Sparsified Adaptive Cross Approximation Algorithm for Accelerated Method of Moments Computations

A. Heldring, J.M. Tamayo, C. Simon, E. Ubeda and J.M. Rius, *Senior Member, IEEE*

Abstract—This paper presents a modification of the Adaptive Cross Approximation (ACA) algorithm for accelerated solution of the method of moments linear system for electrically large radiation and scattering problems. As with ACA, sub-blocks of the impedance matrix that represent the interaction between well separated sub-domains are substituted by ‘compressed’ approximations allowing for reduced storage and accelerated iterative solution. The modified algorithm approximates the original sub-blocks with products of sparse matrices, constructed with the aid of the ACA algorithm and of a sub-sampling of the original basis functions belonging to either sub-domain. Because of the sampling, an additional error is introduced with respect to ACA, but this error is controllable. Just like ordinary ACA, Sparsified ACA is kernel-independent and needs no problem-specific information except for the topology of the basis functions. As a numerical example, RCS computations of the NASA almond are presented, showing an important gain in efficiency. Furthermore, the numerical experiment reveals a computational complexity close to $N \log N$ for sparsified ACA for a target electrical size of up to 50 wave lengths.

Index Terms—Computational electromagnetics, Numerical simulation, Method of Moments, fast solvers, impedance matrix compression

I. INTRODUCTION

The solution of the surface integral equation formulation of electrically large radiation and scattering problems by way of the Method of Moments is notoriously expensive in terms of computation time and storage requirements because a fully populated linear system needs to be solved. In recent years, many methods to mitigate this problem have been published allowing for ever faster solution of ever larger problems. A common approach is to substitute the Method of Moments impedance matrix with a much smaller approximative representation and solve the system iteratively. A few examples of this approach are: AIM [1], MLFMA [2], MLM [3][4], SVD-MDA [5] and $\mathcal{H}^2$ -matrices [6]. A recent addition to this family of accelerated MoM solvers is a method based on Adaptive Cross Approximation or ACA, a linear algebra algorithm proposed by Bebendorf [7] and introduced in the field of electromagnetics by Zhao et. al. [8]. The ACA was originally conceived for solving problems with an asymptotically smooth kernel, which excludes the oscillating kernel of

electromagnetics, but Zhao et. al. showed that, despite a lack of mathematical rigour, in practise the ACA is both efficient and accurate for electromagnetic problems as well.

The ACA based method has in common with practically all of the accelerated solvers a decomposition of the computational domain into subdomains, leading to a block subdivision of the impedance matrix. Subsequently, those blocks that represent interactions between spatially separated subdomains are compressed using the ACA algorithm. Although the ACA based method is not as efficient as, for instance, MLFMA for very large problems, it is competitive for medium sized problems (in the order of 100,000 unknowns), and its major advantage is the fact that it is entirely algebraic; no problem specific, or kernel specific information is needed beyond the geometry information used for the domain decomposition, in contrast to all of the methods mentioned above. It is therefore very easy to build an implementation with a very general range of applicability.

The compressibility of matrix blocks representing mutually distant subdomains is based on the well known phenomenon that the maximum number of degrees of freedom (DoF) present in the interaction between distant groups of elementary scatterers decreases rapidly with the distance between the two groups [9]. Applying a Singular Value Decomposition (SVD) on a block reveals a steeply descending slope for the singular value magnitudes. Compressing the matrix block then consists in discarding all singular vectors corresponding to singular values below a given threshold. The ACA algorithm is essentially an efficient approximation of the SVD.

Unfortunately, the number of DoF grows with the electrical size of the subdomains. Consequently, electrically large subdomains lead to inferior compression rates. This constitutes the main limitation of the ACA based method for electrically large problems. One way to overcome this limitation is the Multilevel ACA [10], which converts the compressed matrix blocks into sequences of sparse matrices, much like fast multipole methods or the FFT. However, the compression phase of ML-ACA is computationally expensive, to such an extent that it only starts to outperform ordinary ACA for problems of almost a million unknowns. What is needed is an accelerated method to further compress the ACA matrices that 1) does not introduce an unacceptable additional approximation error and 2) preserves the advantages related to the algebraic nature of ACA. In this paper we propose such a method, the Sparsified ACA or SPACA.

A. Heldring, J.M. Rius and E. Ubeda are with the AntennaLab, Dept. of Signal Processing and Telecommunications, Universitat Politècnica de Catalunya, Edifici D3, Jordi Girona 1-3, 08034 Barcelona, Spain. email: heldring@tsc.upc.edu, fax: +34 93 401 72 32.

J.M. Tamayo is with Institut Supérieur de l'Aéronautique et de l'Espace, Campus ENSICA, 1 Place E. Blouin, 31500 Toulouse, France

C. Simon is with Unidad de Tecnología Marina, Consejo Superior de Investigacion Científica, Barcelona, Spain

II. SPARSIFIED ACA

A. ACA Based Solver

An iterative solver for scattering and radiation problems based on ACA was first proposed in [8]. The problem geometry is subdivided hierarchically into many subdomains. Impedance matrix sub blocks representing the interactions of the domains with themselves or with immediate neighbours (domains that touch one another) are either subdivided again or fully computed if they are at the bottom of the hierarchy. All other sub-blocks are compressed with the ACA algorithm into a product of two matrices

$$UV^T \quad (1)$$

with dimensions $m \times r$ and $r \times n$, respectively, where m, n are the original block dimensions and $r \ll m, n$ is the rank of the approximation. For the details of the ACA algorithm and the iterative solution we refer to [8]. We use a binary tree for the hierarchical subdivision by recursively splitting the domains in half along the dimension of largest extent. The splitting is based on the number of basis functions. At every level except the lowest (smallest domains), matrix blocks that represent non touching domains are compressed using ACA, while touching blocks are subdivided. Our implementation differs from [8] in a number of aspects which we explain and justify in this section:

1) *Applying ACA to All Off-Diagonal Blocks:* Domain decomposition methods like the MLFMA and MLMDA mentioned in the introduction use the physics of the problem to construct an approximation to the interaction between domains. This generally prohibits the use of this approximation for touching domains. In [8], this restriction was maintained. For the ACA algorithm however, precisely because it is purely algebraic, there is no formal restriction at all. On the other hand, it is likely that the error of the ACA approximation for touching domains has a greater effect on the general solution. This can be mitigated by lowering the ACA threshold, at the cost of reduced compression. We made a test by computing the surface current on the NASA almond [11], discretized into 38,706 RWG basis functions, illuminated by a plane wave at 12.1 GHz. First we used direct LU decomposition as a reference, then the ACA for a range of thresholds. The ACA computations were repeated with and without compressing the touching domains at the lowest level. We used the CFIE formulation, RWG basis functions [12] and a GMRES iterative solver. The decomposition was an $L = 7$ level binary tree. Table I shows the results. One observes that the relative error (in the norm of the surface current coefficient vector) is indeed much better without compression for the same threshold. On the other hand, the gain in storage when we do use compression is important. We chose to use compression for the lowest level touching blocks because in our experience, the storage limitation is often the first bottleneck encountered when scaling up a problem. Regarding the acceptable relative error, this is very much dependent on the context of the problem. For instance, if the objective of the computation is a far-field quantity (RCS, radiation pattern), then a relative error in the surface current up to 1% is generally acceptable.

The threshold of 10^{-4} for the ACA that we chose to use in this paper fulfills this criterion with a generous margin for the target under investigation, as shown in the table.

TABLE I
COMPARISON OF ACA WITH AND WITHOUT COMPRESSING TOUCHING DOMAINS AT THE LOWEST LEVEL. TEST OBJECT: NASA ALMOND, 38706 UNKNOWNNS.

ACA threshold	setup (sec)	size (MB)	rel. error
compress			
10^{-3}	191	459	2.3×10^{-2}
10^{-4}	328	724	1.5×10^{-3}
10^{-5}	471	1037	1.2×10^{-4}
no compress			
10^{-3}	157	1373	4.3×10^{-3}
10^{-4}	226	1523	3.3×10^{-4}
10^{-5}	329	1713	3.2×10^{-5}

2) *ACA-SVD:* A second adaptation with respect to [8] involves the post-compression of the ACA matrices using QR decomposition and thresholded SVD ($SVD(\tau)$, indicating that all singular values σ below $\tau\sigma_{\max}$ are dropped). This procedure is explained in detail in [5]. We have observed experimentally that the SVD threshold can be chosen a factor 10 higher than the ACA threshold, without significant loss of accuracy. Table II shows the performance on the same test case as in section II-A1.

TABLE II
COMPARISON OF ACA WITH AND WITHOUT SVD POST COMPRESSION. TEST OBJECT: NASA ALMOND, 38706 UNKNOWNNS. ACA THRESHOLD 10^{-4} , SVD THRESHOLD 10^{-3} .

SVD post-compression	setup (sec)	size (MB)	rel. error
Yes	328	724	1.5×10^{-3}
No	255	1378	1.2×10^{-3}

B. Sparsifying a Matrix Block

With the procedure described above, at higher levels of the tree hierarchy there are large matrix blocks that represent the interaction between electrically large domains (assuming that the number of unknowns per wave length is globally constant). These blocks are progressively less compressible (the rank r grows with the electrical size). This affects the storage requirements, obviously, and also the computation time, since the cost of the ACA algorithm scales with r^2 .

The SPACA method proposed here aims to further compress large and (relatively) high rank ACA blocks, and to do this efficiently. To start with the efficiency, inspired by the MDA method [3] we approximate the elementary scatterers (basis functions) of the two domains (labeled D_1 and D_2 , see Fig. 1) that generate the block in question, by a reduced set of basis functions. But the MDA introduces ‘equivalent sources’

Fig. 1. The SPACA algorithm approximates the interaction between two subdomains D_1 and D_2 making use of a subsampling (the dark squares, and a secondary subdivision into subdomains d_q and d_p (along the dashed lines).

Fig. 2. Number of singular values larger than 10^{-3} times the largest singular value of the coupling matrix between two square conducting plates, versus electrical radius.

and our aim was to avoid the use of any ‘external information’ apart from the problem geometry. Therefore we choose to use a subset of the original basis functions (samples) in the two domains.

This requires an a priori estimate of the minimum necessary number of samples s . Theoretically, for domains that are asymptotically large with respect to the wave length, the number of DoF scales quadratically with the electrical radius kR of the domains (R is defined as the maximum element distance from the geometric center of the domain), provided that the mutual distance between the two domains is proportional to their radius. The latter condition is globally fulfilled due to the hierarchical subdivision. Therefore, according to the asymptotic theory, and requiring that the number of samples be proportional to the number of DoF for all domain sizes, we should set $s \propto (kR)^2$.

However, in [9] we found experimentally that in practise the scaling factor is unpredictable, but much lower than $(kR)^2$ even for domains with a radius up to hundreds of wave lengths. Since the sample density determines to a large extent the efficiency of the SPACA, it is worthwhile to determine a tighter

criterion. To this aim we executed an experiment with two perfectly conducting square plates of $1m \times 1m$ facing each other at a distance of $3m$. We used the ACA with threshold 10^{-4} and SVD with threshold 10^{-3} to determine the number of significant singular values of the coupling matrix between the plates as a function of the wave length. The result is shown in Fig. 2. The figure also shows a straight fit for the lower part of the curve as an illustration. Although the precise number of DoF will depend on the specific distribution of the basis functions within the domains, the scaling with the frequency is a function of the electrical radius only. Therefore the slope observed in this example is representative for the general case. One observes that up to a radius of about 12 wave lengths a linear fit yields a qualitatively good estimate of the number of DoF. Based on this result we use the following formula for the number of samples:

$$s = \text{int}\left(K \frac{R}{\lambda}\right), \quad (2)$$

where the function $\text{int}(x)$ returns the largest integer smaller than or equal to x and K is an adjustable accuracy parameter. Eq (2) is limited to problems for which the largest sub domains interacting through SPACA have an electrical radius no larger than about 12λ .

In order to find the actual samples we first find a set of $s/2$ basis functions distributed as evenly as possible over the domain. Then we pick a second set of $s/2$ basis functions close to the first set but with a different orientation to capture two independent polarizations in the far field. Using these samples directly to compress the block in question with the MDA algorithm would not do any better than ACA. MDA is a faster algorithm than ACA, but ACA picks its samples more ‘intelligently’, so it needs less of them for the same accuracy. In terms of compression rate we would gain nothing either.

Instead, we proceed as follows. First, just like in MDA we compute the interaction $Z_{s2,s1}$ between the samples of the two domains. The next step is to subdivide the two domains into smaller subdomains. In our implementation this is already done; the smaller subdomains are simply the lowest-level domains of the binary tree. Now we compute the interactions between the samples of D_1 and all the basis functions in the subdomains d_q , $q = 1 \dots P$, of D_2 , using ACA and SVD(τ), as

$$Z_{q,s1} \approx U_{q,s1} S_{q,s1} V_{q,s1}^T, \quad q = 1 \dots P, \quad (3)$$

as well as those between the samples in D_2 and the subdomains d_p , $p = 1 \dots P$, of D_1 , as

$$Z_{s2,p} \approx U_{s2,p} S_{s2,p} V_{s2,p}^T, \quad p = 1 \dots P. \quad (4)$$

These interactions are of much lower rank than the ACA of the entire block, since one side represents a much smaller domain [3]. According to the MDA we can now write the interaction between subdomain d_q in D_2 and subdomain d_p in D_1 as:

$$\begin{aligned} Z_{qp} &\approx Z_{q,s1} Z_{s2,s1}^{-1} Z_{s2,p} \\ &\approx U_{q,s1} (S_{q,s1} V_{q,s1}^T Z_{s2,s1}^{-1} U_{s2,p} S_{s2,p}) V_{s2,p}^T \\ &= U'_{q,s1} T_{qp} V_{s2,p}^T, \end{aligned} \quad (5)$$

where the dimensions of the inner matrix T_{qp} correspond to the ranks of $Z_{q,s1}$ and $Z_{s2,p}$, which are very small. Now we can combine the Z_{qp} for all the subgroups q and p into one sparse sequence approximating the interaction between D_1 and D_2 as follows:

$$Z_{D_2 D_1} \approx \begin{pmatrix} U_{1,s1} & 0 & \cdots \\ 0 & U_{2,s1} & \\ \vdots & & \ddots \end{pmatrix} \times \begin{pmatrix} T_{11} & T_{21} & \cdots \\ T_{12} & T_{22} & \\ \vdots & & \ddots \end{pmatrix} \times \begin{pmatrix} V_{s2,1}^T & 0 & \cdots \\ 0 & V_{s2,2}^T & \\ \vdots & & \ddots \end{pmatrix}. \quad (6)$$

For the ACA-SVD, the dominant matrices in terms of size are always the outer matrices U and V , which have dimensions $m \times r$ and $r \times n$ respectively where m, n are the block dimensions and r is the rank. In (6) the number of non-zero elements is reduced to $m \times \bar{\sigma}$ and $\bar{\sigma} \times n$ where $\bar{\sigma}$ is the average rank of the T_{qp} . Although the dimensions of the T_{qp} are very small, the inner matrix in (6) is generally larger than necessary: its rank is r (it should be, if the approximations are accurate enough), but its dimensions are higher, reflecting a redundancy of information when combining the subdomains. Therefore we apply a final SVD(τ) to it in order to maximize the compression. In consequence, our block $Z_{D_2 D_1}$ is represented by a sequence of five matrices. In our implementation we apply the above procedure to non-touching blocks at all the levels except the finest level. As mentioned before, the subdomains d_q and d_p that we use are always the domains of the finest level. Due to the binary tree structure this leads to two subdomains when D_1 and D_2 are at the second-finest level, then four, etcetera.

The SPACA method has been implemented and applied to the same test case as in section II-A in order to determine an adequate value for the parameter K in (2). This value should be as low as possible for efficiency without compromising the accuracy. Table III shows the evolution of the relative error in the surface current as a function of K . Since the SPACA algorithm also uses ACA compression, the error has two components: the error due to the SPACA sub-sampling and the error due to ACA compression. In the table we also show the error due to the SPACA sub-sampling only. This latter value was computed by repeating the computation with the ACA threshold set to 10^{-6} , which amounts to eliminating the error due to ACA. The value of $K = 40$ was selected because the SPACA sub-sampling error is well below the error already present due to the ACA compression, resulting in a relative error less than twice that of the ACA method.

C. Complexity of SPACA

At this point we should stress that, unlike the ML-ACA, the SPACA method does not in fact directly address the problem

TABLE III
COMPARISON OF SPACA COMPUTATIONS WITH DIFFERENT VALUES OF THE PARAMETER K IN (2). TEST OBJECT: NASA ALMOND, 38706 UNKNOWN. ACA THRESHOLD 10^{-4} , SVD THRESHOLD 10^{-3} . THE RELATIVE ERROR DUE TO THE SPACA SAMPLING ONLY WAS COMPUTED WHILE MINIMIZING THE ACA COMPRESSION ERROR BY USING A VERY LOW THRESHOLD.

K (Eq. 2)	rel. error	rel. error due to SPACA sampling only
20	8.4×10^{-3}	9.1×10^{-3}
30	3.7×10^{-3}	2.7×10^{-3}
40	2.4×10^{-3}	0.7×10^{-3}
no SPACA	1.5×10^{-3}	—

of growing DoF with the electrical size of the domains, since the interaction $Z_{s2,s1}$ between the two entire domains is still computed directly. Rather, it isolates this problem and reduces its impact. As the numerical example in section III illustrates, this impact, measured as the relative size and computation time of the inner matrix in (6), becomes relatively insignificant even for problems of more than a million unknowns and larger than 50 wavelengths.

Because the size (number of non-zero elements) of the outer matrices in (6) scales linearly with their dimensions, the complexity of the SPACA (asymptotical scaling of matrix size and compressed matrix setup time with the total number of unknowns N of the problem) with respect to these outer matrices is $N \log N$. However, for very large problems, the size of $Z_{s2,s1}$ and associated computation time will become significant. When this happens, ML-ACA may be invoked to recompress $Z_{s2,s1}$ and restore the $N \log N$ complexity.

III. NUMERICAL EXAMPLES

A. NASA Almond

In order to assess the performance of the SPACA algorithm in function of the problem size, we compared it to the ordinary ACA algorithm for the NASA almond at a range of frequencies, always adapting the discretization to maintain a ratio of 600 unknowns per square wave length on average. The formulation was the same as in section II-A1. The computations were done on a PC with 64 GB of RAM and a Dual Intel[®] Xeon[®] X5482 processor at 3.20GHz, in single precision. The code is written in MATLAB[®] 7.9.0. The parameters of the computations are given in Table IV. The matrix setup times and the compressed matrix sizes are shown graphically in Figs. 3 and 4. In Fig. 5 and 6, the bistatic RCS computed with ordinary ACA and with SPACA are compared at 62.6 GHz for VV polarization and incidence angle towards the almond tip.

From the figures we can observe that both the setup time and the compressed matrix size scale not far from $N \log N$ over the range investigated. For larger problems, the curve will eventually bend upwards, as both the computation time and storage for the inner matrices T_{qp} become significant. Furthermore, the largest subdomains for the largest almond

had an electrical radius of 11.5λ . For still larger problems, more samples than predicted by (2) are necessary to capture the interaction between large subdomains (see section II-B). Invoking the MLACA algorithm as suggested in section II-C should restore the $N\log N$ scaling, though at the cost of a larger constant factor.

The ACA curves are considerably steeper. In [8] a complexity of $N^{4/3}\log N$ is suggested, based on numerical experiments. The ACA curves are indeed close to this value. However, we believe that in general for ACA the scaling with N cannot be predicted reliably for practical problems, only an upper bound, valid for $k \rightarrow \infty$ [9]. This upper bound was derived to be $N^{3/2}\log N$ in [13].

We analysed the matrices and computation times for the largest case of 62.6 GHz , in relation to section II-C. The NASA Almond has a length of approximately 25cm , which is about 50λ at this frequency. Of the total compressed matrices, the self- and touching blocks at the finest level make up 13 GB. This part is the same for the ACA and the SPACA implementation. The rest takes up 40.8 GB with ordinary ACA and 17.4 GB with SPACA. Out of these 17.4 GB, 1.1 GB is occupied by the inner matrices from (6), about 6%. In terms of computation time: the common part took 4,364 seconds to compute. The rest took 29,094 seconds with ACA and 11,310 seconds with SPACA. Out of these 11,310 seconds, about 850 seconds were needed for the inner blocks, about 7.5%. Clearly, for a problem of this size, the resources needed for the SPACA inner matrices are relatively insignificant.

Finally, in Fig. 7 we show the average rank of the matrix blocks for the entire problem (again at 62.6 GHz) at the different levels, starting with the second finest level (about 1,000 basis functions per block) and going up to the largest level at which non-touching interactions exist, with blocks of about 250,000 basis functions. The ACA curve shows the rank as determined by ACA-SVD, the SPACA curve shows the average rank of the diagonal sub-blocks of the outer matrices of (6), which equals the total number of non-zero elements per row of the outer matrices. The latter is practically constant, as expected. Also shown is the average number of samples prescribed by (2) at each level. We observe that the number of samples prescribed by (2) is reasonably proportional to the rank as determined by ACA-SVD over the represented range of block sizes. Since the DoF are entirely determined by the oscillatory kernel of the integral equation, we can safely generalize equation (2) to any geometry and also to different integral equation formulations. Of course, if a different accuracy is required, both the ACA threshold and the constant in (2) must be adapted accordingly.

B. Parabolic reflector

As a second example we calculated the directivity of a parabolic reflector using ordinary ACA and using the SPACA algorithm. The reflector diameter is $3m$, its focal distance is $1.535m$, and it is fed at 3.3GHz by an infinitesimal dipole in the focal point, backed by a $0.22m$ circular disk at 0.25λ . Three supporting struts with a circular cross section, diameter

TABLE IV
COMPUTATION PARAMETERS FOR SPACA/ACA-GMRES SOLUTION OF NASA ALMOND VERSUS FREQUENCY. N IS THE NUMBER OF BASIS FUNCTIONS. L IS THE NUMBER OF LEVELS IN THE HIERARCHICAL SUBDIVISION. K IS THE ACCURACY PARAMETER FROM EQ. (2). MVP MEANS MATRIX VECTOR PRODUCT.

freq. (GHz)	12.1	19.4	40.7	62.6
N	38,706	99,915	442,089	1,043,577
L	7	8	10	11
K	40			
MVP _{ACA} (sec)	4	11	53	124
MVP _{SPACA} (sec)	3	10	48	106

Fig. 3. Compressed matrix build time for NASA Almond, using ACA and SPACA, versus number of unknowns.

$0.023m$ are included in the model, see the inset of Fig. 8. All surfaces are perfect conductors. This constitutes an open geometry, so we had to use the EFIE integral equation formulation instead of the CFIE. The problem was discretized using $N = 403,170$ RWG basis functions. The parameters for the ACA and SPACA algorithms were identical to those for the NASA almonds: ACA threshold $\tau = 10^{-4}$, $L = 10$ binary tree levels, $K = 40$ in equation (2) for the number of SPACA samples. Although it is outside the scope of this paper we should mention that the EFIE impedance matrix for this problem was very poorly conditioned (a well known problem of the EFIE formulation) and we had to use an ILU preconditioner [15] to speed up the convergence of the iterative solver. Constructing the preconditioner took approximately 40 minutes.

Fig. 8 shows the directivity of the antenna as calculated by ACA and SPACA. Table V compares the performance of ACA and SPACA. Although SPACA again yields a considerably smaller compressed matrix, the construction time is actually slightly larger with SPACA. The reason for this is the follow-

Fig. 4. Compressed matrix size for NASA Almond, using ACA and SPACA, versus number of unknowns.

Fig. 6. Bistatic RCS of NASA Almond at 62.6 GHz, H-plane, computed with ACA/SPACA, 1,043,577 RWG basis functions.

Fig. 5. Bistatic RCS of NASA Almond at 62.6 GHz, E-plane, computed with ACA/SPACA, 1,043,577 RWG basis functions.

Fig. 7. Average rank versus block-size for ACA and SPACA compressed blocks, NASA Almond, 1,043,577 unknowns. Also shown number of samples according to Eq. (2) with accuracy parameter $K = 40$.

ing: Considering the matrix sizes for the second-largest NASA almond of sec III-A, which has a number of unknowns close to that of the reflector, one notes that the almond compressed matrix is more than twice as large. This is because the bulk of the reflector basis functions are situated on the reflector surface, which is relatively flat, making for many nearly coplanar sub-domains. Since the number of DoF between sub-domains is related to the solid angle subtended by each sub-domain as seen from the other, the matrix blocks for these domains are highly compressible. Ordinary ACA compression will be proportionally faster, but the SPACA with accuracy parameter $K = 40$ is over-sampling these blocks, which are later recompressed at a higher cost. The obvious solution

would be to lower the value of K . However, we tried this and it turns out that it considerably affects the result, in particular in the $\theta = 90^\circ$ region, where the dominant contribution of the radiated field originates from the reflector- and disk-edges and the struts. This suggests that an important additional gain in efficiency of SPACA can be achieved by devising a more sophisticated adaptive sampling scheme taking into account the relative orientation of sub-domains. We hope to report on this in a future paper.

TABLE V
COMPUTATION PARAMETERS FOR SPACA/ACA-GMRES SOLUTION OF
PARABOLIC REFLECTOR DIRECTIVITY. MVP MEANS MATRIX VECTOR
PRODUCT.

	ACA	SPACA
matrix build time (sec)	4,181	4,394
matrix size (MB)	7,407	5,024
MVP (sec)	34	32

Fig. 8. Directivity of parabolic reflector antenna, computed with ACA and SPACA algorithm, $N = 403, 170$. The inset shows the antenna geometry.

IV. CONCLUSION

The Sparsified Adaptive Cross Approximation (SPACA) method for accelerated solution of electrically large radiation and scattering problems has been presented. SPACA is a modification of the ordinary ACA method. It produces an additional error, but this error is controllable through an accuracy parameter. As with ordinary ACA, knowledge of the topology of the basis functions that model the problem is necessary in order to construct a binary tree sub-division, and for choosing sample basis functions in the case of SPACA, but no further problem-specific information is necessary. Numerical experiments have been presented showing a considerable gain in efficiency of SPACA over ACA, especially for electrically large problems. The largest target analysed for this paper was the NASA almond of about 50 wave lengths long, using approximately 1,000,000 basis functions. Up to this electrical size, the method showed a computational complexity close to $N \log N$. For even larger problems, this very favourable proportionality will inevitably be lost with the method as presented in this paper. However, a multi-level implementation, to be reported on in a future paper, is expected to yield a truly asymptotic $N \log N$ complexity. A second issue on which research is ongoing, possibly leading to further improvements of the method, is that of finding an optimal distribution of the sample basis functions.

ACKNOWLEDGEMENTS

This work was supported by the Spanish Interministerial Commission on Science and Technology (CICYT) under projects TEC2009-13897-C03-01, TEC2010-20841-C04-02 and CONSOLIDER CSD2008-00068 and through the Ramon y Cajal program.

REFERENCES

- [1] E. Bleszynski, M. Bleszynski and T. Jaroszewicz, "AIM: Adaptive Integral Method for Solving Large-scale Electromagnetic scattering and Radiation Problems", *Radio Science*, vol.31, no.5, pp1225-1251, 1996, Sep.-Oct.
- [2] J. Song, C.C. Lu and W.C. Chew, "Multilevel Fast Multipole Algorithm for Electromagnetic Scattering by Large Complex Objects", *IEEE Trans. on Antennas and Propag.*, Vol. 45, No. 10, pp. 1488-1493, October 1997.
- [3] E. Michielssen and A. Boag, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures", *IEEE Trans. on Antennas and Propag.*, Vol. 44, No. 8, 1086-1093, Aug. 1996.
- [4] J.M. Rius, J. Parrón, E. Úbeda, J.R. Mosig, "Multilevel Matrix Decomposition Algorithm for Analysis of Electrically Large Electromagnetic Problems in 3-D", *Microwave and Optical Technology Letters*, Vol. 22, No. 3, pp. 177-182, 5th August 1999.
- [5] J. M. Rius, J. Parrón, A. Heldring, J. M. Tamayo, E. Ubeda, "Fast iterative Solution of Integral Equations with Method of Moments and Matrix Decomposition Algorithm - Singular Value Decomposition," *IEEE Trans. on Antennas and Propag.*, special issue on "Large and Multiscale Computational Electromagnetics", Vol. 56, No. 8, pp. 2314-2324, August 2008.
- [6] L. Banjai and W. Hackbush, "Hierarchical matrix techniques for lowandhigh-frequency Helmholtz problems," *IMA J. Numerical Anal.*, Vol. 28, pp. 46-79, 2008.
- [7] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Mathematik*, Vol. 86, No. 4, pp. 565-589, 2000
- [8] K. Zhao, M. N. Vouvakis and J.-F. Lee, "The Adaptive Cross Approximation Algorithm for Accelerated Method of Moments Computations of EMC Problems," *IEEE Trans. on Antennas and Propag.*, Vol. 47, No. 4, pp. 763-773, Nov 2005
- [9] A. Heldring, J. M. Tamayo and J. M. Rius, "On the degrees of freedom in the interaction between sets of elementary scatterers," 3rd European Conference on Antennas and Propagation, Eucap 2009, Berlin, 23-27 March 2009.
- [10] J. M. Tamayo, A. Heldring, and J. M. Rius, "Multilevel Adaptive Cross Approximation (MLACA)," *IEEE Trans. on Antennas and Propag.*, Vol. 59, No. 12, pp. 4600-4608, Dec 2011
- [11] A. C. Woo, H. T. G. Wang, M. J. Schuh and M. L. Sanders, "Benchmark Radar Targets for the Validation of Computational Electromagnetics Programs," *IEEE Antennas and Propagation Magazine*, Vol. 35, No. 1 pp. 84-89, February 1993.
- [12] S.M. Rao, D. R. Wilton and A. W. Glisson, "Electromagnetic Scattering by Surfaces of Arbitrary Shape", *IEEE Trans. on Antennas and Propag.*, Vol 30, No 3, pp. 409-418, May 1982.
- [13] A. Heldring, J. M. Rius, J. M. Tamayo, "Comments on Fast Direct Solution of Method of Moments Linear System," *IEEE Trans. on Antennas and Propag.*, Vol 58, No 3, pp. 1015-1016, Mar 2010
- [14] Y. Saad and M. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. on Scientific Computing*, Vol. 7, No. 3, 856-869, 1986
- [15] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996