

Multilevel Adaptive Cross Approximation (MLACA)

J.M. Tamayo*, A. Heldring, and J.M. Rius

Dept. of Signal Theory & Comm, Polytechnic University of Catalonia,
c/ J. Girona 1, 08034, Barcelona, Spain

Introduction

In recent years, several methods have been developed for accelerating the iterative solution of the electromagnetic integral equations discretized by Method of Moments (MoM). Most of them are based on the compressibility of parts of the MoM matrix when they correspond to the interaction between two distant boxes. Among them, a promising one is the Adaptive Cross Approximation (ACA) [1] which combined with a SVD recompression achieves very good results, especially for moderately large number of unknowns N . In this paper we propose a multilevel implementation of the ACA leading to a highly enhanced compression rate.

Description of the Algorithm

Let us define the environment of the problem under study. Imagine we have two objects contained within two spheres that do not intersect each other. We shall call them source and observation objects respectively. Applying the Method of Moments (MoM) it is possible to express the interaction between them with a matrix Z_{mn} with dimensions $m \times n$, where m and n are the number of basis functions in which the observation and source object have been discretized, respectively. In the first place, we must subdivide each object recursively into smaller domains in a binary tree manner, so that in each subdivision approximately half of the basis functions goes to each side. The number of levels is L meaning that at the finest level we have in each subset approximately $M = N/2^L$ elements considering $N = n = m$. For asymptotical analysis L is going to be chosen to yield a fixed value of M or equivalently, a fixed minimum box electrical size.

A scheme of the algorithm is shown in figure 1. The idea is to express the initial matrix Z_{mn} as a product of $L + 2$ matrices $Z_{mn} = A^{(L+1)} B^{(L+1)} B^{(L)} \dots B^{(1)}$.

In the step 0 (figure 1(a)) Z_{mn} is transformed into two new matrices $A_0^{(1)}$ and $B_0^{(1)}$. The procedure to obtain them is to split Z_{mn} into strips, corresponding to the interaction of each subset of basis functions at the finest level in the source object with the whole observation object. Now each of those strips only has k degrees of freedom, therefore can be compressed with the ACA-SVD algorithm and regrouped as is shown in figure 1(a). Note that the matrix to store is $B_0^{(1)}$ which has 2^L blocks in the diagonal of size $k \times \frac{n}{2^L}$ and is orthogonal.

For each $i = 1, \dots, L$ and each $j = 0, \dots, 2^{i-1} - 1$ the matrices from the step $i - 1$, $A_j^{(i)}$, are transformed into four new matrices $A_{2j}^{(i+1)}$, $A_{2j+1}^{(i+1)}$, $B_{2j}^{(i+1)}$, $B_{2j+1}^{(i+1)}$ (figure 1(b)). The

⁰This work was supported by the Spanish Interministerial Commission on Science and Technology (CI-CYT) under projects TEC2006-13248-C04-02/TCM and TEC2007-66698-C04-01/TCM and CONSOLIDER CSD2008-00068 and by the "Ministerio de Educación y Ciencia" through the FPU fellowship program.

initial matrix $A_j^{(i)}$ is split into two sets of rows corresponding to the next subdivision in the observation object tree. On the other hand, the strips are grouped in pairs corresponding to the previous level in the source object tree. As the source is doubled in size and the observation is divided by two, the number of degrees of freedom of the new substrips is again k [3], and therefore can be recompressed using the ACA-SVD algorithm to obtain the new matrices. After the step $i = L$ we have the whole set of matrices which correctly combined represent a compressed version of the matrix Z_{mn} .

It can be proved that the memory required to store those matrices at the end of the algorithm is proportional to $kN \log N$ and as the size of the finest level is constant with respect to N , we have that k is constant, and therefore the memory scales with $O(N \log N)$. Consequently, the computational cost of a product of the matrix Z_{mn} with a vector scales also with $O(N \log N)$. The computational cost to build the compressed matrix can be proved to be $O(N^2)$, which is asymptotically better than for single-level ACA [2].

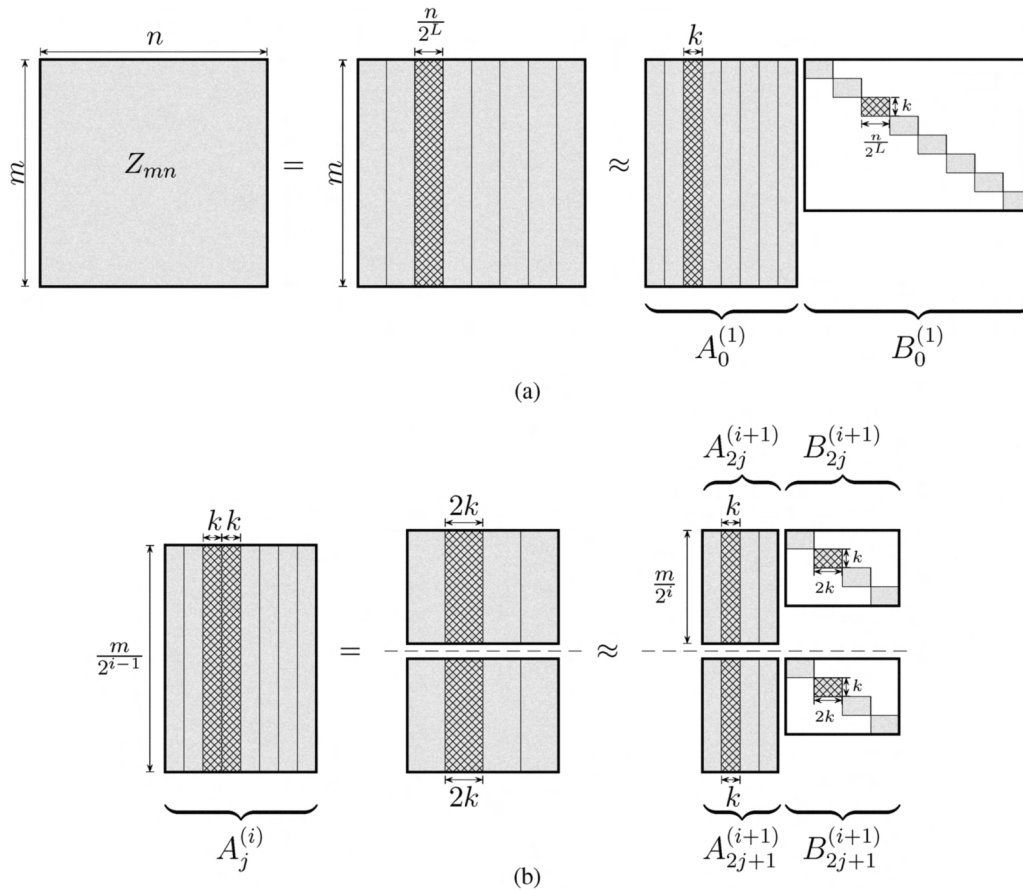


Figure 1: Graphical representation of the matrix transformations in the MLACA algorithm described in section “Description of the algorithm”. First step or step 0 (a) and step i (b) with $i \geq 1$.

Numerical Aspects

Although the resulting matrices from the algorithm described in the previous section have a number of elements proportional to $N \log N$, if it is implemented naively it will temporarily need an amount of memory growing with $O(N^2)$. This can be avoided by changing the order of the operations, achieving again order $O(N \log N)$ during the whole execution. The way to proceed is to go back recursively from the last step and computing at each moment only what is necessary for each block. The computational complexity is maintained as the operations are the same, only reordered.

A theoretical analysis of the error has been performed. As the matrix is decomposed as a product of different matrices, to assure a relative error in the matrix smaller than τ it is necessary that each matrix has an error smaller than $\tau/(L+2)$ as we have $L+2$ matrices. This is because in the worst case the relative errors are added. This change in the threshold τ in each ACA simply affects the value of degrees of freedom k which now can be proved to grow with $O(\log(\log N))$. Therefore, the real complexities of the MLACA algorithm are slightly greater, although a factor $\log(\log N)$ is virtually negligible in practice.

Numerical Experiments

Two experiments have been performed in order to corroborate the computational time and the memory requirements of the MLACA. Both experiments concern PEC objects discretized with RWG basis functions. The first simulation corresponds to the interaction between two square plates facing each other with size $1m \times 1m$. They are separated $3m$ and therefore they have a reduced number of degrees of freedom. To observe the orders, different frequencies have been applied maintaining the average electric size of the RWG basis functions constant at $\lambda/10$. The reference thresholds are 10^{-4} for the ACA and 10^{-3} for the SVD recompression. The results are shown in figure 2(a)-(b). In figure 2(a) it can be seen how the different simulations with the MLACA follow precisely the reference curves proportional to $N \log N$, whereas the ACA starts to grow quicker when N grows because the number of DoF grows. Another important observation is that the results are not very sensitive to the sizes of the finest level boxes, or the number of unknowns in each box M . This means that any reasonable M value will yield a good compression. Figure 2(b) represents the CPU time for the above simulations. Again it behaves as expected theoretically, with the computation time growing as $O(N^2)$. In this respect it is better to have a moderately large value of M . Although the ACA is expected to behave worse than the MLACA in the limit, it can not be observed in the figure because N is still relatively small.

The second experiment is very similar to the first one. The only difference is that now we have spheres with radius $0.5m$ instead of square plates and the distance between their centers is $2m$. The same conclusions can be extracted from the results (figure 2(c)-(d)).

Conclusions

A novel algorithm based on the Adaptive Cross Approximation (ACA) [1] has been developed. It has been proved, both theoretically and numerically, to have a computational complexity scaling with N^2 and memory requirement growing with $N \log N$.

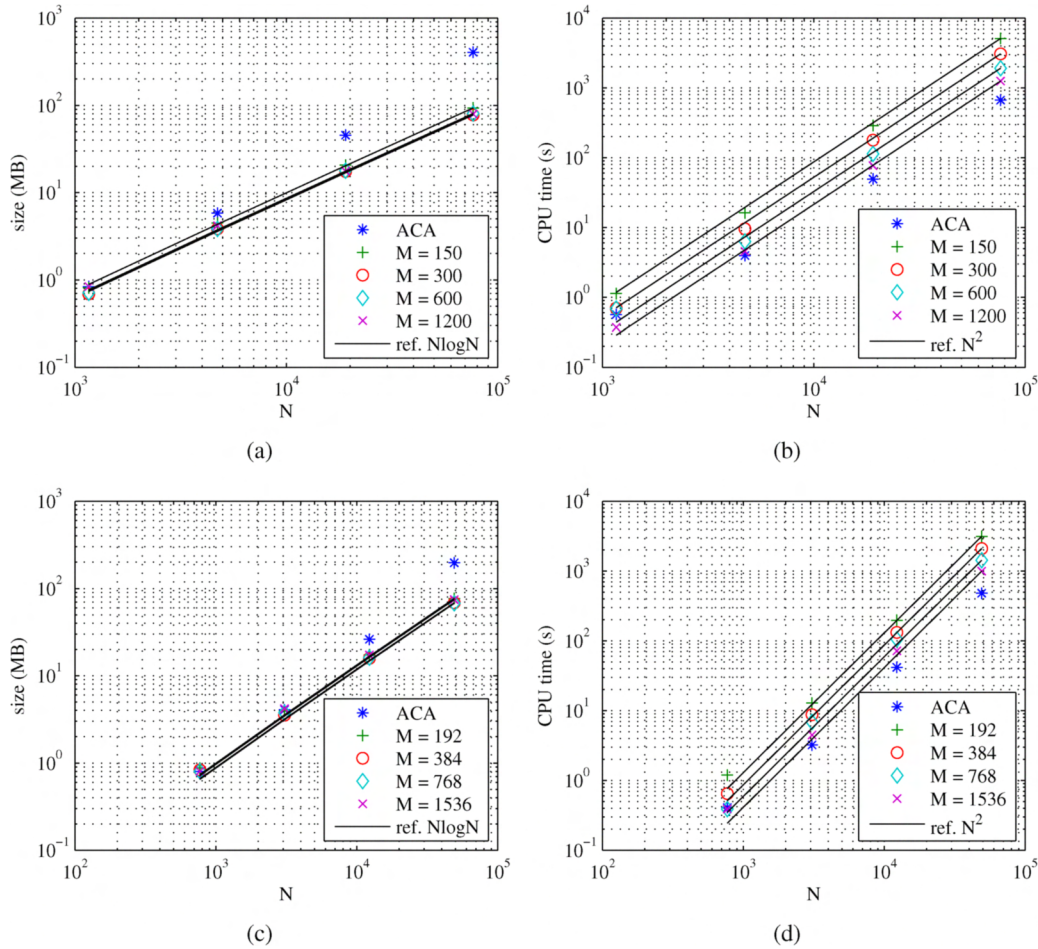


Figure 2: Size of the compressed matrix (a) and CPU time (b) as a function of the number of unknowns N when applying the ACA and the MLACA algorithms to the MoM matrix corresponding to the interaction between two opposite square plates $1m \times 1m$ separated $3m$. M is the approximate number of unknowns in each box at the finest level of the MLACA algorithm. The same results (c) and (d) are shown for two spheres with radius $0.5m$ and distance $2m$.

References

- [1] M. Bebendorf, "Approximation of boundary element matrices," *Numer. Math.* (2000) 86: 565–589
- [2] K. Zhao, M.N. Vouvakis and J.F. Lee, "The Adaptive Cross Approximation Algorithm for Accelerated MoM Computations of EMC problems," *Trans. on Elec. Comp.*, Vol. 47, No.4, pp. 763-773, November 2005
- [3] E. Michielsen, and A. Boag, "A Multilevel Matrix Decomposition Algorithm for Analyzing Scattering from Large Structures," *IEEE Trans. on Antennas and Propagation*, Vol. 44, No.8, 1086-1093, Aug. 1996