Adapted ACA Algorithm with Improved Efficiency and Compression Rate

Alex Heldring, Eduard Ubeda, and Juan M. Rius, *Member, IEEE*

***Abstract*—This contribution proposes an adapted version of the popular Adaptive Cross Approximation algorithm for block wise compression of Method of Moments impedance matrices. The estimation of the matrix block Frobenius norm, necessary to know the relative error in the compressed representation takes up a substantial percentage of the total computation time. In the adapted version it is replaced by a stochastic process with negligible cost. The involved uncertainly is eliminated using an a posteriori SVD recompression which, as a fortunate side-effect, yields an important additional reduction of the compressed rank.**

***Index Terms*—ACA, Fast methods, MoM, Integral Equations.**

I. Introduction

THE Adaptive Cross Approximation (ACA) algorithm [1] is a popular tool for compressing off-diagonal blocks of a Method of Moments impedance matrix. It approximates an $m\times n$-element matrix in k steps, each step consisting of the selection and inclusion in the compressed decomposition of a row and a column. Typically, the approximation error decreases fast with k and convergence to within a chosen relative error is achieved in $k << n,m$ steps. The computational cost of the algorithm is proportional to $k^2(n+m)$ and the storage to $k(n+m)$, both generally much smaller than computing and storing the full block.

A key element of the ACA is the stopping criterion that decides with every step whether convergence is reached. The criterion estimates the relative error in the Frobenius norm of the compressed decomposition and compares it with a chosen threshold value. This requires estimating the Frobenius norm of the full block, which is done iteratively in a process that scales with $k^2(n+m)$, equal to the decomposition itself. The computational cost of the Frobenius norm estimation is typically not quite half of the total cost because the decomposition also involves evaluating the elements of k rows and columns. Depending on the problem kernel and the chosen integration scheme, this latter operation can be made highly efficient in which case the cost of the Frobenius norm estimation approaches one half of the total cost.

In a recent paper [2] we proposed an alternative stochastic approach to estimate the Frobenius norm, which is considerably faster than the iterative procedure above. It does however by its nature introduce an element of probability, which for some applications may be unacceptable. In this paper we propose a combination of the stochastic approach of [2] with a final reorthogonalization of the compressed decomposition, which is demonstrated to be considerably faster than the conventional ACA algorithm (though not as fast as the purely stochastic approach from [2]), while entirely eliminating the influence of chance. Furthermore, at negligible extra cost the proposed procedure allows to convert the ACA decomposition into an equivalent SVD, with a substantially improved compression rate. The reorthogonalization procedure uses QR decompositions, which represent an additional $k^2(n+m)$ operation, but since they are done at once rather than incrementally inside the ACA steps, they can be performed by optimized linear algebra routines, whence the speedup.

II. The algorithm

The proposed algorithm is largely identical to the conventional ACA algorithm, as given in e.g. [3]. The first adaptation is to remove the line that estimates the Frobenius norm (line 6 in [3]) and instead to pre-compute an estimate of the Frobenius norm according to the procedure proposed in [2]. This consists in randomly picking individual sample elements from the matrix block to be compressed, one by one, until the following relation holds:

$$\frac{t(\frac{\alpha}{2},N-1)\sigma_s}{\sqrt{N}\mu_s} < \Delta_{tol}, \tag{1}$$

where N is the number of samples, $t(\alpha/2,N-1)$ is the student-t distribution parameter for probability value α, and σ_s and μ_s are the sample standard deviation and mean, respectively.
Δ_{tol} is the chosen tolerance; with (1) fulfilled, the relative error in the Frobenius norm will exceed Δ_{tol} with probability α. As argued in [2], $\Delta_{tol}=0.5$ is an acceptable tolerance since the uncertainty in the rest of the ACA algorithm is of the same order of magnitude. In [2], Δ_{tol} and α were chosen to virtually

Manuscript received February 25, 2015. This work was supported by the Spanish Ministry of Economy and Competitiveness under projects TEC2012-37582-C04-02, TEC2010-20841-C04-02 and by the European FEDER fund.

All three authors are with Universitat Politècnica de Catalunya, c/Jordi Girona 1, 08034 Barcelona, Spain (phone: 934017216; fax: 934017232; e-mail: heldring@ tsc.upc.edu).

exclude the possibility of an unacceptably high relative error, reducing it to less than one in a billion. The algorithm proposed here sets Δ_{tol} to 0.5 and α to 0.1%. Consequently, one in every 1.000 computations, the estimated norm will be unacceptably high.

The second adaptation concerns the line where the convergence criterion is evaluated (line 7 in [3]). Rather than exiting when convergence is detected, the current ACA matrices *U* and *V* are SVD-recompressed as in e.g. [4].

Subsequently, obtaining the Frobenius norm from the compressed SVD representation is immediate. If it turns out that the stochastically estimated norm was unacceptably large, the ACA iterations are resumed until convergence is reached based on the new, accurate norm, followed by a final SVD-recompression. The above 'restart' is expensive, but far less so than a full ACA compression. Since it will only take place once every 1.000 executions on average, it represents a negligible computational overhead.

The stochastically estimated norm can of course be too small as well. In that case, convergence will be detected 'too late' and the compressed block will be 'too accurate'. This has no consequence other than another small computational overhead due to a few unnecessary extra ACA iterations.

III. Numerical Example

We illustrate the proposed algorithm on the same test case that we used in [2], the mutual interaction of two *5m×5m* square plates, facing each other broad side, at a mutual distance of *10m*, at a frequency of *300 Mhz*. The plates are each modeled with *7400* RWG basis functions. Note: The presented simulations were executed on a PC with an Intel Core i7-2630QM CPU at *2.00 GHz* and *8 GB* of RAM. The timings are different from [2] because a different computer was used.

Table I compares the computation times and compression rates for conventional ACA with and without SVD recompression and for the proposed algorithm. The ACA convergence threshold was set to 10^{-3}. The stochastic norm estimation took less than *0.01 sec*.

Table I: Computation times and compressed rank of ACA compression of a *7400×7400* element interaction matrix of two PEC square plates.

	Conventional ACA	Conventional ACA+SVD	Proposed algorithm
UV-build (sec)	*3.0*	*3.0*	*3.0*
Norm estim. (sec)	*1.0*	*1.0*	-
SVD-recompr. (sec)	-	*0.4*	*0.4*
Total time (sec)	*4.0*	*4.4*	*3.4*
Compressed rank	*92*	*74*	*74*

In Table I we observe that the proposed algorithm is faster than the conventional ACA even without SVD recompression. Since it yields a 20% gain in compression rate, one may argue that a fair comparison should include the SVD compression, in which case the obtained speed-up is about 22% in the example.

The timing of the proposed algorithm as reported in Table I is in fact the average over 10.000 realizations; the histogram of the total computation time is shown in Fig. 1. In fact, although theoretically the number of unacceptable norm estimations should be close to 10, in this case it happened only once. The reason is that for this case, the distribution of the absolute values of the matrix elements is so flat, that most of the time, the initial number of 11 samples was amply sufficient to fulfill (1) so we are 'oversampling' from the start. As for the number of ACA iterations used by the algorithm, they ranged between 90 and 99, with an average value of 92.8.

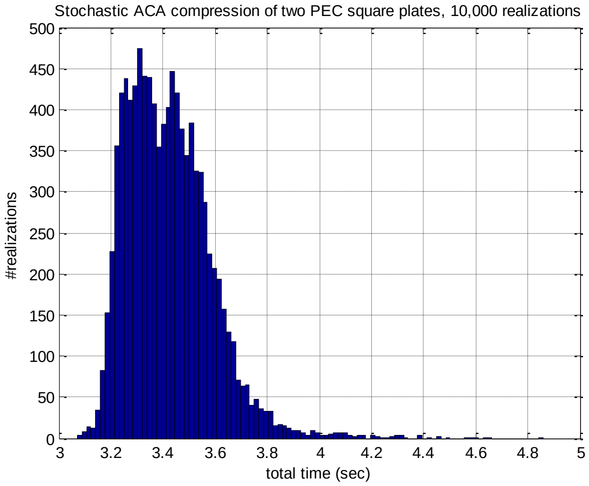

Fig. 1. Histogram of total computation times for the proposed method.

IV. Conclusion

An adapted ACA algorithm is proposed that replaces the conventional iterative Frobenius norm estimation with a stochastic estimation. The resulting uncertainty in the estimated value is eliminated using an a posteriori SVD recompression, which simultaneously yields a substantial additional compression. A numerical example is presented which shows a 23% speed-up with respect to the conventional ACA, but this percentage may be considerably higher, up to almost 50% with a highly optimized code.

References

[1] M. Bebendorf, "Approximation of boundary element matrices," *Numerische Matematik*, Vol. 86, No. 4, pp. 565-589, 2000.

[2] A. Heldring, E. Ubeda and J.M. Rius, "Stochastic Estimation of the Frobenius Norm in the ACA Convergence Criterion," *IEEE Trans. Ant. and Prop*. Vol. 63, No. 3, pp. 1-4, March 2015

[3] 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 Electromag. Compatibility*, Vol. 47, No. 4, pp. 763-773, Nov 2005

[4] J. M. Rius, J. Parrón, A. Heldring, J. M. Tamayo, and E. Ubeda, "Fast iterative solution of integral equations with method of moments and MDA-SVD," *IEEE Trans. Ant. and Prop.*, vol. 56, no. 8, pp. 2314–2324, Aug. 2008