Improved Multilevel Fast Multipole Method for Higher-Order Discretizations

Oscar Borries[1,2], Peter Meincke[2], Erik Jørgensen[2], Stig Busk Sørensen[2], Per Christian Hansen[1]
[1]Technical University of Denmark, DTU Compute, Kgs. Lyngby, Denmark, {opbo,pcha}@dtu.dk
[2]TICRA, Læderstræde 34, DK-1201 Copenhagen, Denmark, {ob,pme,ej,sbs}@ticra.com

***Abstract*—The Multilevel Fast Multipole Method (MLFMM) allows for a reduced computational complexity when solving electromagnetic scattering problems. Combining this with the reduced number of unknowns provided by Higher-Order discretizations has proven to be a difficult task, with the general conclusion being that going above 2^{nd} order is not worthwhile. In this paper, we challenge this conclusion, providing results that demonstrate the potential performance gains with Higher-Order MLFMM and showing some modifications to the traditional MLFMM that can benefit both Higher-Order and standard discretizations.**

***Index Terms*—Multilevel Fast Multipole Method, Computational Electromagnetics, Higher-Order Discretization, Electromagnetic Scattering**

I. Introduction

Discretization of the surface current density occuring in either the Electric Field Integral Equation (EFIE) or the Combined Field Integral Equation (CFIE) can be done using a wide variety of methods, most popularly the RWG [1] first-order basis functions on a triangular mesh. When discretizing the scattering problem using a Method of Moments (MoM) approach, the resulting matrix equation has N unknowns and thus requires $\mathcal{O}(N^2)$ memory and computational time to solve iteratively. Thus, a key issue in an efficient solution is the reduction of N relative to the target accuracy of the obtained surface current density. It has been demonstrated [2] that Higher-Order (HO) basis functions significantly reduces N as compared to a RWG discretization. Higher-Order discretizations achieve their efficiency by using smooth polynomials with large domains to represent the surface current density.

Another line of research has focused on reducing the complexity of the matrix vector product $\bar{\bar{Z}}\bar{I}$. One of the most popular approaches is the Multilevel Fast Multipole Method (MLFMM) [3], [4], which achieves $\mathcal{O}(N \log N)$.

However, combining the advantages of those two approaches has proved elusive. While several groups have made the attempt [5], [6], [7], each has independently arrived at the conclusion that basis functions above 2^{nd} order resulted in a memory increase and thus were not efficient for MLFMM use. We challenge that consensus with a carefully revised algorithm that is tailored towards the larger group sizes occuring in HO MLFMM, thereby allowing the reduced number of unknowns to result in a reduction in memory and, nearly as important, a significant reduction in computation time.

II. Multilevel Fast Multipole Method

MLFMM achieves reduced complexity by grouping the basis functions hierarchically, using the Octree algorithm [8], and letting larger and larger groups interact over greater and greater distances. The grouping is based on the center of the geometric elements of the mesh, and the smallest allowed groups have sidelengths not smaller than the largest geometric element in the mesh.

This splitting allows performing the matrix-vector product as

$$\bar{\bar{Z}}\bar{I} = \bar{\bar{Z}}_{\text{near}}\bar{I} + \mathcal{F}(\bar{I}) \tag{1}$$

where $\bar{\bar{Z}}_{\text{near}}$ is the near-matrix, containing the interactions between basis functions that are too closely spaced to apply MLFMM — the elements in this matrix are computed as in the normal MoM approach. $\mathcal{F}$ denotes the operation perfomed by applying MLFMM.

The interaction between two well-seperated basis functions $\boldsymbol{f}_j, \boldsymbol{f}_i$, belonging to groups m and m' respectively, can be computed by

$$\bar{\bar{Z}}_{j,i} = \kappa \oiint \boldsymbol{R}_{jm}(k,\hat{\boldsymbol{k}}) \cdot \Big(T_L(k,\hat{\boldsymbol{k}},\boldsymbol{r}_{mm'})\boldsymbol{V}_{im'}(k,\hat{\boldsymbol{k}})\Big)\, d^2\hat{\boldsymbol{k}}, \tag{2}$$

with $\boldsymbol{r}_{mm'} = \boldsymbol{r}_m - \boldsymbol{r}_{m'}$, where $\boldsymbol{r}_m$ is the center of group m, and for EFIE $\kappa = -j\frac{k\eta}{4\pi}$, while for the Magnetic Field Integral Equation (MFIE), $\kappa = -\frac{\eta}{4\pi}$, where η is the free-space impedance. For EFIE, the basis function signature $\boldsymbol{R}_{jm}(k,\hat{\boldsymbol{k}}_p) = \boldsymbol{V}_{jm}(k,\hat{\boldsymbol{k}}_p)^*$ and

$$\boldsymbol{V}_{jm}(k,\hat{\boldsymbol{k}}) = \int_{\boldsymbol{r}^2} \boldsymbol{f}_j(\boldsymbol{r}) \cdot [\bar{\bar{I}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}] e^{-jk\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}_m - \boldsymbol{r})}\, d^2\boldsymbol{r}, \tag{3}$$

and Rokhlins translation function T_L [9] is computed as

$$T_L(k,\hat{\boldsymbol{k}},\boldsymbol{x}) = \sum_{l=0}^{L} (-j)^l (2l+1) h_l^{(2)}(k|\boldsymbol{x}|) P_l(\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{x}}), \tag{4}$$

where $\hat{\boldsymbol{k}}$ is the unit wave vector, $\boldsymbol{x}$ is the vector between two group centers directed towards the receiving group, $\hat{\boldsymbol{x}} = \boldsymbol{x}/|\boldsymbol{x}|$, $h_l^{(2)}$ is the spherical Hankel function of second kind and order l, and P_l is the Legendre polynomial of order l.

Discretizing (2), we get

$$\bar{\bar{Z}}_{j,i} = \kappa \sum_{p=1}^{K} w_p \boldsymbol{R}_{jm}(k,\hat{\boldsymbol{k}}_p) \cdot \Big(T_L(k,\hat{\boldsymbol{k}}_p,\boldsymbol{r}_{mm'})\boldsymbol{V}_{im'}(k,\hat{\boldsymbol{k}}_p)\Big). \tag{5}$$

Format	COO	CSR	New
Values [GB]	6.97	6.97	6.97
Row Indices [GB]	3.4	0.007	0.007
Col. Indices [GB]	3.4	3.4	0.2
Total [GB]	13.6	10.2	7.2

TABLE I
THE MEMORY USE FOR STORING THE SPARSE NEAR-FIELD MATRIX FOR A SPECIFIC CONFIGURATION IN EACH OF THREE DIFFERENT FORMATS: COO (COORDINATE LIST), CSR (COMPRESSED SPARSE ROW) AND THE NEW FORMAT DISCUSSED HERE.

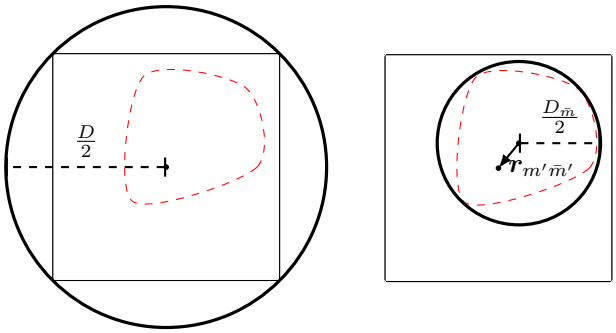

Fig. 1. 2-D illustration of adaptive grouping. The dashed line is the projection of a patch onto a plane, while the square is the box at the finest level of the Octree. To the left is shown the scenario obtained with using the Octree grouping at the lowest level as a foundation for the basis function patterns. Further subpartitioning would disect the patch, which is suboptimal. Instead, shown to the right, we introduce an adaptive grouping layer, which has its center such as to minimize the size of the circle enclosing the patch. We thus see that the region of validity, indicated by the bold black circle, is much smaller than if it had to enclose the entire square. The $\boldsymbol{r}_{m'\bar{m}'}$ vector indicates the phase-center shift needed to start the upward pass.

Here, w_p are the integration weights.

The number of terms K in the sum (5) is related to the number of terms L in the translation function (4) by the relation $K = 2(L+1)^2$. Since L is determined from the diameter D of the groups and the required relative accuracy $10^{-\beta}$ as

$$L = kD + 1.8\beta^{2/3}(kD)^{1/3}, \tag{6}$$

the obvious question is whether the decrease in the number of unknowns can counteract the increase in L and thus in K that comes from the increased basis function support and thus increased D. For a standard MLFMM implementation with a HO discretization, the increase in K (along with a larger $\bar{\bar{Z}}_{\text{near}}$) is far more devastating than the benefits of reducing N, which is essentially the conclusion reached by other research groups. However, the results in this paper show that with a few careful modifications to the standard algorithm, the benefits of Higher-Order discretizations can be combined with MLFMM to result in a low memory and computationally efficient algorithm.

III. IMPROVEMENTS

To achieve high performance for Higher-Order MLFMM, some improvements are needed. First, to reduce the noticeable memory burden for storing index integers for the sparse near-matrix, we apply a new format which uses only two integers to store an arbitrary number of consecutive column indices. With a good ordering of the basis functions, this yields a significant saving for a HO discretization, illustrated in Table I for a specific example.

Second, the use of an adaptive grouping scheme allows for much lower memory to be used for storing the basis function patterns (3) by adding an additional level beneath the lowest level in the Octree. With this, the center of the group $\boldsymbol{r}_m$ is changed to the center of an adaptive group $\boldsymbol{r}_{\bar{m}}$ in (3), such that (5) is changed to

$$\begin{aligned}\bar{\bar{Z}}_{j,i} =& \kappa \sum_{p=1}^{K} w_p \boldsymbol{V}^*_{j\bar{m}}(k, \hat{\boldsymbol{k}}_{\bar{p}}) \cdot \Big(W^T e^{-jk\hat{\boldsymbol{k}}\cdot\boldsymbol{r}_{\bar{m}m}} \\ & \cdot T_L(k, \hat{\boldsymbol{k}}_p, \boldsymbol{r}_{mm'}) e^{-jk\hat{\boldsymbol{k}}\cdot\boldsymbol{r}_{m'\bar{m}'}} W \boldsymbol{V}_{i\bar{m}'}(k, \hat{\boldsymbol{k}}_{\bar{p}}) \Big), \end{aligned} \tag{7}$$

where the overline refers to groups on the adaptive level. Noticeably, (7) requires an interpolation using the matrix W, which means that the adaptive grouping requires additional computational time. However, since this happens on the lowest level, where the number of samples is very low, the cost is low relative to the overall runtime. The concept of adaptive grouping is illustrated in Figure 1.

Finally, the application of the Spherical Harmonics Expansion (SHE), as introduced by Eibert [10], for storing the basis function patterns on the adaptive level is a vital component in any modern MLFMM implementation. The fundamental idea is the representation of the basis function patterns as a set of coefficients to the orthonormalized spherical harmonics, such that

$$\int_{\boldsymbol{r}^2} \boldsymbol{f}_j(\boldsymbol{r}) \cdot [\bar{\bar{I}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}] e^{-jk\hat{\boldsymbol{k}}\cdot(\boldsymbol{r}_{\bar{m}} - \boldsymbol{r})} \, d^2\boldsymbol{r} = \sum_{p=0}^{W} \sum_{q=-p}^{p} \boldsymbol{p}^j_{pq} Y_{pq}(\theta, \phi), \tag{8}$$

The coefficients $\boldsymbol{p}^j_{pq}$ are thus stored instead of k-space samples of (3). By representing the incoming patterns with another SHE, with coefficients $\boldsymbol{q}^i_{pq}$, the integration step (5) can be replaced with

$$\bar{\bar{Z}}_{j,i} = \kappa \sum_{p=0}^{W} \sum_{q=-p}^{p} \left(\boldsymbol{p}^j_{pq}\right)^* \cdot \boldsymbol{q}^i_{pq}. \tag{9}$$

To ensure a fast convergence of the SHE in (8), the basis function patterns are expressed with cartesian rather than spherical components. An on-the-fly conversion is then done after aggregation at the adaptive level to allow the group patterns to be expressed using only the two spherical components $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}})$.

IV. RESULTS

This section deals with two testcases, a PEC sphere and a PEC corner scatterer. Throughout, we use GMRES and an overlapping near-field preconditioner [11], but stress that the results are independent of the solution method. We note that whenever we discuss *total memory*, we refer to the memory required to store the entire MLFMM structure but not to solve the scattering problem. Thus, we disregard the storage required

for solvers, preconditioners and geometry, since this is not of focus in the present paper, but include everything required to perform a matrix-vector product; from basis function patterns and near-interaction matrix as well as minor temporary data such as interpolation matrices and various bookkeeping.

The discretization employed is the Higher-Order Legendre discretization [12]. The MLFMM implementation is described in detail in [13], but the discussion in the present paper focuses on practical considerations and includes a comparison with other results from the litterature.

A. Sphere

The first testcase concerns the scattering from a $1\,\mathrm{m}$ radius PEC sphere at $8\,\mathrm{GHz}$, illuminated by an $\hat{\boldsymbol{x}}$-polarized plane wave propagating along $+z$. The scattered electric field in the E-plane is computed and compared to the Mie series solution. The surface of the sphere is discretized using 4^{th} order curved quadrilaterals. We apply MLFMM with $\beta = 3$ to the CFIE.

As a measure of accuracy, we use the relative RMS, defined as

$$\text{Relative RMS} = \sqrt{\frac{\sum_{i=1}^{N_s}(|\boldsymbol{E}_{i,\mathrm{ref}}| - |\boldsymbol{E}_{i,\mathrm{cal}}|)^2}{\sum_{i=1}^{N_s}|\boldsymbol{E}_{i,\mathrm{ref}}|^2}}, \tag{10}$$

where $\boldsymbol{E}_{i,\mathrm{ref}}$ and $\boldsymbol{E}_{i,\mathrm{cal}}$ denotes the reference and calculated electric field at the i^{th} sample point, respectively, and N_s is the number of samples.

The memory usage is shown in Figure 2 as a function of the RMS for each order. The problem requires between 235200 and 940800 unknowns for the 1^{st} order (in the direction of the current) basis functions and between 187500 and 367500 unknowns for the 5^{th} order functions. We see that for higher accuracies, 3^{rd}–5^{th} order are more efficient than 2^{nd} order, and the first-order solution is substantially worse than the HO discretizations. However, we also clearly see that applying the modications detailed in the previous section results in huge savings, even for a first-order discretization.

Another interesting aspect, the computational time per matrix-vector product, is shown in Figure 3, where we see the strong incentive for using HO MLFMM. Aside from the fact that the order and time per matvec are inversely proportional, we also see that for higher orders, since the number of levels are constant throughout the RMS interval, the time is independent of the requested accuracy. Further, it is evident that the modifications in the previous section only results in moderate increases in the computational time, primarily due to the adaptive grouping. From the figure, it is also evident that the RMS is unchanged by the modifications, showing that there is no loss of accuracy associated with the memory savings, aside from the effects of applying an iterative solver.

Finally, while the sphere is an important reference case due to the existence of an analytical solution, most practical cases will have to use the EFIE. Figure 4 shows the results corresponding to Figure 2 for the EFIE — note the different scale on the y-axis. We see that for orders higher than 1, the memory usage is essentially independent of the order, though for very accurate results, 2^{nd} order is somewhat less efficient.

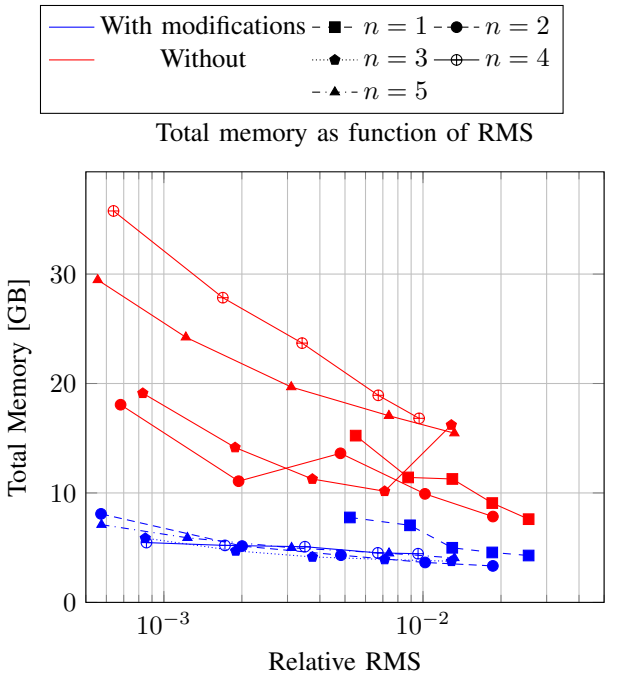

Fig. 2. The total memory for varying RMS and polynomial order n for $\beta = 3$, using CFIE, with and without the modifications.

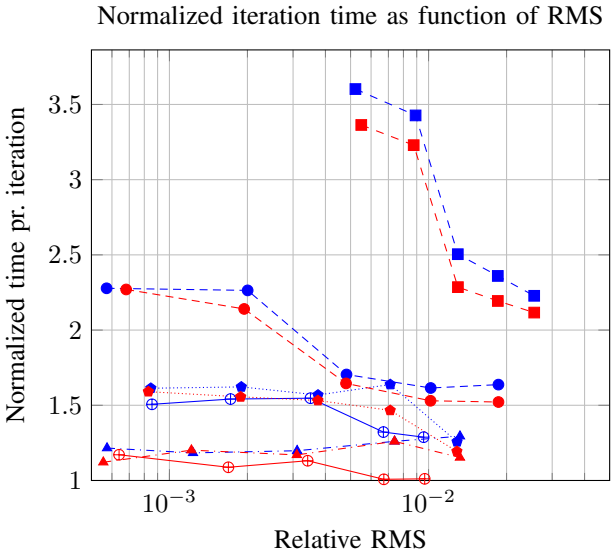

Fig. 3. The time pr. iteration, normalized to 1 for the fastest run, for the solutions shown in Figure 2. We see a direct connection between increasing the order and reducing the time pr. iteration. We also see that the cost of adding the modifications, as compared to the standard MLFMM implementation is noticeable, representing roughly 20% increase.

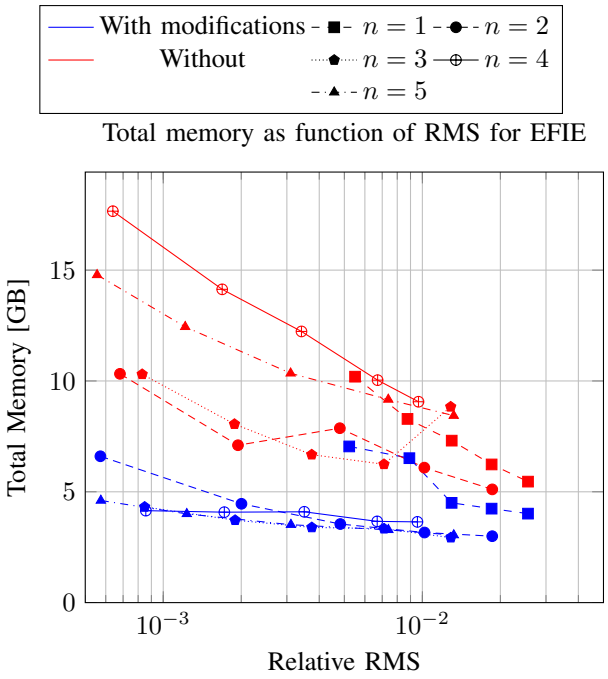

Fig. 4. The total memory for varying RMS and polynomial order for $\beta = 3$ for EFIE.

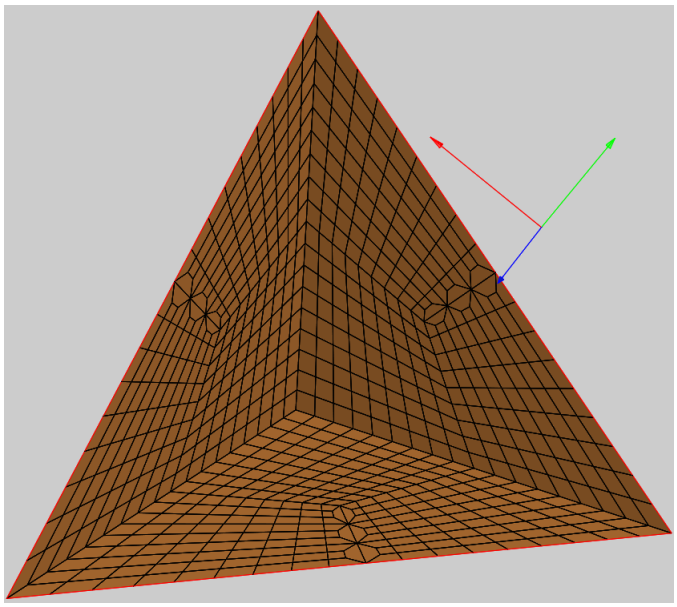

Fig. 5. The corner scatterer, looking into the aperture, along with an example of the mesh used. In front of the scatterer, a coordinate system illustrates the direction of incidence for the plane wave. The red, green and blue arrows detones the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$ components, respectively.

B. Corner reflector

As a second testcase, we consider an $\hat{x}$-polarized plane wave travelling in the positive z-direction, illuminating a PEC corner reflector, as illustrated in Figure 5. This excellent testcase was considered by Kolundzija et al. [7] and provides a good testcase for an MLFMM implementation due to the strong interactions between well-seperated parts of the scatterer. To compare our results with [7] we first consider the corner scatterer with a side length of 20λ, yielding a total surface area of $600\lambda^2$, a small testcase for MLFMM but designed to allow validation against existing MoM code.

In [7], the error quantifier was the mean error, defined as

$$\text{Mean error [dB]} = \frac{1}{N}\sum_{i=1}^{N} \Delta G_i \tag{11}$$

where ΔG_i was defined effectively as

$$\Delta G_i = \begin{cases} 0, & G_i < G_{\max} - R \\ |H_i - G_i| & G_i \geq G_{\max} - R \end{cases} \tag{12}$$

where H_i and G_i are the i^{th} points of the calculated and reference bistatic radar cross sections in dB, respectively, while $G_{\max}$ is the peak value of G_i. The reference solution is obtained by very finely discretized MoM. R is a threshold, set to 40 dB in [7]. However, we see no need to introduce a threshold and merely consider

$$\text{Mean error [dB]} = \frac{1}{N}\sum_{i=1}^{N} |H_i - G_i| \tag{13}$$

Having extracted the relevant results from that paper, Figure 6 is a comparison between the memory required for various orders in our implementation.

The figure shows the same overall conclusion as in the previous example — the modifications result in a very significant reduction in memory, and regardless of the choice of order, the memory required is roughly the same. In a direct comparison, we see that our implementation yields an order of magnitude better accuracy at roughly half the memory, except from order 2 from [7], where the accuracy is comparable but the memory used by our implementation reduced roughly by a factor of 5.

Finally, we consider a 60λ sidelength scatterer, a somewhat larger testcase. Here, since there is no straightforward way to achieve a reference solution, [7] just documents the required memory (5.6 GB) and number of unknowns (147987, up to 4th order). We apply a fairly fine discretization, yielding 432537 unknowns up to 4th order, using 1.15 GB memory, a factor of 5 reduction in memory. Although a direct accuracy comparison is of course unavailable, using nearly 3 times as many unknowns intuitively suggests that our accuracy is at least as good as in [7].

V. Conclusion

In spite of previous work suggesting otherwise, MLFMM with basis functions of orders higher than 2 can indeed be very efficient, both in terms of memory and speed, provided that extra care is taken in the implementation. We have demonstrated some additional techniques, both novel and known, and showed how their succesful implementation leads to a very efficient Higher-Order MLFMM. Finally, we have compared our implementation to previously published

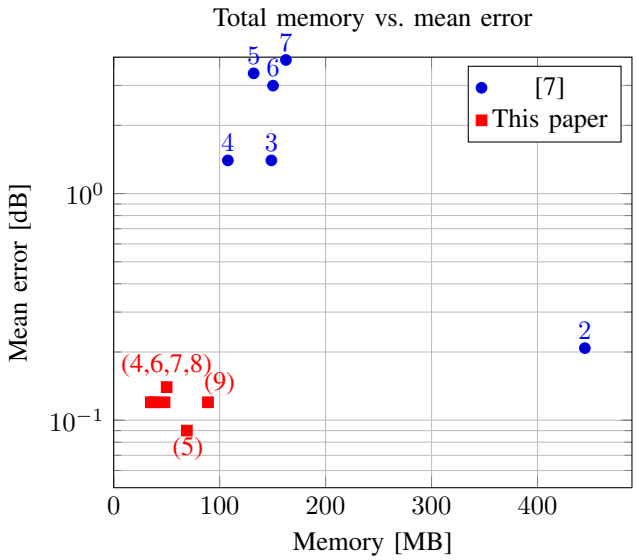

Fig. 6. The memory used for a specific accuracy for the 20λ sidelength corner scatterer, with results from [7] in blue and our implementation in red.

results, demonstrating a significant reduction in memory at comparable and even improved accuracies.

ACKNOWLEDGMENT

This work was partially funded by ESA-ESTEC, Contract No. 4000107963/13/NL.

REFERENCES

[1] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, 1982.

[2] B. M. Kolundzija and B. D. Popović, "Entire-domain Galerkin method for analysis of metallic antennas and scatterers," in *Microwaves, Antennas and Propagation, IEE Proceedings H*, 1993, pp. 1–10.

[3] C.-C. Lu and W. C. Chew, "A Multilevel Algorithm for Solving a Boundary Integral Equation of Wave Scattering," *Microwave and Optical Technology Letters*, vol. 7, no. 10, Jul. 1994.

[4] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microwave and Optical Technology Letters*, vol. 10, no. 1, pp. 14–19, Sep. 1995.

[5] K. C. Donepudi, J.-M. Jin, S. Velamparambil, J. M. Song, and W. C. Chew, "A Higher Order Parallelized Multilevel Fast Multipole Algorithm for 3-D Scattering," *IEEE Transactions on Antennas and Propagation*, vol. 49, no. 7, pp. 1069–1078, Jul. 2001.

[6] Ismatullah and T. F. Eibert, "Surface Integral Equation Solutions by Hierarchical Vector Basis Functions and Spherical Harmonics Based Multilevel Fast Multipole Method," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 7, pp. 2084–2093, Jul. 2009.

[7] B. M. Kolundzija, M. S. Pavlovic, and B. Mrdakovic, "Optimum Choice of Currents' Expansion Order in MLFMM algorithm for Electromagnetic Scattering," in *2009 IEEE Antennas and Propagation Society International Symposium (APSURSI)*, 2009.

[8] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, Jun. 1982.

[9] V. Rokhlin, "Diagonal Forms of Translation Operators for Helmholtz Equation in Three Dimensions," Tech. Rep., Mar. 1992.

[10] T. F. Eibert, "A diagonalized multilevel fast multipole method with spherical harmonics expansion of the k-space Integrals," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 2, pp. 814–817, Feb. 2005.

[11] E. Jørgensen, J. Volakis, P. Meincke, and O. Breinbjerg, "Divergence-Conforming Higher Order Hierarchical Basis Functions and Preconditioning Strategies for Boundary Integral Operators," in *Proc. 6th Int. Workshop on Finite Elements in Microw. Eng.*, Chios, Greece, May 2002.

[12] ——, "Higher Order Hierarchical Legendre Basis Functions for Electromagnetic Modeling," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 11, pp. 2985–2995, Nov. 2004.

[13] O. Borries, P. Meincke, E. Jørgensen, and P. C. Hansen, "Multi-level Fast Multipole Method for Higher-Order Discretizations," *IEEE Transactions on Antennas and Propagation*, submitted 2013.