked off is doubled. When the whole block is thus precompressed, all the terms are concatenated and a final SVD(τ) is applied to the result. This strategy considerably accelerates the compression of the blocks corresponding to adjacent boxes.

III. Compressed Block Decomposition Algorithm

A. Block Decomposition

Once the blockwise compressed impedance matrix is generated, it can be decomposed with an adapted version of the block LU algorithm we previously developed to factorize a sparse preconditioner for iterative methods [9], [10]. The block subdivision of the impedance matrix is depicted in Fig. 2. On the left-hand side, the matrix is represented in block format, while the right-hand side represents the decomposition. Note that all

[1] Notation: lower case latin letters in italics like n denote an integer number, the same symbol in square brackets denotes a set of indices. For example $[n]$ denotes the set of indices that identify n elementary scatterers.

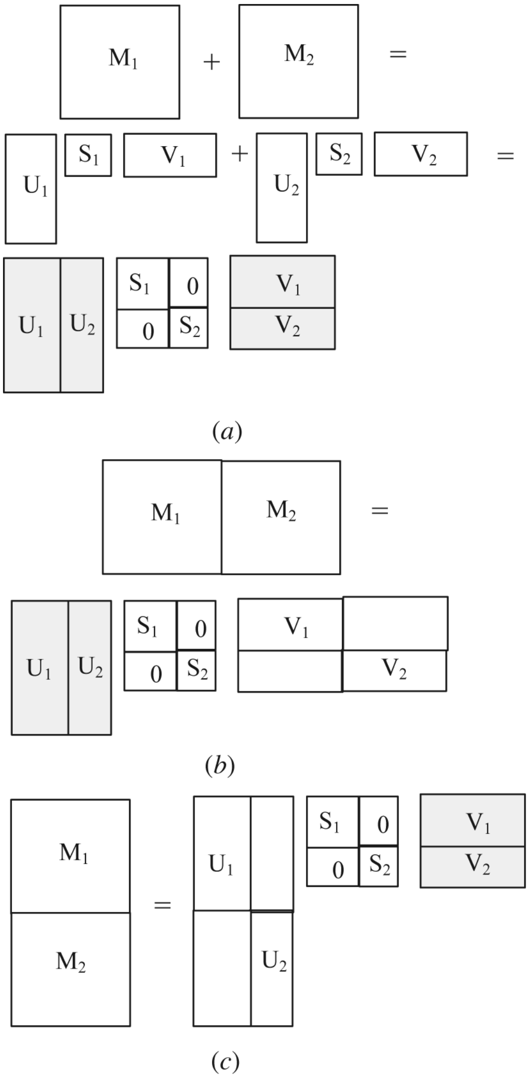


Fig. 1. (a) Summation, (b) horizontal concatenation, and (c) vertical concatenation of compressed matrices. Matrices in grey need to be reorthonormalized.

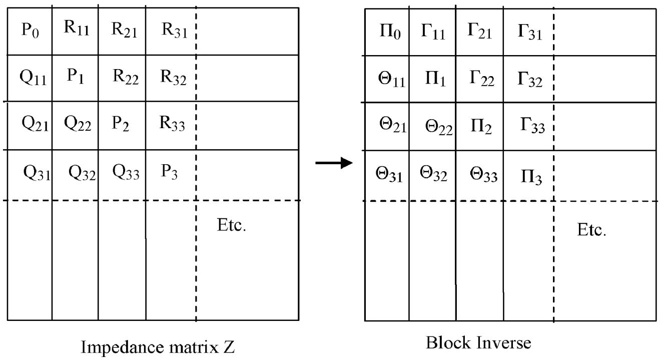


Fig. 2. Transformation from blocked impedance matrix to block inverse.

the Q and R blocks of the impedance matrix are stored in compressed form as the product of three matrices (5).

It is important to remark here that in Fig. 1 each matrix block is associated to a pair of source and observation boxes in the space subdomain decomposition. P, Q, and R are submatrices of Z with noncontiguous rows and column indices corresponding to the basis and testing functions within the observation and the source box, respectively.

The CBD algorithm is as follows.

Algorithm CBD (M blocks):

```
1) $\Pi_0 = P_0^{-1}$
2) For $i = 1$ to $M-1$ do
3)     For $j = 1$ to $i$ do
4)         $\mathrm{A}_{jj} = \Pi_{j-1}(\mathrm{R}_{ij} - \sum_{k=1}^{j-1} \Theta_{j-1,k}\mathrm{R}_{ik})$
5)         $\mathrm{B}_{jj} = (\mathrm{Q}_{ij} - \sum_{k=1}^{j-1} \mathrm{Q}_{ik}\Gamma_{j-1,k})\Pi_{j-1}$
6)         reorthonormalize $\mathrm{A}_{jj}$ and $\mathrm{B}_{jj}$
7)         For $k = 1$ to $j-1$ do
8)             $\mathrm{A}_{jk} = A_{j-1,k} - \Gamma_{j-1,k}\mathrm{A}_{jj}$
9)             $\mathrm{B}_{jk} = B_{j-1,k} - \mathrm{B}_{jj}\Theta_{j-1,k}$
10)            reorthonormalize $\mathrm{A}_{jk}$ and $\mathrm{B}_{jk}$
11)        End $(k)$
12)    End $(j)$
13)    For k = 1 to i do; $\Gamma_{ik} = \mathrm{A}_{ik}$; $\Theta_{ik} = \mathrm{B}_{ik}$; End $(k)$
14)    $\Pi_i = (\mathrm{P}_i - \sum_{j=1}^{i} \mathrm{Q}_{ij}\Gamma_{ij})^{-1}$
15) End $(i)$
```

End algorithm

Steps 6) and 10) of the algorithm are explained in Section III-B. Finally, in order to apply the decomposed matrix to an independent vector $\mathbf{x}$ to obtain the linear system solution $\mathbf{y}$, the independent vector is subdivided according to the blocks of the impedance matrix, yielding a set of vectors $\mathbf{x}_0, \ldots, \mathbf{x}_{M-1}$. Then the algorithm Apply_CBD given in the following is used to compute the solution block by block.

Algorithm Apply_CBD(x):

```
1) For $i = 0$ to $M-1$ do
2)     $\mathbf{y}_i = \Pi_i(\mathbf{x}_i - \sum_{j=1}^{i} \Theta_{ij}\mathbf{x}_{j-1})$
       For $j = 1$ to $i$ do
3)         $\mathbf{y}_{j-1} = \mathbf{y}_{j-1} - \Gamma_{ij}\mathbf{y}_i$
4)     End $(j)$
5) End $(i)$
```

End algorithm

The work for the algorithm Apply_CBD is, for a single independent vector, almost negligible compared to the decomposition step (see the numerical examples in Section IV).

It is also possible to use the conventional block-LU algorithm given by Golub and van Loan in [13] instead of the previously shown CBD algorithm. Indeed, the Golub and van Loan algorithm involves only half the number of matrix summations and, as we shall see in Section III-B, the summations of compressed matrix blocks are bottleneck of the algorithm. The Golub and van Loan algorithm is not twice as fast, however, because the multiple summations in lines 4 and 5 of the CBD algorithm can be done more efficiently than the single summations in lines 8 and 9 while the Golub and van Loan algorithm only involves single summations. To do the multiple summations of lines 4 and 5 efficiently, it suffices to sort the terms of the summation with respect to the size of the inner matrix, and to start summing the two smallest terms, then adding the third smallest term, and so on.

Furthermore, there are the following two very common situations in which the CBD algorithm has additional advantages that make it preferable to conventional block-LU.

1) When the matrix is symmetrical, anti-symmetrical, Hermitian or, in general, when it is possible to know the elements in a given block from the elements in the transposed block, this property is conserved in the decomposition. For instance, the EFIE-MoM impedance matrix is symmetrical (excluding anisotropic media), in which case $\Gamma_{ik} = \Theta_{ik}^T$, for all i and k. This means that the matrices B_{ik} do not need to be computed. Also, much less storage space is needed. The numerical examples in Section IV all use this property.
2) As mentioned earlier, the CBD algorithm is based on an algorithm originally conceived for out-of-core factorization of large preconditioning matrices [9]. At every decomposition step (index i in the algorithm), the only blocks that are modified are the current blocks (of index i). In contrast, the conventional block-LU requires the modification, at every level i, of all the not-yet-factorized blocks. If the decomposition needs to be done using hard disk storage, for lack of available memory, and the total number of elements of the compressed matrix equals m, the CBD algorithm needs $O(m^2)$ reading instructions but only $O(m)$ writing instructions, while conventional block-LU needs $O(m^2)$ for both reading and writing.

B. Compressed Decomposition

The CBD algorithm from Section III-A represents the various blocks as single matrices. In reality, all nondiagonal blocks are available in compressed form, that is, a sequence of three matrices with the inner matrix being a small diagonal matrix (5) and the outer matrices orthonormal. In the proposed block decomposition algorithm, two types of operations are used: multiplication and summation.

The multiplication of two sequences of matrices poses no problem, since the product of the inner matrices leads to a new sequence of three matrices and the two outer matrices are still orthonormal. The inner matrix is no longer diagonal, not even square, in general, but this is not essential to the algorithm.

The sum is obtained as shown in Fig. 1(a). This destroys the orthonormality of the outer matrices. At the same time, with every summation, the compression rate is progressively lost. In order to restore both of them, we reorthonormalize the sequences representing the decomposed blocks A_{ij} in steps 6) and 10). This is done according to the following algorithm.

```
Algorithm Reorthonormalize(X):
1) set U,S and V to the three factors of the sequence $\mathbf{X}$
2) $(\mathrm{Q}_1, \mathrm{R}_1) = \mathrm{QR}(\mathrm{U})$; $(\mathrm{Q}_2', \mathrm{R}_2') = \mathrm{QR}(\mathrm{V}')$
3) $(\mathrm{U}_1, \mathrm{S}, \mathrm{V}_1) = \mathrm{SVD}(\mathrm{R}_1\mathrm{S}\mathrm{R}_2')$
4) $\mathrm{U} = \mathrm{Q}_1\mathrm{U}_1$; $\mathrm{V} = \mathrm{V}_1'\mathrm{Q}_2'$
5) replace the three factors of $\mathbf{X}$ with U,S and V
End algorithm
```

The QR decompositions in the reorthonormalization algorithm can be optimized making use of the fact that the input consists of a concatenation of two matrices, which are themselves already orthonormal.

Suppose that the matrix U in the reorthonormalization algorithm consists of two column-wise orthonormal matrices U_1 and U_2. We first determine which is the larger of the two. Supposing that U_1 has more columns than U_2, the algorithm goes as follows ([X Y] denotes horizontal concatenation and [X;Y] denotes vertical concatenation).

```
Algorithm horzcat_qr($[\mathbf{U_1}\ \mathbf{U_2}]$):
1) $\mathrm{A} = \mathrm{U}_1'\mathrm{U}_2$
2) $\mathrm{W}_2 = \mathrm{U}_2 - \mathrm{U}_1\mathrm{A}$
3) $(\mathrm{Q}_2, \mathrm{R}_2) = \mathrm{QR}(\mathrm{W}_2)$;
4) $\mathrm{Q} = [\mathrm{U}_1\ \mathrm{Q}_2]$
5) $\mathrm{R} = [[\mathrm{I}\ \mathrm{A}]; [0\ \mathrm{R}_2]]$
End algorithm
```

If U_2 has more columns than U_1, the algorithm is trivially adapted. For vertical concatenation, the same algorithm is applied to the adjoined matrices.

Regarding the actual recompression inside the reorthonormalization algorithm, in line 3, we used $\mathrm{SVD}(\tau)$ with the same threshold τ as in the initial compression. In our experience, this leads to a decomposed matrix of approximately the same size as the original one in (5) and an error in the direct CBD solution of the same order of magnitude as the error in the MDA-SVD compression of matrix subblocks.

The previous is illustrated with several representative numerical examples in Section IV. It is necessary for the compressed matrices to be true SVD decompositions, that is, an inner diagonal matrix of singular values and two outer orthonormal matrices. Consequently, unlike with iterative MDA, we cannot suffice with just an MDA compression, as was mentioned in the introduction.

C. Computational Complexity

The computational complexity of the method is the relation between the number of unknowns N of a problem and the number of elementary operations needed by the algorithm, for asymptotically large N. We analyze it for radiation and scattering problems in free space. Consider a problem with N

unknowns, subdivided into M subgroups of approximately $\overline{n}$ unknowns, such that $N = \overline{n}M$. The expensive operations in the algorithm are as follows:

1) inversion of the diagonal blocks [step 14) in the CBD algorithm]. The number of elementary operations for this step is proportional to

$$O_{\text{inv}} \sim M\overline{n}^3. \tag{7}$$

2) multiplications and summations (which include two QR and one SVD operations, see Section III-B).

If S_i denotes the rank of the compressed matrices, equal to the number of singular values, then, the summations scale with $S_i\overline{n}$ and the SVD scales with S_i^3, while the multiplications and the QR scale with $S_iS_j\overline{n}$. Since for far-field subdomains $S_i \ll \overline{n}$, the multiplications and QR are asymptotically the most expensive operations in the algorithm. From the CBD algorithm, we see that the total number of all of these operations needed is determined by the three loops over the M groups, and, therefore, proportional to M^3. However, the cost per operation varies from group to group, depending on the S_i and S_j of the two blocks involved in the operation. We can account for this, by writing the total operation cost (for the asymptotically dominant operations QR and multiplication) as

$$O_{\text{qr}} \sim M\overline{n}\left(\sum_{i=1}^{M} S_i\right)^2 \tag{8}$$

where the leading M reflects the outer loop in the algorithm and the squared summation the two inner loops. The total number of interactions K is proportional to M^2.

The maximum rank for an interaction is obviously $\overline{n}$. For interactions between blocks representing touching boxes (labeled zero) we can say nothing further, so we have

$$S_0 \sim \overline{n}. \tag{9}$$

The number of touching observation boxes for one source box is a constant, so the operation count for all the touching boxes equals

$$O_{\text{near}} \sim M\overline{n}^3. \tag{10}$$

We subdivide the remaining interactions into levels with index p according to the mutual distance of the concerned boxes. For every source box, the first level $(p = 1)$ interactions are with the nearest nontouching observation boxes, the next layer is level two, etc. The rank is proportional to $1/D^2$ [see Appendix A, (17)], where D is the distance between the source- and observation box, until it stabilizes at some minimum value S_∞, when, to the required precision, the boxes are point-sources to each other. S_∞ equals two because there are two independent polarizations in the far field. If all the interactions were rank S_∞, the complexity of (8) would equal $M^3\overline{n}$ and the total complexity would be

$$O_{\text{tot}} \sim (O_{\text{inv}} + O_{\text{near}}) + O_{\text{qr}} \sim M^3\overline{n} + M\overline{n}^3 \tag{11}$$

which, obviously, can be minimized by choosing $\overline{n}$ and M such that they scale with N as

$$\overline{n} \sim M \sim \sqrt{N} \tag{12}$$

yielding $O_{\text{tot}} \sim N^2$. In reality, the nearby interactions have a rank $S_i > S_\infty$. Clearly, for an asymptotically large problem it is always possible to choose many small boxes, such that the number of far interactions of rank S_∞ is much larger than the number of near interactions K', but this is not enough to prove that the algorithm scales with N^2. For that, we must show that, if we scale up a problem by a factor χ, the proportion of minimum-rank interactions does not change. To this aim, we investigate the distribution of the number of nonempty observation boxes n_p in layer p. This depends on the problem geometry at the scale of the boxes. At one extreme, we have a very elongated (1-D) problem structure (labeled $\alpha = 1$), for example, a large dipole, such that the boxes are all on a straight line. In this case, $n_p \sim 1$, and the total number of unknowns N scales with χ. At the other extreme, we may have a problem that entirely fills all the boxes in a 3-D region in space, for example a large 3-D array of small elements $(\alpha = 3)$. This leads to $n_p \sim p^2$ and $N \sim \chi^3$. For a 2-D $(\alpha = 2)$ structure, like a metal plate, we find $n_p \sim p$ and $N \sim \chi^2$. In general, for curved 1-D and 2-D cases (e.g., loops, spheres), the given relation between p and n_p will generally only hold locally. However, this is sufficient, since the initial box-size can be chosen freely (arbitrarily small), and we are addressing asymptotically large problems. Consequently, we can make sure the relation holds for the first several layers p, up to the first layer for which $S_i \sim S_\infty$, which is sufficient for our argument.

Therefore, barring very specific problems that exhibit high curvatures at *any* scale (fractal-like geometries), we can characterize a structure by its dimension α and we have

$$n_p \sim p^{\alpha-1} \quad N \sim \chi^\alpha. \tag{13}$$

The distance D' (and corresponding p') at which the rank has reduced to S_∞ scales with χ [see Appendix A, (20)]. Furthermore, if the rib-length of the boxes equals d, then, on the one hand, we have $p \sim D/d$, and, on the other hand, we have

$$d^\alpha \sim \overline{n} \sim \sqrt{N} \sim \sqrt{\chi^\alpha}$$

or $d \sim \sqrt{\chi}$. Therefore, $p' \sim D'/d \sim \sqrt{\chi}$ and the number of near interactions for one box equals

$$K' \sim \sum_{p=1}^{\sqrt{\chi}} n_p \sim \chi^{\alpha/2}. \tag{14}$$

But, with (12), the total number of interactions $K \sim M \sim N^{1/2} \sim \chi^{\alpha/2}$. Therefore, the proportion of S_∞-interactions is independent of N, and the algorithm, with the proper initial choice for $\overline{n}$, has a complexity $O_{\text{tot}} \sim N^2$.

Observing that the storage of one block scales with $S\overline{n}$, we find that the total asymptotic storage requirement for nondiagonal matrix blocks is

$$O_{\text{stor}} \sim M^2\overline{n} + M\overline{n}^2 \tag{15}$$

TABLE I

PERFORMANCE PARAMETERS FOR 16-BLOCK CBI APPLIED TO A $4\lambda \times 4\lambda$ SQUARE PLATE WITH 4720 UNKNOWNS. SVD THRESHOLD τ, MATRIX BUILDING AND INVERSION SIZE S_c AND S_f, BUILDING AND INVERSION TIME t_c AND t_f. RELATIVE ERROR ε

τ	S_c^*	S_f^*	$t_c(s)$	$t_f(s)$	$\varepsilon(\%)$
10^{-2}	36.6	35.6	10.1	9.2	4.07
10^{-3}	46.8	45.9	12.7	13.8	0.55
10^{-4}	58.5	56.5	17.0	23.1	0.05
10^{-5}	70.8	69.4	23.0	36.9	0.01
*Size in MB (double precision)					

TABLE II

PERFORMANCE PARAMETERS FOR CBI APPLIED TO A $4\lambda \times 4\lambda$ SQUARE PLATE WITH 4720 UNKNOWNS. SVD THRESHOLD $\tau = 0.001$, NUMBER OF BLOCKS M, MATRIX BUILDING AND INVERSION SIZE S_c AND S_f, BUILDING AND INVERSION TIME t_c AND t_f. RELATIVE ERROR ε

M	S_c^*	S_f^*	$t_c(s)$	$t_f(s)$	$\varepsilon(\%)$
9	82.7	77.2	16.9	20.6	0.59
16	46.8	45.9	12.7	13.8	0.55
25	42.4	42.2	11.1	19.3	1.11
36	42.2	43.4	11.6	35.3	0.89
49	44.6	47.3	13.8	64.8	0.81
*Size in MB (double precision)					

TABLE III

PERFORMANCE PARAMETERS FOR CBI APPLIED TO SQUARE PLATES SVD(0.001). NUMBER OF BASIS FUNCTIONS N, NUMBER OF BLOCKS M, MATRIX BUILDING AND INVERSION SIZE S_c AND S_f, BUILDING AND INVERSION TIME t_c AND t_f. RELATIVE ERROR ε

Plate size	N	M	S_c^*	S_f^*	$t_c(s)$	$t_f(s)$	$\varepsilon(\%)$
$8\lambda \times 8\lambda$	19040	25	395	395	86	180	1.01
$12\lambda \times 12\lambda$	42960	36	1314	1364	288	900	1.29
*Size in MB (double precision)							

and, for the diagonal blocks, $O_{\text{diag}} = M\overline{n}^2$. Thus, with (12), the storage scales with $N^{3/2}$.

It should be emphasized again that the previous analysis poses no lower limit on the box-size. For instance, if we choose boxes such that $\overline{n} = 1$ (one elementary scatterer per box), and, when upscaling the problem we obey (12), the predicted complexity will be observed. This, however, is evidently not optimum in practice. In fact, because of the constant factors leading the various complexity terms, it is generally advantageous to choose few and large boxes (see also Section IV-A).

IV. NUMERICAL EXAMPLES

All the numerical experiments reported in this section have been performed on a PC with 8 GB of RAM and an AMD Opteron processor at 2.2 GHz. The computations were done in MATLAB 7.0.4 (the problems in Section IV-A) and in MATLAB 7.3.0, always using only one CPU.

Three different perfectly conducting structures in free space were analyzed, all discretized using RWG basis functions [14]. First, a series of square plates of varying sizes to investigate the scaling of various performance parameters with the problem size. Second, a sphere, to evaluate the performance of the method on a full 3-D problem and, third, a pyramidal X-band horn, to give an example of a practical application of the method. The analyzed objects exhibit four-fold symmetry, which allowed to compute the direct LU solution with no compression, but with exactly the same discretization as the CBD computation, as a reference for error computation. The symmetry was ignored in the CBD computations in order to test the algorithm with a large number of unknowns.

The reported relative errors due to the CBD are calculated as

$$\varepsilon = \frac{|\mathbf{y} - \mathbf{y}_d|}{|\mathbf{y}_d|}$$

where $\mathbf{y}$ is the computed solution coefficient vector and $\mathbf{y}_d$ is the coefficient vector of the direct noncompressed solution using four-fold symmetry. Consequently, ε is a measure of the additional error (on top of the inherent discretization errors in the MoM) due to the MDA-CBD compression.

A. Square Plates

We have started with a relatively small problem, a $4\lambda \times 4\lambda$ square plate, discretized into $N = 4720$ RWG basis functions with an average edge length of 0.114λ. The plate was subdivided into $M = 16$ subdomains of $1\lambda \times 1\lambda$ with an approximate block size of $\overline{n} = 300$ basis functions. Table I shows the performance of CBD for various SVD compression parameters τ. As Table I shows, the compressed matrices and the decomposed matrices are approximately equal in size. The choice of τ for any particular problem is a compromise between computational effort and desired accuracy.

Next, we investigated the performance as a function of the number of blocks, for a fixed value of $\tau = 0.001$. The results are shown in Table II. For this particular problem, a choice of 16 blocks seems optimum in terms of decomposition speed, and near optimum in terms of storage. One observes that the decomposition time grows strongly with the number of blocks. In fact, of the two computationally expensive operations O_{inv} and O_{qr} from Section III-C, O_{qr}, which grows strongly with M, has a much higher leading constant. Therefore, if allowed by the available memory, it is generally recommendable to choose few, large blocks.

To test the derived complexity (see Section III-C), we discretized two larger plates of $8\lambda \times 8\lambda$ and of $12\lambda \times 12\lambda$ with the same average RWG edge length as shown before. The resulting number of basis functions was $N_1 = 19\,040$ and $N_2 = 42\,960$, respectively, with $N_2/N_1 \approx 2.26$. We chose to subdivide the plates into $M_1 = 25$ and $M_2 = 36$ subdomains, which complies approximately with the theoretical optimum of $M_2/M_1 = \sqrt{N_2/N_1}$. The diameters of the spheres enclosing the resulting subdomains are $d_1 = 2.77\lambda$ and $d_2 = 3.46\lambda$, respectively.

Table III shows the performance of CBD with SVD(0.001). In Fig. 3, the CBD decomposition time is shown for the two plates and compared with an N^2-slope. The slopes correspond remarkably well.

Incidentally, the matrix construction time using MDA should scale with $N^{3/2}$, and indeed it does, since $t_{c1}(N_2/N_1)^{3/2} = 292$ s while $t_{c2} = 288$ s. If no MDA is used, the matrix construction time can be shown to scale with N^2, using the technique explained at the end of Section II.

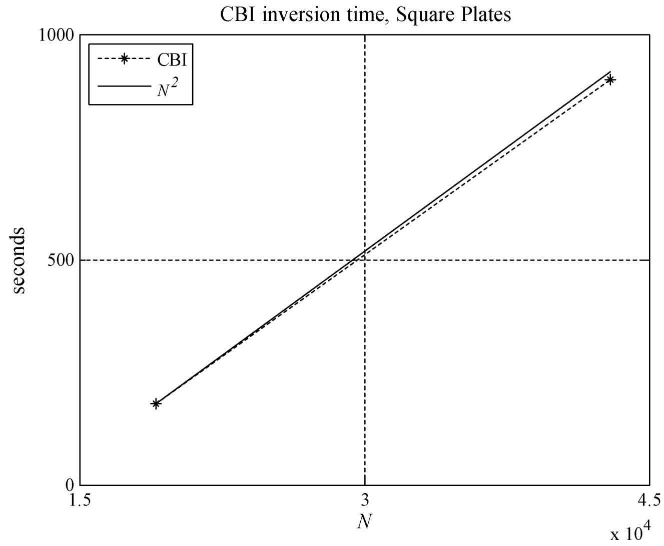


Fig. 3. CBD computation time versus number of unknowns for the square plates from Section IV-A.

TABLE IV
PERFORMANCE PARAMETERS FOR CBI APPLIED TO SPHERES. COMPRESSION USING SVD(0.001). SPHERE RADIUS R, NUMBER OF UNKNOWNS N, NUMBER OF BLOCKS M, MATRIX BUILDING AND INVERSION SIZE S_c AND S_f, BUILDING AND INVERSION TIME t_c AND t_f. RELATIVE ERROR ε

$R(\lambda)$	N	M	$S_c(MB)$	$t_c(s)$	$S_f(MB)$	$t_f(s)$	$\varepsilon(\%)$
1.43	10200	26	171	82	181	107	1.62
2.86	44508	26	2040	872	2254	1927	1.99
2.86	44508	55	1530	679	1929	2439	2.69

B. Spheres

In theory, there is no fundamental difference in the application of the method to 2-D geometries like the square plates from Section IV-A or to a 3-D geometry, like a sphere. However, the sphere is considerably more compact (more unknowns for the same electrical size), and the impedance matrix conditioning is much worse if we compare similar discretizations and number of unknowns [8]. These two circumstances are expected to degrade the CBD performance, just like they do for iterative methods like MDA and MLFMA. It should be mentioned though, that closed surfaces like spheres can be analyzed with the combined field integral equation (CFIE) which is typically much better conditioned. This results in much better convergence for the iterative methods, while the CBD performance is not expected to change much.

We analyzed two PEC spheres, of radius $R = 1.43\lambda$ and $R = 2.86\lambda$, calculating the surface current and the RCS due to an incident plane wave. The results are summarized in Table IV.

Comparing the largest sphere with the largest square plate, we do observe that the matrix sizes are larger and the computation times longer. This is because the sphere is electrically smaller, and, therefore, less compressible. The spheres are too small to serve as illustration for the derived complexity from Section III-C.

Just as with the square plates, using more, smaller subdomains leads to a longer decomposition time, and a slightly higher error, but more compression. Finally, Fig. 4 compares the RCS for the largest sphere with the exact (Mie series) solution, with excellent agreement.

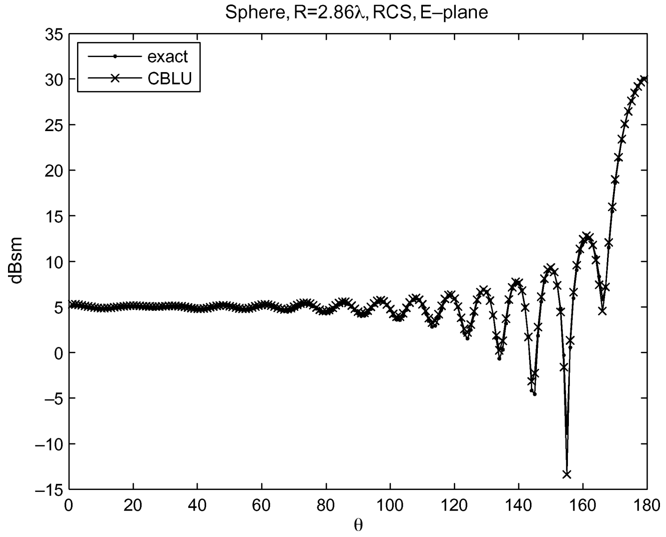
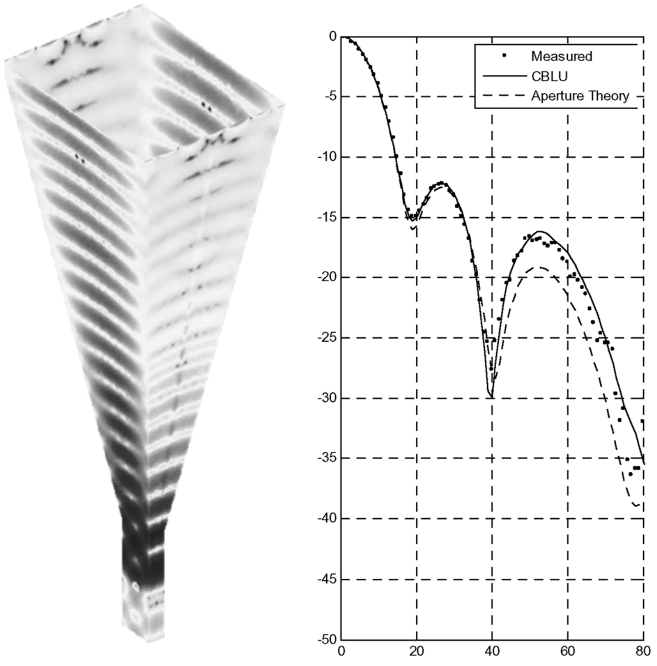


Fig. 4. RCS of a PEC sphere, computed using the Mie series and using CBD with $N = 44508$.



(*a*) (*b*)

Fig. 5. CBD analysis of a 3λ × 4λ X-band horn: (a) real part of the induced current distribution (inner plus outer sides of the surface) and (b) radiation pattern compared with measurements and aperture theory Kirchoff aproximation.

C. Pyramidal Horn

Last, we present an example of the CBD applied to the analysis of a practical antenna: the same pyramidal X-band horn as was analyzed in [8]. The pyramidal section has a length of 8λ, the aperture measures $3\lambda \times 4\lambda$, and the feeding consists of a 2λ section of single mode waveguide with an infinitesimal dipole excitation at the center [see Fig. 5(a)]. As was shown in [8], with RWG basis functions, a fine discretization of approximately 0.06λ must be used to achieve convergence of the radiation pattern, leading to $N = 69228$ basis functions.

Table V shows the performance of CBD on this problem. The results show that, while the CBD does not seem to allow for a

TABLE V
PERFORMANCE PARAMETERS FOR X-BAND HORN, $N = 69228$. NUMBER OF BLOCKS $M = 52$. SVD THRESHOLD τ. SOLVE TIME t_s. OTHER SYMBOLS AS IN TABLE IV

CBLU						
τ	$S_c(GB)$	$t_c(s)$	$S_f(GB)$	$t_f(s)$	$t_s(s)$	$\varepsilon(\%)$
0.01	2.87	998	3.55	2171	2.7	11.2
0.001	3.43	1223	3.69	3436	3.5	1.37
0.0001	4.04	2528	4.37	5233	4.3	0.17

very fast, highly approximative solution (with an SVD threshold τ as high as 1% the total computation time is still almost 1 h), on the other hand, the computational requirements do not grow excessively with higher accuracy (about 1 h and 20 min for $\tau = 0.1\%$). This indicates that the CBD is particularly interesting when high accuracy is required. In our experience, this contrasts with fast iterative methods like MLFMA where highly approximative results can be obtained very fast, but for high accuracy, the computational requirements grow quickly.

Another noteworthy result is the solution time reported in Table V. This is the time for multiplying the decomposed matrix with one excitation vector. It is virtually negligible compared with the matrix setup and decomposition time. This step consists of only matrix-vector products, that are performed efficiently by level 2 BLAS routines [16]. For multiple excitation vectors, even more efficient level 3 BLAS routines for matrix-matrix products can be used.

Fig. 5(b) shows the radiation pattern computed with CBD and $\tau = 0.001$, compared to a measurement and to the result from aperture theory Kirchoff approximation. As expected, beyond forty degrees from broadside direction CBD matches the measurements much closer than the aperture theory approximation.

V. CONCLUSION

A new algorithm, the CBD technique, has been presented. It allows for the direct solution of scattering and radiation problems with the MoM, with highly reduced computational and storage requirements compared to conventional LU decomposition. For scattering and radiation problems in free space, the computational complexity of the algorithm is shown to be of order N^2 and the storage requirements of order $N^{3/2}$.

The surface current on a perfectly conducting sphere, discretized with almost 45 000 RWG basis functions, is computed in less than 1 h. A pyramidal horn antenna, with almost 70 000 unknowns takes about 1 h and 20 min. In both cases, the difference with the direct, uncompressed LU solution is about 1%. The storage requirement for these problems is about 2 and 4 GB, respectively.

A major advantage of the presented algorithm is the small solution time per incident field vector once the impedance matrix is decomposed. For the previous two problems, this takes only a few seconds. Consequently, the method is particularly useful for problems with multiple incident fields such as monostatic RCS computations.

Another noteworthy advantage is the fact that only two parameters need to be optimized for a specific problem: the compression threshold and the number of blocks. The first directly controls the accuracy versus computational cost tradeoff, while the second is directly linked to the available computer memory. This makes the practical application of CBD very simple.

In this paper, only perfectly conducting structures in free space have been addressed. However, since the MDA method used for the initial matrix compression is independent of the specific Greens function of the problem, as long as it decreases with the source-observer distance, as is commonly the case, and the rest of the algorithm is entirely algebraic, the proposed technique is applicable to almost any electromagnetic problem provided it is discretized with local basis functions.

As a final remark, we have not yet studied the numerical stability of the CBD method. We intend to report on this issue in a future paper. However, we have not observed any instability in any of the reported or any other numerical experiments.

APPENDIX A

We subdivide the 3-D region containing the problem structure into cubic boxes of riblength d. We address the interaction between two such boxes, at a mutual minimum distance $D \geq d$. Now we construct the two minimum radius spheres that entirely enclose both boxes. This radius equals $R \sim d$. From [18], we know that, as long as the two spheres do not overlap, the field radiated by the first, evaluated at the second, is entirely and uniquely determined by the far field. In [17], it is shown that, for two spherical arrays, in each other's far field, both of effective aperture

$$A = \pi R^2 \tag{16}$$

[17, (38)], the interaction has a number of spatial degrees-of-freedom (rank) given by, as a function of the solid angle subtended by either sphere as seen from the center of the other $\Omega \sim d^2/D^2$

$$S = 2A\Omega \tag{17}$$

[17, (41)]. Consequently, the rank is proportional to $1/D^2$.

The degrees-of-freedom reach a minimum (the spheres become point-sources to each other), when the maximum difference Δ in path length over the aperture A vanishes with respect to the wavelength λ

$$\Delta = d \sin\theta \sim d^2/D = \gamma\lambda \quad \gamma \ll 1 \tag{18}$$

with $\theta = \sqrt{\Omega} = d/D$. Thus, for fixed box-size d and fixed precision γ, the distance at which this happens is

$$D_\infty \sim 1/\lambda. \tag{19}$$

Consequently, if we scale up a problem by a factor χ, by multiplying the frequency with χ and refining the discretization accordingly

$$D_\infty \sim \chi. \tag{20}$$

REFERENCES

[1] R. F. Harrington, *Field Computation by Moment Methods*. New York: MacMillan, 1968.

[2] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, no. 5, pp. 1225–1251, Sep./Oct. 1996.

[3] J. Song, C. C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.

[4] E. Michelsen and A. Boag, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures," *IEEE Trans. Antennas Propag.*, vol. 44, no. 8, pp. 1086–1093, Aug. 1996.

[5] J. M. Rius, J. Parrón, E. Úbeda, and J. R. Mosig, "Multilevel matrix decomposition algorithm for analysis of electrically large electromagnetic problems in 3-D," *Microw. Opt. Technol. Lett.*, vol. 22, no. 3, pp. 177–182, Aug. 1999.

[6] J. Parrón, J. M. Rius, and J. R. Mosig, "Application of the multilevel decomposition algorithm to the frequency analysis of large microstrip antenna arrays'," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 721–724, Mar. 2002.

[7] Y. Saad*, Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS Publishing Co., 1996.

[8] A. Heldring, "Full wave analysis of electrically large reflector antenna," Ph.D. dissertation, Dept. Elect. Eng., Delft Univ. Technol., Delft, The Netherlands, April 2002.

[9] A. Heldring, J. M. Rius, and L. Ligthart, "New block ILU preconditioner scheme for numerical analysis of very large electromagnetic problems," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 337–340, Mar. 2002.

[10] A. Heldring, J. M. Rius, L. P. Ligthart, and A. Cardama, "Accurate numerical modeling of the TARA reflector system," *IEEE Trans. Antennas Propag.*, vol. 52, no. 7, pp. 1758–1766, Jul. 2004.

[11] K. Sertel and J. L. Volakis, "Incomplete LU preconditioner for FMM implementation," *Microw. Opt. Technol. Lett.*, vol. 26, no. 4, pp. 265–267, Aug. 2000.

[12] W. H. Press*, Numerical Recipes in C*, 2nd ed. Cambridge, MA: Cambridge Univ. Press, 1994.

[13] G. H. Golub and C. F. Van Loan*, Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[14] R. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shapes," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.

[15] R. Coifman, V. Rohklin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian description," *IEEE Antennas Propag. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.

[16] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling, "A set of level 3 basic linear algebra subprograms," *ACM Trans. Math. Soft.*, vol. 16, pp. 1–17, 1990.

[17] A. S. Y. Poon, R. W. Brodersen, and D. N. C. Tse, "Degrees of freedom in multiple-antenna channels: A signal space approach," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 523–536, Feb. 2005.

[18] A. J. Devaney and E. Wolf, "Radiating and nonradiating classical current distributions and the fields they generate," *Phys. Rev. D*, vol. 8, no. 4, pp. 1044–1047, Aug. 1973.

Alex Heldring was born in Amsterdam, The Netherlands, on December 12, 1966. He received the M.S. degree in applied physics and the Ph.D. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1993 and 2002, respectively.

He is currently an Assistant Professor with the Antenna Laboratory, Department of Signal Processing and Telecommunications, Universitat Politecnica de Catalunya, Barcelona, Spain. His special research interests include integral equation methods for electromagnetic problems and wire antenna analysis.

Juan M. Rius (M'89) received the "Ingeniero de Telecomunicación" degree and the "Doctor Ingeniero" degree from the Universitat Politòcnica de Catalunya (UPC), Barcelona, Spain, in 1987 and 1991, respectively.

In 1985, he joined the Department of Signal Theory and Telecommunications (TSC), Electromagnetic and Photonic Engineering Group, UPC, where he currently holds a position of "Catedrático" (equivalent to Full Professor) with the Antenna Laboratory, Department of Signal Processing and Telecommunications. From 1985 to 1988, he developed a new inverse scattering algorithm for microwave tomography in cylindrical geometry systems. Since 1989, he has been engaged in the research for new and efficient methods for numerical computation of electromagnetic scattering and radiation. He is the developer of the Graphical Electromagnetic Computation (GRECO) approach for high-frequency RCS computation, the Integral Equation formulation of the Measured Equation of Invariance (IE-MEI), and the Multilevel Matrix Decomposition Algorithm (MLMDA) in 3-D. His current research interests are the numerical simulation of electrically large antennas: multiband fractal antennas, large microstrip arrays, and reflector antennas. He has held positions of "Visiting Professor" with EPFL, Lausanne, France, from May 1, 1996 to October 31, 1996; "Visiting Fellow" with the City University of Hong Kong, Hong Kong, from January 3, 1997 to February 4, 1997; "CLUSTER chair" with EPFL from December 1, 1997 to January 31, 1998; and "Visiting Professor" with EPFL from April 1, 2001 to June 30, 2001. He has more than 40 papers published or accepted in refereed international journals (17 in IEEE Transactions) and more than 100 in international conference proceedings.

José Maria Tamayo was born in Barcelona, Spain, on October 23, 1982. He received the Degree in Mathematics and the Degree of Telecommunications Engineering from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, both in 2006.

In 2004, he joined the Telecommunications Department, Universitat Politecnica de Catalunya, where he is currently coursing the Ph.D. degree in Signal and Communication Theory with the Antenna Laboratory, Department of Signal Processing and Telecommunications. His current research interests include accelerated numerical methods for solving electromagnetic problems and parallelization.

Josep Parrón was born in Sabadell, Barcelona, Spain, in 1970. He received the telecommunication engineer degree and the Doctor Ingeniero degree from the Politechnic University of Catalunya (UPC), Barcelona, Spain, in 1994 and 2001, respectively.

Since June 2002, he has been an Associate Professor with the Department of Telecommunications and Systems Engineering, Universitat Autònoma de Barcelona, Bellaterra, Spain. From September 2000 to May 2002, he was an Assistant Professor with UPC. His main research interests include fast algorithms for computational electromagnetics and their application to fractal antennas, multilayered mediums, and superconductors.

Eduard Úbeda was born in Barcelona, Spain, in 1971. He received the Telecommunication Engineer degree and the Doctor Ingeniero degree from the Politechnic University of Catalonia (UPC), Barcelona, Spain, in 1995 and 2001, respectively.

Since 2003, he has been an Associate Researcher with the Antenna Laboratory, Department of Signal Processing and Telecommunications, Universitat Politecnica de Catalunya, Barcelona, Spain. In 1996, he was with the Joint Research Center, from the European Commission, Ispra, Italy. From 1997 to 2000, he was a Research Assistant with the Electromagnetic and Photonic Engineering Group, UPC. From 2001 to 2002, he was a Visiting Scholar in the Electromagnetic Communication Laboratory, the Electrical Engineering Department, the Pennsylvania State University (PSU), Philadelphia. He is the author of 12 papers in international journals and 27 in international conference proceedings. His main research interests include numerical computation of scattering and radiation using integral equations.