Multilevel Fast Multipole Method for Higher Order Discretizations

Oscar Borries, *Student Member, IEEE*, Peter Meincke, *Member, IEEE*, Erik Jørgensen, *Member, IEEE*, and Per Christian Hansen

***Abstract*—The multi-level fast multipole method (MLFMM) for a higher order (HO) discretization is demonstrated on high-frequency (HF) problems, illustrating for the first time how an efficient MLFMM for HO can be achieved even for very large groups. Applying several novel ideas, beneficial to both lower order and higher order discretizations, results from a low-memory, high-speed MLFMM implementation of a HO hierarchical discretization are shown. These results challenge the general view that the benefits of HO and HF-MLFMM cannot be combined.**

***Index Terms*—Fast multipole method, higher order basis functions, integral equations.**

I. Introduction

The electromagnetic scattering problem can be solved by setting up a surface integral equation for the unknown surface current density. The discretization of the integral equation is often done using the method of moments (MoM) with local-domain basis functions [1]. The resulting system of linear equations is subsequently solved using either a direct or an iterative approach. The latter is the only viable option for large-scale problems, though some research has been done in applying direct methods to large problems [2].

Unfortunately, the matrix itself quickly grows too large to store as the electrical size of the problem increases. With N unknowns, the memory cost of storing the matrix increases as $\mathcal{O}(N^2)$ and so does the computational cost of the matrix-vector products required during the iterative solution. Both of these factors make the iterative MoM approach prohibitive. The computational electromagnetics community has sought to avoid this by using two different approaches; either by reducing the number of unknowns required to achieve a given accuracy or by applying fast solution methods that have an asymptotic complexity less than $\mathcal{O}(N^2)$.

The first of these approaches has primarily involved the study of basis functions and geometric discretizations with more desirable properties than the standard Rao–Wilton–Glisson (RWG) functions [3], which are first-order basis functions in the direction of the current, hereafter defined as lower order (LO). Improved basis functions can be obtained simply by defining basis function families that allow increasing the order of the RWG functions [4]. Other function families, which have the desirable property that they are hierarchical, have also been considered [5]–[7]. The hierarchical definition of the functions allows the expansion order to be chosen independently for each geometrical element (patch). A required supplement to higher order basis functions is the use of curved geometrical elements [4]–[6], that allows each element to represent a larger part of a curved scatterer and thus increases the support of the basis functions. The choice of basis functions and geometrical elements is critical for the accuracy of the solution and the number of iterations required for convergence [6].

The second approach has resulted in several new methods, some of which are still being refined. The most popular one seems to be the fast multipole method [8], [9] and its hierarchical variant, multilevel fast multipole method (MLFMM) [10]–[12].

Early research on combining the benefits of HO discretizations and MLFMM was done in [13], concluding that the larger patch sizes involved in HO discretizations severely limit the performance. To alleviate this problem, another approach was suggested in [14], where a Nyström-type approach is used to reduce the memory requirements. However, no comparisons between varying polynomial orders were done, and the savings achieved with this approach are significantly lower than those presented in the present paper. Later research focused on choosing basis functions suitable for MLFMM [15], thereby sacrificing the generality of the approach. Another line of research has focused on modifying some of the aspects of MLFMM [16], thereby alleviating some of the concerns regarding HO implementations. However, the results [17]–[19] only considered polynomials of first and second orders, as did [20], which describes an interpolatory HO implementation. Although [21] compared the number of iterations for a few implementations of HO function families, it did not consider the memory nor the accuracy of these implementations. Also, the effects of varying the basis function order were not considered.

More recently, research has switched to focusing on standard implementations of MLFMM with higher order, hierarchical discretizations [22]–[24]. However, these papers are rather brief and they suggest some suboptimal choices in the underlying implementations, causing concerns about whether the conclusions are viable and, more importantly, they fail to demonstrate higher order convergence. The explicit conclusion from [13],

Manuscript received October 25, 2013; revised April 25, 2014; accepted May 29, 2014. Date of publication June 13, 2014; date of current version September 01, 2014.

O. Borries is with TICRA, DK-1201 Copenhagen, Denmark, and also with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark (e-mail: ob@ticra.com).

P. Meincke and E. Jørgensen are with TICRA, DK-1201 Copenhagen, Denmark.

P. C. Hansen is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2014.2330582

[17], [22], [24] is that going beyond second-order is not worthwhile for MLFMM, either due to memory or accuracy concerns.

The main contribution of this paper is to challenge the previously established conclusion: HO convergence can be achieved while maintaining an efficient, low-memory, high-speed MLFMM implementation for high-frequency problems, and going beyond second order is indeed advantageous. The major contributions of this paper can be summarized as follows.

1) We demonstrate that the higher order convergence of the basis function set is maintained when using our MLFMM implementation.
2) We show that the memory cost of using HO discretizations with our proposed modifications (see below) to MLFMM is far lower than previously described in the literature.
3) We show that the computational time for a matrix-vector product is significantly lower using HO MLFMM compared to LO MLFMM.

To arrive at the accurate, low-memory, high-speed HO MLFMM scheme, several parts of the standard MLFMM implementation must be modified. The key modifications necessary for obtaining the new HO MLFMM implementation of the present paper are as follows:

1) We use a sparse matrix storage format suited for HO implementations.
2) We introduce a new grouping scheme, significantly reducing the memory required, especially for nonuniform discretizations.
3) We demonstrate the savings achieved by Eibert's SE-MLFMM [16] for hierarchical HO discretizations of arbitrary order.

Some of the modifications are novel and of interest to both HO and LO MLFMM, but our main contribution is the revision of the paradigm that HO MLFMM is not worthwhile.

The paper is organized as follows. Section II describes the discretization employed, highlighting the key parameters to be used later. Section III then recaps the main points of an MLFMM implementation, focusing on the key differences imposed by the use of HO bases and introducing the novel contributions. Section IV analyzes three different test cases, discussing the relationship between computation time, memory usage and accuracy for various discretizations. Section V concludes on the results. The major notation is as follows.

- Scalar quantities are typeset using *italics*. Physical vectors, either with two or three components, are typeset using **bold**. Other vectors are overlined while matrices are double overlined.
- The symbol $*$ denotes complex conjugation, while $\cdot$ denotes the vector dot product with neither vector conjugated.

II. Discretization

The fundamental problem to solve is the integral equation obtained by considering an incident time-harmonic electromagnetic wave on a perfectly electrically conducting (PEC) scatterer. The time factor $e^{j\omega t}$, where ω is the angular frequency, is assumed and suppressed throughout. This problem can be solved via a mixed potential electric field integral equation (EFIE) formulation [25]. Denoting the surface $\mathcal{S}$, the EFIE is then

$$\mathcal{L}\boldsymbol{J}_{\mathcal{S}} = \hat{\boldsymbol{n}} \times \boldsymbol{E}^i \tag{1}$$

where $\hat{\boldsymbol{n}}$ is a unit vector normal to $\mathcal{S}$, $\boldsymbol{E}^i$ is the incident electric field, and $\boldsymbol{J}_{\mathcal{S}}$ is the surface current density. $\mathcal{L}$ is the integral operator

$$\mathcal{L}\boldsymbol{J}_{\mathcal{S}} = \hat{\boldsymbol{n}} \times j\omega\mu \left[\int_{\mathcal{S}} \boldsymbol{J}_{\mathcal{S}}(\boldsymbol{r}')G(\boldsymbol{r},\boldsymbol{r}')d^2\boldsymbol{r}' + \frac{1}{k^2}\int_{\mathcal{S}} \nabla'_{\mathcal{S}} \cdot \boldsymbol{J}_{\mathcal{S}}(\boldsymbol{r}')\nabla G(\boldsymbol{r},\boldsymbol{r}')d^2\boldsymbol{r}' \right] \tag{2}$$

where μ is the free-space permeability and $k = 2\pi/\lambda$, λ being the free-space wavelength. $G(\boldsymbol{r},\boldsymbol{r}')$ is the free-space Green's function $G(\boldsymbol{r},\boldsymbol{r}') = e^{-jk|\boldsymbol{r}-\boldsymbol{r}'|}/4\pi|\boldsymbol{r}-\boldsymbol{r}'|$ and $\boldsymbol{r}$, $\boldsymbol{r}'$ denote observation and integration points, respectively.

To eliminate the interior resonances of the EFIE operator it can be combined with the magnetic field integral equation (MFIE) [25], which for a smooth, closed scatterer is

$$\left(\frac{1}{2}\mathcal{I} + \mathcal{K}\right)\boldsymbol{J}_{\mathcal{S}} = \hat{\boldsymbol{n}} \times \boldsymbol{H}^i \tag{3}$$

in which $\boldsymbol{H}^i$ is the incident magnetic field, $\hat{\boldsymbol{n}}$ is an outward unit normal vector, $\mathcal{I}$ is the identity operator, and $\mathcal{K}$ is the operator

$$\mathcal{K}\boldsymbol{J}_{\mathcal{S}} = \hat{\boldsymbol{n}} \times ⨍_{\mathcal{S}} \boldsymbol{J}_{\mathcal{S}}(\boldsymbol{r}') \times \nabla G(\boldsymbol{r},\boldsymbol{r}')d^2\boldsymbol{r}' \tag{4}$$

where $⨍$ denotes the Cauchy principal value. This yields the combined field integral equation (CFIE) [25]

$$\left[\alpha\mathcal{L} + (1-\alpha)\eta\left(\frac{1}{2}\mathcal{I} + \mathcal{K}\right)\right]\boldsymbol{J}_{\mathcal{S}} = \alpha\hat{\boldsymbol{n}}\times\boldsymbol{E}^i + (1-\alpha)\eta\hat{\boldsymbol{n}}\times\boldsymbol{H}^i. \tag{5}$$

Here, $\eta = \sqrt{\mu/\epsilon}$ is the free-space impedance, ϵ is the free-space permittivity and $\alpha \in [0,1]$ is a weighting factor.

Through the Galerkin approach, a matrix equation $\overline{\overline{Z}}\,\overline{I} = \overline{V}$ is obtained, the solution to which will yield the required surface current densities—whose accuracy depends on the discretization. This motivates the discretization used in the present implementation, taken from [6], [26], where both the current $\boldsymbol{J}_{\mathcal{S}}$ and the surface geometry $\mathcal{S}$ are discretized using a higher order approach. The geometry is discretized with curved quadrilaterals [4], such that a point $\boldsymbol{r}(u,v)$ on a pth order patch can be expressed as

$$\boldsymbol{r}(u,v) = \sum_{i=0}^{p}\sum_{j=0}^{p} \boldsymbol{r}_{ij}\phi_i^p(u)\phi_j^p(v) \tag{6}$$

where $\boldsymbol{r}_{ij}$ denotes an interpolation node and $\phi_i^p(u)$ is the Lagrange polynomial of pth order

$$\phi_i^p(u) = \prod_{k=0,k\neq i}^{p} \frac{u-u_k}{u_i-u_k} \tag{7}$$

where u_i is the parametric coordinate of the interpolation node.

The currents are discretized using modified Legendre polynomials along the direction of the current, and Legendre polynomials in the transverse direction. Hence, $\boldsymbol{J}_S$ is expanded as

$$\boldsymbol{J}_{\mathcal{S}} = J_{\mathcal{S}}^u \boldsymbol{a}_u + J_{\mathcal{S}}^v \boldsymbol{a}_v \tag{8}$$

where $\boldsymbol{a}_u = \partial \boldsymbol{r}/\partial u$ is the covariant unitary vector and similarly for $\boldsymbol{a}_v$. Considering a u-directed current, $J_{\mathcal{S}}^u$ is thus expanded as

$$J_{\mathcal{S}}^u(u,v) = \frac{1}{\mathcal{J}_{\mathcal{S}}(u,v)} \sum_{m=0}^{M^u} \sum_{n=0}^{N^v} a_{mn}^u C_m^u \widetilde{P}_m(u) C_n^v P_n(v) \tag{9}$$

where the modified Legendre polynomials $\widetilde{P}_m$ are defined as

$$\widetilde{P}_m(u) = \begin{cases} 1-u & m=0 \\ 1+u & m=1 \\ P_m(u) - P_{m-2}(u) & m \geq 2 \end{cases} \tag{10}$$

and P_m are the Legendre polynomials of order m. $\mathcal{J}_{\mathcal{S}}(u,v)$ is the surface Jacobian. The expansion for a v-directed current $J_{\mathcal{S}}^v$ is obtained by interchanging u and v in (9). Moreover, C_m is a factor chosen to minimize the condition number of $\overline{\overline{Z}}$. To satisfy the Nedelec constraint [27], $N^v = M^u - 1$ for a u-directed current. The order of the expansion is said to be M^u, but it is complete to order $M^u - 1$ only. The term *order*, denoted throughout this paper by n, refers to the value of M^u and M^v.

While our discretization is as detailed above, we emphasize that the improved higher order MLFMM algorithm presented in this paper is applicable for any choice of higher order basis functions and does not depend on any specific property of the Legendre basis functions nor of the curved quadrilateral patches.

III. MLFMM FOR HIGHER ORDER BASES

The MLFMM is a procedure for performing the operation $\overline{\overline{Z}}\,\overline{I}$ in $\mathcal{O}(N \log N)$ time and memory, where N is the number of basis functions. MLFMM relies on a hierarchical grouping of the patches, often done using an Octree [28], based on the center points of the patches. The interactions between closely positioned groups are computed as usual in MoM, resulting in a sparse near-matrix $\overline{\overline{Z}}_{\text{near}}$ containing the elements from $\overline{\overline{Z}}$ that cannot be approximated by MLFMM.

Interactions that can be approximated with MLFMM are computed by performing an integral over the sphere, which allows for computing interactions between entire groups of basis functions simultaneously. This process involves gathering the radiation pattern of a group by summing up the excitations of the basis functions in the group, translating this using the Rokhlin transfer function (12), and then integrating against the receiving pattern of a testing group. Furthermore, by utilizing the hierarchical scheme, involving interpolations due to differing sample rates, interactions between larger and larger groups can be computed using this integral.

This yields a splitting of the operation $\overline{\overline{Z}}\,\overline{I}$ into

$$\overline{\overline{Z}}\,\overline{I} \simeq \overline{\overline{Z}}_{\text{near}}\overline{I} + \mathcal{F}(\overline{I}). \tag{11}$$

It has been shown [11] that $\overline{\overline{Z}}_{\text{near}}$ contains $\mathcal{O}(N)$ elements, thus requiring $\mathcal{O}(N)$ memory and computational time, while the operation $\mathcal{F}(\overline{I})$ requires $\mathcal{O}(N \log N)$ memory and computational time.

The essential part of MLFMM is Rokhlin's translation function [8]

$$T_L(\boldsymbol{k}, \boldsymbol{x}) = \sum_{l=0}^{L} (-j)^l (2l+1) h_l^{(2)}(k|\boldsymbol{x}|) P_l(\hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{x}}) \tag{12}$$

where $\hat{\boldsymbol{k}}$ is the unit wave vector, $\boldsymbol{x}$ is the vector between two group centers, $\hat{\boldsymbol{x}} = \boldsymbol{x}/|\boldsymbol{x}|$, and $h_l^{(2)}$ is the spherical Hankel function of second kind and order l. It is important to note that T_L does not depend on the absolute position of the groups, but only on the vector $\boldsymbol{x}$ between their centers. Thus, T_L can be reused for pairs of groups with the same $\boldsymbol{x}$; a key factor in keeping memory requirements low.

The truncation number L from (12) is determined from the excess bandwidth formula [29]

$$L = kD + 1.8\beta^{\frac{2}{3}}(kD)^{\frac{1}{3}} \tag{13}$$

where $D = \sqrt{3}a$ is the diameter of the group, a is the side-length, and $10^{-\beta}$ is the desired relative error. We apply the one buffer-box criterion $N_{\text{BUF}} = 1$, such that interactions between groups that are *well-separated*, i.e., $|\boldsymbol{x}| > N_{\text{BUF}}D$, are calculated by MLFMM. We note that this can potentially cause problems for small group sizes—this problem is treated in detail in [30]. However, the approach from [30] yields extremely conservative estimates, which significantly increase the resource consumption, especially for small group sizes. Thus, to avoid being unfair towards LO basis functions, we do not use the methods from [30] but only warn that this can yield a pessimistic upper bound to the accuracy achievable, particularly for LO discretizations.

With T_L, the mutual impedance between two basis functions $\boldsymbol{f}_j$ and $\boldsymbol{f}_i$, located in groups m and m', respectively, that are well separated, can be expressed as

$$Z_{j,i} \simeq \kappa \oiint \boldsymbol{R}_{jm}(\boldsymbol{k}) \cdot (T_L(\boldsymbol{k}, \boldsymbol{r}_{mm'}) \boldsymbol{V}_{im'}(\boldsymbol{k}))\, d^2\hat{\boldsymbol{k}} \tag{14}$$

where $\boldsymbol{r}_{xy}$ is defined as $\boldsymbol{r}_{xy} = \boldsymbol{r}_x - \boldsymbol{r}_y$ and $\boldsymbol{r}_m$ is the center of the group m. The basis function patterns $\boldsymbol{V}_{jm}$ and $\boldsymbol{R}_{jm}$ for EFIE are

$$\boldsymbol{V}_{jm}(\boldsymbol{k}) = \int_{\mathcal{S}_j} \boldsymbol{f}_j(\boldsymbol{r}) \cdot \left[\overline{\overline{I}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}\right] e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}_m - \boldsymbol{r})} d^2\boldsymbol{r} \tag{15}$$

$$\boldsymbol{R}_{jm} = \boldsymbol{V}_{jm}^* \tag{16}$$

and for MFIE

$$\boldsymbol{V}_{jm}(\boldsymbol{k}) = -\hat{k} \times \int_{\mathcal{S}_j} e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}_m - \boldsymbol{r})} \left[\boldsymbol{f}_j(\boldsymbol{r}) \times \hat{n}(\boldsymbol{r})\right] d^2\boldsymbol{r} \tag{17a}$$

$$\boldsymbol{R}_{jm}(\boldsymbol{k}) = \int_{\mathcal{S}_j} e^{-j\boldsymbol{k}\cdot(\boldsymbol{r} - \boldsymbol{r}_m)} \boldsymbol{f}_j(\boldsymbol{r}) d^2\boldsymbol{r} \tag{17b}$$

where $\boldsymbol{r}$ covers the patch $\mathcal{S}_j$ on which $\boldsymbol{f}_j$ is defined. For EFIE, $\kappa = -j(k\eta/4\pi)$, while for MFIE $\kappa = -(\eta/4\pi)$.

We note that only the $\hat{\theta}$ and $\hat{\phi}$ components of the basis function patterns are nonzero. We also mention that there is a symmetry [31] that can be employed to store only half of the basis function patterns.

When (14) is evaluated numerically, the expression is rewritten through the substitution $t = \cos\theta$

$$Z_{j,i} \simeq \kappa \int_{-1}^{1} \int_{0}^{2\pi} \boldsymbol{R}_{jm}(\boldsymbol{k}) \cdot (T_L(\boldsymbol{k}, \boldsymbol{r}_{mm'})\boldsymbol{V}_{im'}(\boldsymbol{k}))\, d\phi dt \quad (18)$$

which is then discretized to

$$Z_{j,i} \simeq \kappa \sum_{p=1}^{K} w_p \boldsymbol{R}_{jm}(\boldsymbol{k}_p) \cdot (T_L(\boldsymbol{k}_p, \boldsymbol{r}_{mm'})\boldsymbol{V}_{im'}(\boldsymbol{k}_p)). \quad (19)$$

Here, $K = 2(L+1)^2$ is the quadrature points; $L+1$ Gauss–Legendre points in t and $2(L+1)$ equidistant points between 0 and $2\pi - 2\pi/2(L+1)$ in ϕ. w_p are the corresponding quadrature weights.

The above expressions are valid for a single-level scheme. With two levels, (19) becomes

$$\begin{aligned} Z_{j,i} \simeq \kappa \sum_{p=1}^{K^{(3)}} w_p^{(3)} \boldsymbol{R}_{jm_4}(\boldsymbol{k}_p^{(4)}) \cdot \Big(e^{-j\boldsymbol{k}_p^{(3)} \boldsymbol{r}_{m_4 m_3}} \\ \cdot T_L(\boldsymbol{k}_p^{(3)}, \boldsymbol{r}_{m_3 m_3'}) e^{-j\boldsymbol{k}_p^{(3)} \cdot \boldsymbol{r}_{m_3' m_4'}} \boldsymbol{V}_{im_4'}(\boldsymbol{k}_p^{(4)})\Big). \quad (20) \end{aligned}$$

Here, the superscript (q) denotes quantities at the qth level, with lower numbers indicating coarser levels, i.e., the diameter of the groups increases as q decreases, such that $D^{(q)} = 2D^{(q+1)}$. Further, m_q is the group at the qth level containing group m, and levels 1 (the original bounding box) and 2 are not used in the MLFMM.

The above approach can be applied hierarchically for additional levels, i.e., the centers of the plane-wave expansions are moved from the center of groups at successive levels by the phase-center movement $e^{-j\boldsymbol{k}\cdot\boldsymbol{r}}$. As (20) reveals, the key difference between the single- and multilevel scheme is that for multilevel we need an interpolation routine to get from one level to another in the Octree, since the sampling rates differ (13). In the present implementation, this is done by Lagrangian interpolation with anti-spherical and spherical boundary conditions on the $\hat{\theta}$ and $\hat{\phi}$ components, respectively,

$$f(\theta, \phi) = \begin{cases} f(2\pi - \theta, \phi + \pi) & \text{Spherical,} \\ -f(2\pi - \theta, \phi + \pi) & \text{Anti-spherical,} \end{cases} \theta \in]\pi, 2\pi].$$

To avoid allocating both outgoing and incoming radiation patterns for each group, we reuse the storage such that memory is allocated for the outgoing patterns for each group plus a reusable space for the incoming patterns on each level.

Having set the stage and described the basic concepts of a standard MLFMM implementation, we now turn to the novel contributions of the paper, as detailed in the Introduction.

A. Near-Matrix Elements

An important consideration in HO-MLFMM is that each half of a basis function overlapping two patches, also referred to as an *edge basis function*, should be placed into separate groups. The scenario is illustrated in Fig. 1. Here, we see that half of the edge basis function c is placed in group 1, while the other half is placed in group 2. As a consequence, when considering the interaction of function c with function b in group 6, the interaction will be split between the near-matrix and the MLFMM operation. Thus, the corresponding matrix element $Z_{c,b}$ will require a contribution from both $\overline{\overline{Z}}_{\text{near}}$ and (20). In our implementation, we allow for this and thus let the group size equal the largest patch size, rather than expanding the groups such that $Z_{c,b}$ is entirely contained in $\overline{\overline{Z}}_{\text{near}}$.

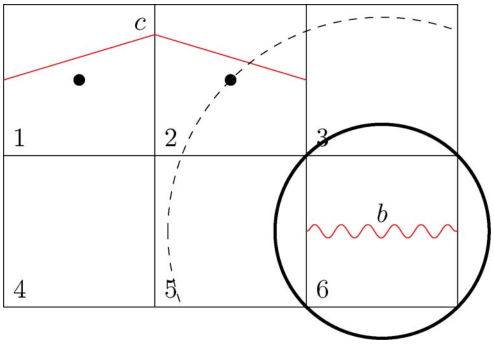

Fig. 1. In this simple example the grid lines represent the Octree grouping in 2-D, which precisely coincides with six patches. The edge basis function c is split between groups 1 and 2, while the patch basis function b belongs to group 6. The solid circle has diameter D and the dashed line denotes the range outside which the FMM approximation with $N_{\text{BUF}} = 1$ is applicable. We thus see that group 2 is in the near-interaction list of group 6, while group 1 is not. Since c is an edge basis function, its interaction with b will consist of a contribution from both the near- and far-interaction parts of the MLFMM.

An important part of the standard MLFMM implementation is the storage format for the sparse near-field matrix. A common choice is a block storage format, where each block represents the near-interaction between two groups. Unfortunately, a block format is impractical when basis functions span two groups, since the contributions to element $Z_{\text{near}(i,j)}$, when either $\boldsymbol{f}_i$ or $\boldsymbol{f}_j$ is an edge basis function, are stored in more than one block, leading to increased memory usage due to duplicated elements. Further, there is additional bookkeeping involved because it is not possible to find a contiguous ordering of the basis functions within each group for an arbitrary scatterer. Therefore, another format is desirable.

Another typical choice is the compressed sparse row (CSR) format [32], which is well suited for matrix-vector products. It does, however, have the unfortunate property of spending more than one-third of the storage of the near-matrix on storing integers. To avoid this, another approach is used here.

For most discretizations, where the basis functions belonging to a given group are placed consecutively in the impedance matrix, the elements in the near-matrix lie in contiguous blocks. This is particularly pronounced for HO discretizations, where there is a large number of basis functions on each patch.

We thus propose to modify the CSR format by not explicitly storing consecutive column indices. Instead, we store the first index in such a set, and store the number of consecutive indices in the set as a negative number. Thus, we only store roughly $2t$ integers for column indices, where t is the number of sets in total. This has the advantage of being completely adaptive to the discretization and requiring only minor modifications to existing code based on CSR, as well as being just as parallelizable and easy to modify for the symmetric case.

B. Well-Separated Elements

The limiting factor of performance in HO MLFMM is the size of the domains. The following presents two techniques, one novel and one which is adopted from [16], which together drastically reduce the memory required for storing the basis function patterns. Note that while these are relatively more advantageous for HO discretizations, they will still reduce the memory requirements, even for LO MLFMM. Whether they are worthwhile or not for LO MLFMM is not of concern here, although it is demonstrated in Section IV.

1) Adaptive Grouping: The major problem with the large patches in HO discretizations in connection with MLFMM is the very significant discrepancy between the size of the patches and the size of the finest group. Since for a given scatterer the group size at each level of an Octree is fixed, and since it is not worthwhile partitioning a patch into distinct groups, we easily risk a scenario where the sidelength of the group is close to twice the patch sidelength. This is extremely inefficient because the basis function patterns will have a far larger bandwidth than necessary, and a new approach has to be devised.

We propose an *adaptive grouping*, which essentially adds a separate, extra layer, called the *adaptive level*, at the finest level. This layer is distinct from the Octree grouping, such that the center and group size of each adaptive group can be chosen as desired. In principle, this works analogously to an extra layer in the Octree, but crucially, it is not restricted by the same rules as those that apply to the Octree grouping scheme. The advantages are:

- The center of each group can be chosen such that the region of validity is as small as possible, while still being conformal to the patch inside each group.
- For highly irregular meshes, the Octree grouping scheme results in groups that are not smaller than the largest patch, which might be a very poor choice in very finely meshed regions. The adaptive grouping allows for arbitrarily sized groups in different regions of the mesh.
- There is little need to carefully choose the patch size to fit with the Octree grouping, as [17] mentions is necessary if HO MLFMM is to be useful. With adaptive grouping, the bandwidth of the basis function patterns will fit the patch size.
- This approach allows us to efficiently solve the classic problem of patches sticking out of the groups. Instead of expanding all groups at the lowest level, one can instead expand just the adaptive group and the groups at higher levels, which is less costly since the groups now only hold aggregated patterns instead of basis function patterns.

We stress that there is no translation done on the adaptive level, and therefore the near-interaction matrix is still based on the finest level in the Octree, and there is no loss of translation accuracy from the adaptive grouping since the categorization of well-separated elements is unaffected. The only change from the usual MLFMM scheme is thus that the basis function patterns are tabulated on the adaptive level. Fig. 2 illustrates the effects of the reduced size of the adaptive group.

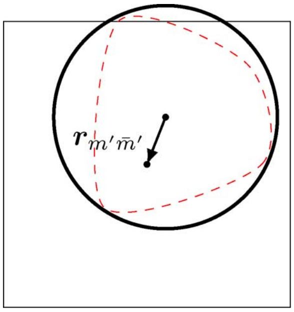

Fig. 2. 2-D illustration of adaptive grouping. The dashed line is the projection of a patch onto a plane, while the square is the box at the finest level of the Octree. However, further sub-partitioning would dissect the patch, which is suboptimal. Instead, we introduce an adaptive grouping layer, which has its center $\boldsymbol{r}_{\bar{m}'}$ such as to minimize the size of the circle enclosing the patch. We note that the patch is allowed to overlap the box, as shown, as long as it isn't further than $D/2$ from the center. We thus see that the region of validity, indicated by the solid black circle, is much smaller than it would be if it had to enclose the entire square. The $\boldsymbol{r}_{m'\bar{m}'}$ vector indicates the phase-center movement needed to start the upward pass.

The work by Pan *et al.* [20] discusses an approach to locally extend the Octree, providing extra levels that allow a lower sampling rate. The extra levels are obtained by further subdivision of the finest level of the Octree. Their approach differs from the adaptive grouping presented here in a few key areas:

- Being based on further subdivision, their approach does not allow for arbitrarily sized groups, instead only allowing for further halving of the group diameters. Adaptive grouping allows for any group size, adaptive to the specific patch, providing the lowest possible sampling rates.
- Their approach does not allow for an adaptive group center. This is a key feature of adaptive grouping, as it allows for significantly lower sampling rates, even for very irregular meshes or meshes where the finest level groups of the Octree has their centers positioned far from the patches.
- Both of these advantages of adaptive grouping over the technique presented in [20] are particularly important for very large patches. [20] did not consider patches larger than 0.2λ.

When implementing the adaptive grouping in the matrix-vector product, (19) is thus expanded to

$$Z_{j,i} \simeq \kappa \sum_{p=1}^{K} w_p \boldsymbol{R}_{j\bar{m}}(\boldsymbol{k}_{\bar{p}}) \cdot \Bigg(\overline{\overline{W}}^{T} e^{-j\boldsymbol{k}_p \cdot \boldsymbol{r}_{\bar{m}m}} \cdot T_L(\boldsymbol{k}_p, \boldsymbol{r}_{mm'}) e^{-j\boldsymbol{k}_p \cdot \boldsymbol{r}_{m'\bar{m}'}} \overline{\overline{W}} \boldsymbol{V}_{i\bar{m}'}(\boldsymbol{k}_{\bar{p}}) \Bigg) \tag{21}$$

where $\bar{m}$ is the adaptive group in which $\boldsymbol{f}_j$ resides and equivalently for $\bar{m}'$ and $\boldsymbol{f}_i$. Thus, we can express the adaptive grouping as an expansion of the basis function pattern $\boldsymbol{R}_{jm}$ or $\boldsymbol{V}_{im'}$ from (19) as

$$\boldsymbol{V}_{im'}(\boldsymbol{k}) = e^{-j\boldsymbol{k} \cdot \boldsymbol{r}_{m'\bar{m}}} \overline{\overline{W}} \boldsymbol{V}_{i\bar{m}'}(\boldsymbol{k}). \tag{22}$$

Aside from the controllable interpolation error, (22) is exact.

Since this modification requires some additional auxiliary data, in particular an interpolation matrix W, a practical implementation would choose a number of possible group sizes, and then categorize the adaptive groups into these, instead of having auxiliary data for every adaptive group. For fairly uniform meshes, only 2 or 3 possible group sizes are required, while strongly nonuniform meshes might require a few more. In any case, the auxiliary data is negligible, but has been included in the memory counts in the numerical results presented later.

2) Spherical Harmonics Expansion: In [16], Eibert develops a method for storing the basis function patterns as coefficients to a spherical harmonics expansion (SHE), which drastically reduces the required memory. The key equations are repeated here for completeness.

We expand the basis function patterns (either (15) or (17) can be used here) as

$$\int_S \boldsymbol{f}_j(\boldsymbol{r}) \cdot \left[\overline{\overline{I}} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}\right] e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}_m - \boldsymbol{r})} d^2\boldsymbol{r} = \sum_{p=0}^{W} \sum_{q=-p}^{p} \boldsymbol{p}_{pq}^{j} Y_{pq}(\theta, \phi) \tag{23}$$

where Y_{pq} are the orthonormalized spherical harmonics, P_p^q are the associated Legendre functions, and W is the order of the SHE. The coefficients $\boldsymbol{p}_{pq}^j$ are then stored instead of the sampled basis function patterns. The coefficients are computed numerically via the integral

$$\boldsymbol{p}_{pq}^{i} = \oiint \boldsymbol{V}_i(\theta, \phi) Y_{pq}^*(\theta, \phi) d^2\hat{\boldsymbol{k}}. \tag{24}$$

Although V is stored most efficiently in $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}})$-components, W can be made significantly lower by using $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$-components due to the singularity of $\hat{\boldsymbol{\theta}}$ at the poles.

Further, converting the incoming fields as

$$T_L(\boldsymbol{k}, \boldsymbol{r}_{mm'}) \boldsymbol{V}_{im}(\boldsymbol{k}) = \sum_{p=0}^{W} \sum_{q=-p}^{p} \boldsymbol{r}_{pq}^{i} Y_{pq}(\theta, \phi) \tag{25}$$

we can convert (19) to

$$Z_{j,i} \simeq \kappa \sum_{p=0}^{W} \sum_{q=-p}^{p} \left(\boldsymbol{p}_{pq}^{j}\right)^* \cdot \boldsymbol{r}_{pq}^{i}. \tag{26}$$

The procedure for computing a matrix-vector product is only changed at the finest level. Here, we aggregate the SHE coefficients, evaluate the resulting radiation pattern for the group, convert to $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}})$ components, and then continue as in the standard MLFMM procedure. For disaggregation, we convert the incoming fields to Cartesian components, convert to a SHE for a given group using (25), and then evaluate (26). The integration weights required to evaluate (25) are multiplied onto the translation operators, so very little additional work is required to implement this method in an already working MLFMM scheme.

The choice of W in (23) is proposed in [16] to be determined roughly as $W \approx L/2$ for LO discretizations. In [17], it is mentioned that the choice $W = \min(\mathrm{ceil}(L/2), 5)$ is appropriate for HO discretizations. The latter estimate seems far too optimistic and a more theoretical approach is taken here.

Since the spherical harmonics form an orthonormal basis, we can calculate $\boldsymbol{V}_j$ for the largest of the adaptive groups and the highest order basis function $\boldsymbol{f}_j$ in that group. We can then adjust W until $g(W) < 10^{-2\beta}$, where $g(W)$ is defined as

$$g(W) = \max_{(c)} \left\{ \frac{\oiint \left|V_j^{(c)}(\boldsymbol{k})\right|^2 d^2\hat{\boldsymbol{k}} - \sum_{p=0}^{W} \sum_{q=-p}^{p} \left|p_{pq}^{j,(c)}\right|^2}{\oiint \left|V_j^{(c)}(\boldsymbol{k})\right|^2 d^2\hat{\boldsymbol{k}}} \right\} \tag{27}$$

where (c) runs through the components. The W found here is then used in all subsequent calculations.

In practice, however, it is sufficient to investigate the general behavior of $g(W)$ for the scalar case of $V_i(\boldsymbol{k}) = e^{j\boldsymbol{k}\cdot\boldsymbol{r}}$, letting $|\boldsymbol{r}| = D/2$ vary. This allows for rough estimates of W to be tabulated in advance using (27), as a function of D. It is worth mentioning that for $\beta \leq 2$, this approach reveals that the original $W \approx L/2$ estimate is reasonable, while for better accuracies it is slightly too optimistic. Thus, we use $W = L/2 + \max(\beta - 2, 0)$.

IV. Numerical Results

This section contains three test cases designed to demonstrate the savings achieved by the modified implementation. The first two examples have reference solutions, allowing us to compare the achieved accuracy and computational requirements across varying discretizations. The last example is a demonstration of the capabilities of the implementation on a realistic scatterer. We note that whenever we discuss *MLFMM 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 the focus of 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. We apply GMRES and an overlapping near-field preconditioner [33], but stress that the results are independent of the iterative method and preconditioner. The iteration timings include the inner operations of GMRES and the cost to precondition, but these represent less than 0.5% of the times and are thus irrelevant to the conclusions.

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

$$\text{Relative RMS Error} = \sqrt{\frac{\sum_{i=1}^{N_s} |\boldsymbol{E}_{i,\mathrm{ref}} - \boldsymbol{E}_{i,\mathrm{cal}}|^2}{\sum_{i=1}^{N_s} |\boldsymbol{E}_{i,\mathrm{ref}}|^2}} \tag{28}$$

where $\boldsymbol{E}_{i,\mathrm{ref}}$ and $\boldsymbol{E}_{i,\mathrm{cal}}$ denote the electric far field at the $\{i$th sample point from the reference and calculated scattered fields, respectively, and N_s is the number of samples. The computation times shown are measured without parallelization.

A. Sphere

The first example concerns the scattering from a 1-m radius PEC sphere at 8 GHz, illuminated by an $\hat{\boldsymbol{x}}$-polarized plane wave propagating along $+z$. The sphere is discretized using fourth-order curved quadrilaterals. The problem requires between 235 200 and 940 800 unknowns for the first-order basis functions and between 187 500 and 367 500 unknowns for the fifth-order functions. We apply MLFMM with $\beta = 3$ in (13)

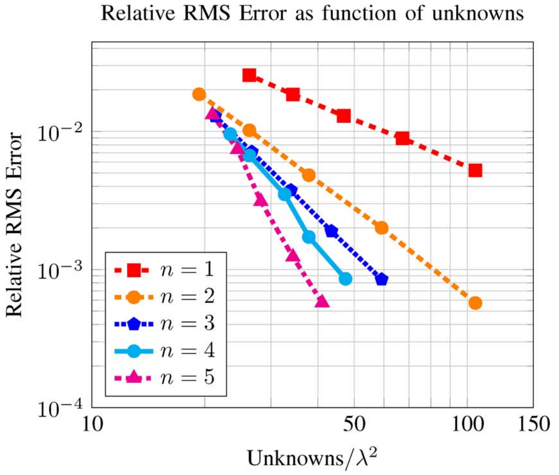

Fig. 3. Higher order convergence for the 8-GHz unit sphere when MLFMM is applied with accuracy $\beta = 3$ in (13).

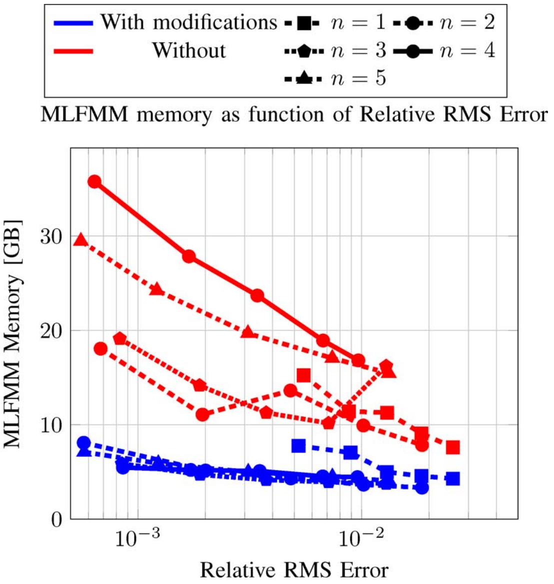

Fig. 4. MLFMM memory for varying relative RMS error and polynomial order n for accuracy $\beta = 3$ in (13), using CFIE, with and without the modifications.

to the CFIE with $\alpha = 0.5$, calculate the scattered field in the E-plane, and compare it to the Mie series. The results are shown in Fig. 3 as a function of the mesh size for each order.

The key observation from Fig. 3 is the increasing slope of the curves as the order of the basis functions increases. This corresponds to the theoretical estimate [27] of the discretization error behaving as $\mathcal{O}(h^n)$, where h is the mesh spacing and n is the polynomial order. We note that this is to our knowledge the first time such behavior has been demonstrated with MLFMM for hierarchical basis functions and for orders larger than 3. This demonstrates that there is no additional error when using MLFMM with HO rather than LO basis functions.

Fig. 4 demonstrates the required memory, as a function of relative RMS error, for varying polynomial order. While noting the nontrivial memory behavior as opposed to what we see with standard MoM, we also see that polynomial orders above 2 are perfectly competitive with lower orders in terms of memory. The lowest order $n = 1$ leads to a significantly higher memory consumption for a given relative RMS error, while the memory consumption for $n \geq 2$ is roughly independent of the polynomial order, except for high accuracies where $n = 2$ is not competitive. This behavior differs significantly from previously published conclusions [13], [22]–[24], where the conclusion was that $n > 2$ was not worthwhile.

The reduced memory consumption is due to the modifications described in Section III-B. This is clearly demonstrated by comparing the blue and the red set of curves that show the memory used with and without the modifications, respectively. The modifications result in a greater portion of the memory being spent in the group radiation patterns and the near-interaction matrix. The memory for storing group radiation patterns is far more significant for LO discretizations due to the vastly increased number of groups, while the near-interaction matrix is more significant for HO discretizations due to the larger group size at the lowest level. This is also evident in the plot, since the relative reduction in memory is much larger for the higher orders, although there is still a noticeable reduction for first order. We mention that the sidelength of the patches at the coarsest accuracy for order 5 is 1.6λ. Considering the low amount of memory used, these are far larger patches than we have seen used successfully in other higher order MLFMM implementations.

We note that the results from this test case, along with results from other experiments not shown here, indicate that the "best" order in terms of memory is very hard to quantify in general, because it depends on the geometry of the problem at hand as well as the desired error. Thus, we caution against recommending a specific order for all problems based on the results shown here, since these are based on a very theoretical case. Indeed, the almost uniformly meshed sphere, with no exterior edges and fixed polynomial order everywhere is not a common problem in practice.

Instead, we emphasize that, contrary to previously reported results, there is no significant penalty in terms of memory when increasing the order of the basis functions, provided that one applies the modifications described in the present paper. Indeed, increasing the order can be very beneficial in terms of memory, particularly when going for solutions with low relative RMS error.

Fig. 5 shows the most important feature of HO MLFMM—a significantly reduced time per iteration as the order is increased. This is the result of HO MLFMM having fewer levels and fewer unknowns, while also representing a larger portion of the interaction in the near-interaction matrix. Furthermore, the two sets of curves (in blue and red) allow us to gauge the time cost of the modifications. Overall, the increase in computational cost is roughly 20%, except for order 4 which fits very poorly with the mesh, resulting in a slightly larger cost for the interpolation involved in adaptive grouping. We note that the SHE could be used to render the interpolation step in adaptive grouping unnecessary, which would reduce the computational cost of adaptive grouping significantly.

The conclusion from Figs. 4 and 5, along with other experiments not shown here, is thus clear: Use as high an order as possible, since this significantly reduces the iteration time, while requiring roughly the same memory as lower orders.

TABLE I
Memory Use for Test Case B. The Accuracy Is $\beta = 3$ in (13), and We Have Used One Buffer Box Except for Order 1, Where We Needed to Use Two Boxes to Get Sufficient Accuracy. MLFMM Memory Includes the Temporary Data and Thus Exceeds the Sum of the Main Contributors Listed Here

Order	Mesh Size [λ]	Unknowns	Near matrix [GB] Indices	Near matrix [GB] Values	Bas. Patterns [GB]	Translation [GB]	Group patterns [GB]	MLFMM [GB]	MLFMM Levels	Time pr. Iter Normalized
1	0.20	2927300	0.37	7.22	1.42	1.27	10.42	23.64	8	3.92
2	0.49	1889496	0.15	2.94	1.45	1.09	6.73	14.49	7	1.85
3	0.97	1077948	0.09	3.62	1.33	1.08	4.76	12.97	6	1.37
4	1.52	784576	0.05	1.96	1.50	1.08	4.63	11.32	6	1.35
5	2.38	494340	0.04	2.99	1.55	1.08	3.49	11.21	5	1.05

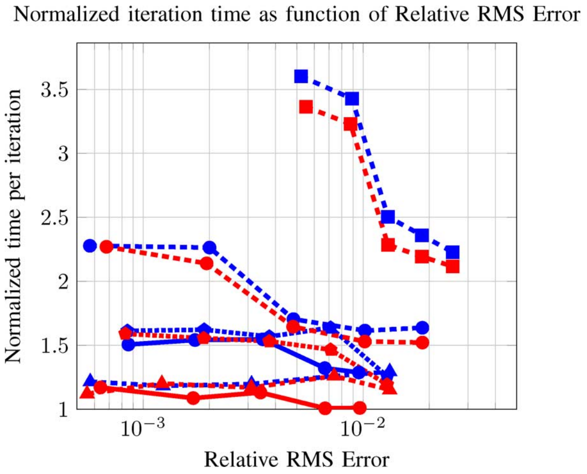

Fig. 5. Time per iteration, normalized to 1 for the fastest run, for the solutions shown in Fig. 4. We see a direct connection between increasing the order and reducing the time pr. iteration. We also see that the cost of adding the modifications, as compared to the standard MLFMM implementation, is noticeable, representing a roughly 20% increase. The legend is as in Fig. 4.

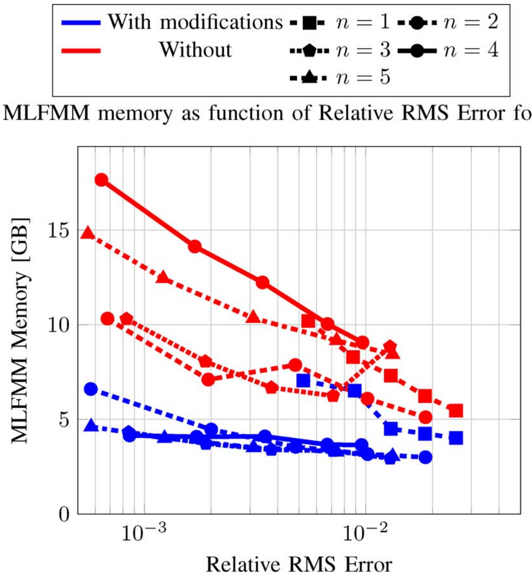

Fig. 6. MLFMM memory for varying relative RMS error and polynomial order for $\beta = 3$ for EFIE.

We also bear in mind that we are using CFIE on a PEC scatterer, which doubles the number of basis function patterns and breaks the symmetry in the near-interaction matrix that would come from using EFIE. This is a greater disadvantage for HO than for LO, because a larger proportion of the memory is spent in the near-field matrix and on storing basis function patterns for HO, while LO uses a significant amount of memory on storing the group patterns, which is unaffected by switching between EFIE and CFIE. To illustrate the memory behavior in EFIE, Fig. 6 is the equivalent of Fig. 4 but with the memory used in EFIE. Again, while the “best order” will depend on the scatterer and desired accuracy, it is clear that going beyond second order is indeed beneficial, reducing both memory requirements and computation time.

B. Disk

The second example involves a PEC disk, located at $z = 0$ with a radius of 1 m, illuminated by an $\hat{\boldsymbol{x}}$-polarized, $+z$ propagating plane wave at a frequency of 30 GHz and solved with MLFMM applied to the EFIE. This rotationally symmetric test case allows the use of the body of revolution method of moments (BoR-MoM) code [34] implemented in GRASP [35] as a very accurate reference solution.

Having chosen the mesh size such that the solution for each polynomial order yields roughly 0.3% relative RMS error, Table I shows the memory costs and time per iteration for each order. We see that it is clearly advantageous going above second order—all the way to order 5, both memory and computation time are reduced compared to both first and second order. We also see that the reduction in number of unknowns as the polynomial order is increased is now more significant than it was for the sphere. Thus, the higher order basis functions are relatively more advantageous than for the first case. This is due to the presence of edges, where the behavior of the true current is hard to model accurately using lower order polynomials due to its singularity.

To illustrate the savings achieved by the adaptive grouping and spherical harmonics techniques, Table II shows the memory for the basis function patterns and group patterns without these techniques and with only the adaptive grouping. The numbers without SHE take into account the symmetry of the basis function patterns. We see that both techniques are instrumental, particularly for HO implementations. The saving achieved with adaptive grouping depends significantly upon the initial grouping achieved by the Octree, but even for the lower orders, the saving is significant. For SHE, the savings are much larger; roughly a factor of 4–5. We note that both of these techniques require additional group storage at the lowest level, but we see that particularly for HO implementations, this is negligible.

Regarding the computation time, we see that the cost of the adaptive technique is negligible for HO, since it trades the cost of an extra level of interpolation/anterpolation with a faster aggregation/disaggregation step. For the SHE, there is a roughly

TABLE II
MEMORY USE FOR VARIOUS STRATEGIES—THE COST FOR ADAPTIVE, SHE STRATEGY CAN BE SEEN IN TABLE I. TIMES ARE NORMALIZED

Order	No Adaptive, No SHE			Adaptive, No SHE		
	Bas. Patterns [GB]	Group Patterns [GB]	Time pr. Iter	Bas. Patterns [GB]	Group Patterns [GB]	Time pr. Iter
1	12.56	7.69	3.36	4.50	9.94	3.89
2	13.68	5.29	1.65	4.99	6.12	1.87
3	16.78	3.88	1.31	5.17	4.31	1.38
4	11.45	3.87	1.22	6.28	4.18	1.26
5	20.35	2.89	1.01	6.95	3.12	1.00

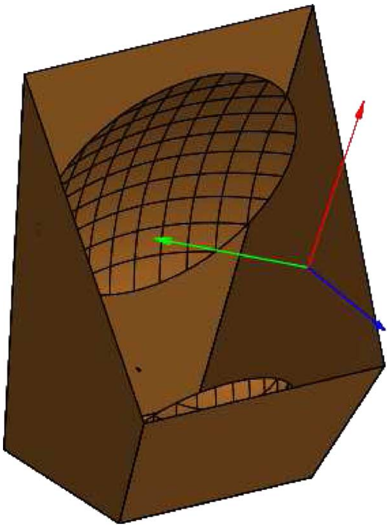

Fig. 7. Illustration of the Planck telescope mock-up model, looking into the shielding structure. A small, black line indicates the location of the feed, while the red, green, and blue axes indicate $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, respectively.

5% time increase for HO, since the cost of (25) and conversion between components is traded for a much faster aggregation and disaggregation. All-in-all, we believe that these results demonstrate that any modern HO implementation of MLFMM should apply both adaptive grouping and SHE, since the massive memory reduction comes at a very minor computational cost.

C. Planck Mock-Up

As a final example, we compute the radiation pattern of a simple mock-up of the Planck space telescope [36]. The main part of the telescope features two large reflectors in a dual reflector configuration, inside a shielding structure. The structure is shown in Fig. 7. The ellipsoidal subreflector, at the bottom of the figure, is fed by a horn located at the bottom of the shielding structure.

The resulting mesh at 30 GHz is quite irregular, featuring 126 596 patches with sidelengths between 0.34λ and 2λ and a surface area of $170 \cdot 10^3\lambda^2$. This results in 4 583 755 unknowns, with up to ninth-order polynomials used on the largest patches. A typical RWG discretization would require roughly 20 million unknowns.

The required MLFMM memory for HO MLFMM is 39.5 GB, and the problem converges in 148 iterations using a preconditioner requiring an additional 9 GB of memory. The scattered field is shown in Fig. 8.

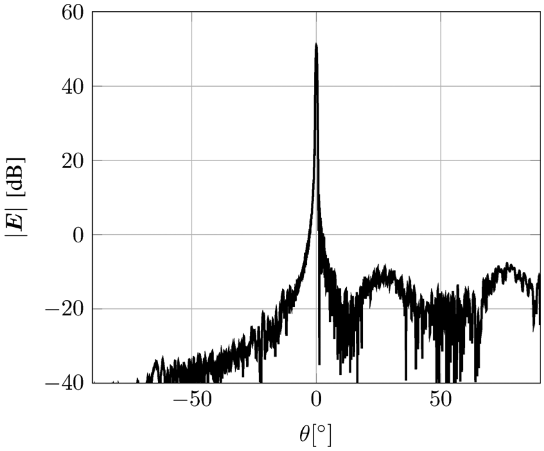

Fig. 8. Total copolar field for $\phi = 0$ from the mock-up of the Planck space telescope.

V. CONCLUSION

The results in this paper clearly show that the advantages of higher order basis functions and MLFMM can be combined, provided some additional modifications to the standard MLFMM setup are implemented. We stress that all test cases have employed Legendre basis functions—had we used RWG for the LO solutions, the comparison would have been even more beneficial to HO MLFMM, since RWG basis functions require a much greater basis function density [26]. We further note that while the exact memory requirements obviously depend on the scatterer, we have clearly demonstrated that HO MLFMM is advantageous both in terms of memory and particularly in terms of speed. Thus, although the optimal order cannot be generally determined since it depends strongly on the required accuracy and the geometry of the scatterer, we have demonstrated that using higher order MLFMM results in significantly better performance than LO MLFMM.

ACKNOWLEDGMENT

The authors would like to thank Dr. S. Järvenpää, Department of Radio Science and Engineering at Aalto University, Finland, and Prof. T. Eibert, Lehrstuhl für Hochfrequenztechnik, Technische Universität München, for helpful advice.

REFERENCES

[1] R. F. Harrington, "Matrix methods for field problems," *Proc. IEEE*, vol. 55, no. 2, pp. 136–149, Feb. 1967.
[2] J. Shaeffer, "Direct solve of electrically large integral equations for problem sizes to 1 M unknowns," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2306–2313, Aug. 2008.

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

[4] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 329–342, Mar. 1997.

[5] S. M. Wandzura, "Electric current basis functions for curved surfaces," *Electromagnetics*, vol. 12, no. 1, pp. 77–91, Jan. 1992.

[6] E. Jørgensen, J. Volakis, P. Meincke, and O. Breinbjerg, "Higher order hierarchical Legendre basis functions for electromagnetic modeling," *IEEE Trans. Antennas Propag.*, vol. 52, no. 11, pp. 2985–2995, Nov. 2004.

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

[8] V. Rokhlin, "Diagonal forms of translation operators for Helmholtz equation in three dimensions,' Yale Univ., New Haven, CT, USA, 1992, Tech. Rep..

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

[10] C.-C. Lu and W. C. Chew, "A multilevel algorithm for solving a boundary integral equation of wave scattering," *Microw. Opt. Technol. Lett.*, vol. 7, no. 10, pp. 466–470, Jul. 1994.

[11] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microw. Opt. Technol. Lett.*, vol. 10, no. 1, pp. 14–19, Sep. 1995.

[12] J. M. 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.

[13] 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 Trans. Antennas Propag.*, vol. 49, no. 7, pp. 1069–1078, Jul. 2001.

[14] K. C. Donepudi, J. M. Song, J.-M. Jin, G. Kang, and W. C. Chew, "A novel implementation of multilevel fast multipole algorithm for higher order Galerkin's method," *IEEE Trans. Antennas Propag.*, vol. 48, no. 8, pp. 1192–1197, Aug. 2000.

[15] K. C. Donepudi, J.-M. Jin, and W. C. Chew, "A grid-robust higher-order multilevel fast multipole algorithm for analysis of 3-D scatterers," *Electromagnetics*, vol. 23, no. 4, pp. 315–330, 2003.

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

[17] Ismatullah and T. F. Eibert, "Surface integral equation solutions by hierarchical vector basis functions and spherical harmonics based multilevel fast multipole method," *IEEE Trans. Antennas Propag.*, vol. 57, no. 7, pp. 2084–2093, Jul. 2009.

[18] Ismatullah and T. F. Eibert, "Higher order surface integral equation solutions employing spherical harmonics based multilevel fast multipole method," in *Proc. IEEE Antennas Propag. Soc. Int. Symp. APSURSI'09*, 2009, pp. 1–4.

[19] Ismatullah, "Analysis of space-borne antennas by higher-order method of moments and inverse equivalent current methods," Ph.D. dissertation, Technical Univ. of Munich, Munich, Germany, Apr. 2010.

[20] X.-M. Pan, L. Cai, and X.-Q. Sheng, "An efficient high order multilevel fast multipole algorithm for electromagnetic scattering analysis," *Progress In Electromagn. Res.*, vol. 126, pp. 85–100, 2012.

[21] X. Q. Sheng, M. L. Yang, X. M. Pan, and C. Q. Deng, "Parallel higher-order FE-BI-MLFMA for 3D scattering," in *Proc. Int. Workshop Antenna Technol. (iWAT)*, 2011, pp. 66–69.

[22] S. Wang, Y. Su, and X. Guan, "Applying fast multipole method to higher-order hierarchical Legendre basis functions in electromagnetic scattering," *IET Microw., Antennas, Propag.*, vol. 2, no. 1, pp. 6–9, 2008.

[23] B. M. Kolundzija and D. Sumic, "Multilevel fast multipole method for higher order basis functions implemented in WIPL-D Pro," in *Proc. Eur. Conf. Antennas Propag.* , 2009, pp. 2136–2140.

[24] B. M. Kolundzija, M. S. Pavlovic, and B. Mrdakovic, "Optimum choice of currents' expansion order in MLFMM algorithm for electromagnetic scattering," in *Proc. IEEE Antennas Propag. Soc. Int. Symp. (APSURSI)*, 2009, pp. 1–4.

[25] J. R. Mautz and R. F. Harrington, "H-Field, E-field, combined field solutions for bodies of revolution," Dept. Electr. Comput. Eng.. Syracuse Univ., Syracuse, NY, USA, Mar. 1977.

[26] E. Jørgensen, "Higher-order integral equation methods in computational electromagnetics," Ph.D. dissertation, Technical Univ. Denmark, Lyngby, Denmark, May 2003.

[27] J.-C. Nédélec, "Mixed finite elements in R3," *Numer. Math.*, vol. 35, no. 3, pp. 315–341, Sep. 1980.

[28] D. Meagher, "Geometric modeling using octree encoding," *Comput. Graphics Image Process.*, vol. 19, no. 2, pp. 129–147, Jun. 1982.

[29] J. Song and W. C. Chew, "Error Analysis for the Truncation of Multipole Expansion of Vector Green's Functions," *IEEE Microw. Wireless Compon. Lett.*, vol. 11, no. 7, pp. 311–313, Jul. 2001.

[30] M. L. Hastriter, S. Ohnuki, and W. C. Chew, "Error control of the translation operator in 3D MLFMA," *Microw. Opt. Technol. Lett.*, vol. 37, no. 3, pp. 184–188, May 2003.

[31] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: Is it that big?," *IEEE Antennas Propag. Mag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.

[32] W. C. Gibson*, The Method of Moments in Electromagnetics*. Boca Raton, FL, USA: Chapman & Hall/CRC, Oct. 2007.

[33] 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 Finite Elements in Microw. Eng.*, Chios, Greece, May 2002, p. 63.

[34] E. Jørgensen and P. Meincke, "Design tool for high-performance rotationally symmetric reflector antennas," in *Proc. ISAP'12*, 2012, pp. 1381–1384.

[35] K. Pontoppidan*, GRASP Technical Description*. Copenhagen, Denmark: TICRA, Mar. 2008.

[36] J. A. Tauber, H. U. Norgaard-Nielsen, P. A. R. Ade, J. Amiri Parian, T. Banos, M. Bersanelli, C. Burigana, A. Chamballu, D. de Chambure, P. R. Christensen, O. Corre, A. Cozzani, B. Crill, G. Crone, O. D'Arcangelo, R. Daddato, D. Doyle, D. Dubruel, G. Forma, R. Hills, K. Huffenberger, A. H. Jaffe, N. Jessen, P. Kletzkine, J. M. Lamarre, J. P. Leahy, Y. Longval, P. de Maagt, B. Maffei, N. Mandolesi, J. Martí-Canales, A. Martín-Polegre, P. Martin, L. Mendes, J. A. Murphy, P. Nielsen, F. Noviello, M. Paquay, T. Peacocke, N. Ponthieu, K. Pontoppidan, I. Ristorcelli, J. B. Riti, L. Rolo, C. Rosset, M. Sandri, G. Savini, R. Sudiwala, M. Tristram, L. Valenziano, M. van der Vorst, K. van 't Klooster, F. Villa, and V. Yurchenko, "Planck pre-launch status: The optical system," *Astron. Astrophys.*, vol. 520, p. A2, Sep. 2010.

Oscar Borries (S'13) was born in Gentofte, Denmark, on November 4, 1986. He received the B.Sc. and M.Sc. degrees in mathematical engineering from the Technical University of Denmark (DTU), Lyngby, Denmark, in 2009 and 2011, respectively, where he is currently working towards the Ph.D. degree.

His research interests include fast methods and integral equation methods, particularly higher order methods.

Peter Meincke (S'93–M'96) was born in Roskilde, Denmark, on November 25, 1969. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Technical University of Denmark (DTU), Lyngby, Denmark, in 1993 and 1996, respectively.

In spring and summer 1995, he was a Visiting Research Scientist at the Electromagnetics Directorate of Rome Laboratory, Hanscom Air Force Base, MA. In 1997, he was with a Danish cellular phone company, working on theoretical aspects of radio-wave propagation. In spring and summer of 1998, he was visiting the Center for Electromagnetics Research, Northeastern University, Boston, MA, USA, while holding a Postdoctoral position from DTU. In 1999, he became a staff member of the Department of Electromagnetic Systems, DTU. He was an Associate Professor with Ørsted-DTU, Electromagnetic Systems, DTU, where his teaching and research interests included electromagnetic theory, inverse problems, high-frequency and time-domain scattering, antenna theory, and microwave imaging. Since 2008, he has been with TICRA, Copenhagen, Denmark, where he is currently developing software for reflector antenna analysis and antenna diagnostics.

Dr. Meincke won the first prize award in the 1996 IEEE Antennas and Propagation Society Student Paper Contest in Baltimore, MD, for his paper on uniform physical theory of diffraction equivalent edge currents and received the R.W.P. King Paper Award in 2000 for his paper "Time-domain version of the physical theory of diffraction" published in the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, February, 1999.

Erik Jørgensen (S'98–M'03) was born in 1974 in Denmark. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Technical University of Denmark (DTU), Lyngby, Denmark, in 2000 and 2003, respectively.

In fall 1998, he spent five months at the University of Siena, Italy. In fall 2001, he was a Visiting Scholar at the Radiation Laboratory, University of Michigan, Ann Arbor, MI, USA. In 2003, he joined the Danish company TICRA, Copenhagen, where he is now leading the development of commercial software for reflector antenna analysis and antenna diagnostics. His research interests include computational electromagnetics and high-frequency techniques.

Per Christian Hansen received the M.Sc. degree in electrical engineering in 1982, the Ph.D. degree in numerical analysis in 1985, and the Dr.Techn. degree in scientific computing in 1996, all from the Technical University of Denmark (DTU), Lyngby, Denmark.

He was a Research Associate at Copenhagen University from 1985 to 1988, and senior consultant at UNI●C from 1988 to 1996. Since 1996, he has been a Professor of scientific computing at DTU Compute, the Technical University of Denmark. He primarily works with numerical regularization algorithms for inverse problems, and he has published 80+ papers in leading journals as well as three books on numerical methods for inverse problems. He has also developed several related software packages.