An Open Source Implementation for Full-wave 2D Scattering at Million Wavelength Size Objects

Jan Fostier, Femke Olyslager, *Fellow, IEEE*

Abstract

In this contribution, we demonstrate that recent improvements in 'fast methods' allow for fully error-controlled, full-wave simulations of two-dimensional objects with sizes over a million wavelengths on relatively simple computing environments. We review how a fully scalable parallel version of the Multilevel Fast Multipole Algorithm (MLFMA) is obtained to accelerate a two-dimensional boundary integral equation for the scattering at multiple large dielectric and/or perfectly conducting objects. Several complex and large scale examples demonstrate the capabilities of the algorithm. Its implementation is available as open source under GPL license (http://www.openfmm.net).

Index Terms

Electromagnetic scattering, Multilevel Fast Multipole Algorithm (MLFMA), scalable asynchronous parallelization.

I. Introduction

Since their first application to electromagnetic wave propagation problems, integral equations have remained one of the main techniques in computational electromagnetics [1], [2]. In particular for the scattering at homogeneous objects, the use of boundary integral equations is attractive

Manuscript received January 7, 2009;

The authors are with the Department of Information Technology (INTEC), Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium. Contact person: Daniël De Zutter (daniel.dezutter@intec.ugent.be).

This work was supported by an SBO grant No. 231/01114 from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). The work of J. Fostier was supported by a doctoral grant from the same institution.

because only the discretization of surfaces is needed, reducing the dimensionality of the problem. Another aspect that has added to the success of the integral equation approach is its accuracy and error controllability. The price to be paid for these properties is that the method is strongly analytically involved and that a dense linear system of equations needs to be solved.

It is especially the solution of this dense system of equations that has plagued integral equation techniques for a long time. Some of the consequences for the development of the method are summarized. Commercial applications of the method have for long been limited to the simulation of planar circuits embedded in layered media of infinite extent. By using a Green function of the layered medium, the discretization could be limited to the metallization, in that sense reducing the number of unknowns of the system dramatically [3]. Considerable attention to the use of special basis functions, either to reduce the number of unknowns or to sparsify the linear system of equations [4], [5]. For volume integral equations, the bottleneck associated with the size of the linear system is even more limiting.

The pressure on the size of the system for *volume* integral equations led, on the one hand, to the use of iterative, conjugate gradient based techniques to solve the linear system. On the other hand, the matrix vector product that needs to be evaluated in each iterative step, was physically interpreted as the calculation of the fields generated by a large number of sources, in a large number of observation points. If these source and observation points were placed on a regular lattice, these convolution type of evaluations could be drastically accelerated by using Fast Fourier Transform (FFT) techniques [6].

What was possible for volume integral equations was not possible for surface integral equations, because the lack of a regular lattice prevents the use of FFTs. In many particle methods, the Fast Multipole Method (FMM) was a major breakthrough to evaluate the Coulomb interactions between large sets of particles [7]. Particles were grouped and the field generated by such a group could be represented by only a limited number of multipoles. By hierarchically grouping these groups in larger groups, the computational cost of the evaluation of all interactions between the particles could be made proportional to the number of particles. Although this method can be applied for electromagnetic wave problems, it quickly loses efficiency when considering problems that extend over a few wavelengths. Later, Rokhlin extended the FMM to Helmholtz problems [8].

The major breakthrough came when a Multilevel Fast Multipole Algorithm (MLFMA) was de-

vised that allowed to compute the fields due to N sources in N observation points in $O(N \log N)$ operations, no matter the size of the ensemble of sources [9]. The MLFMA exploits an addition theorem for the Green kernel in the integral equation that decomposes the field of a group of sources in plane waves. The MLFMA also has as property that it is fully error controllable. For a good introduction to the MLFMA we refer to [10].

Although not changing its basis philosophy, the MLFMA has seen considerable improvements since its conception. Let us mention a few of these improvements, all of which are still subject of ongoing research. The numerical evaluation of the addition theorem starts to fail when it is used to represent propagation over distances that become small compared to wavelength, the so-called low-frequency breakdown of the method. Several methods have been proposed to remedy this problem [11], [12], [13]. Another issue is the convergence of the iterative solution of the system of linear equations, which is governed by the condition number of the coefficient matrix. Several preconditioning techniques have been developed to dramatically improve the condition number and reduce the number of iterations [14], [15], [16]. Advances in computing power come from the use of multiple cores or processors in a cluster environment [17]. Optimal use of the capabilities of these systems requires the development of parallel algorithms. In the realm of the MLFMA, several of these parallelization schemes have been the subject of recent intensive research [18], [19], [20], [21], [22], [23], [24], [25].

In this paper, we will review and demonstrate the capabilities of a general purpose two-dimensional boundary integral equation that is accelerated using the MLFMA. The algorithm is parallelized such that it is scalable [26] and asynchronous [25]. The term *scalable* means that the computational time is proportional to the logarithm of the number of unknowns in the problem, when the number of unknowns is proportional to the number of processors. The algorithm is asynchronous since at a given point in time, not all processors are doing the same type of computations. This makes the algorithm more robust for imperfect load balancing over the processors and avoids communication bursts, allowing the use of slower communication networks between the processors.

First, we will briefly review the boundary integral equation that is used and how it is discretized with the Method of Moments (MoM). This boundary integral equation is a two-dimensional version of the PMCHWT integral equation (Poggio, Miller, Chang, Harrington, Wu, Tsai) [27], [28], [29]. Special attention is devoted to the self- and neighbor-patch integrations. Next, we will

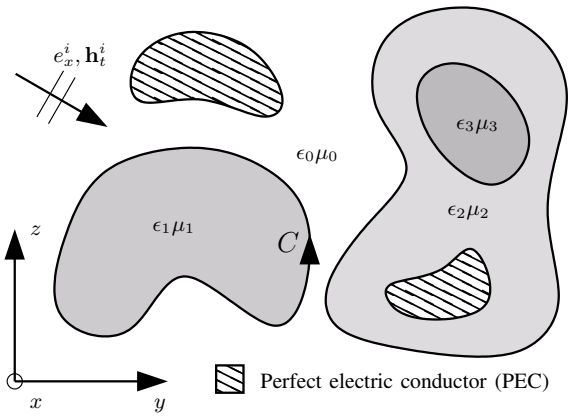


Fig. 1. Objects illuminated by an incoming electromagnetic field.

discuss the MLFMA and how the plane wave spectrum can be separated in different parts, the so-called k-space partitioning. This partitioning is needed to obtain a scalable parallel algorithm as explained in Section IV. In that section, the asynchronous parallelization is also outlined. An extensive example section demonstrates the capabilities of the algorithm and its implementation. Scattering problems with a size up to two million wavelengths or comprising more than 70 million unknowns are solved efficiently on cheap computer clusters [30]. The implementation of the algorithm that was used to calculate these examples is available as open source under GPL license (http://www.openfmm.net).

Although some parts of our methods have been published in detail in a number of international publications [23], [25], [30], the aim of this paper is to provide a tutorial overview of all the pieces in the jigsaw puzzle and to provide the interested reader with the necessary background to fully exploit the capabilities of the online open source implementation of the algorithm.

II. Boundary Integral Equation

We consider a two-dimensional scattering problem under TM illumination with an $e^{j\omega t}$ time-harmonic dependence. The geometry can consist of multiple dielectric (with material parameters ϵ_i and μ_i) and/or perfect electric conducting (PEC) cylindrical objects, parallel to the x-axis and embedded in free space with parameters ϵ_0 and μ_0. Objects can be embedded into other dielectric objects (see Fig. 1). The boundary curve of an object is denoted by C and e_x^i, $\mathbf{h}_t^i$ represents the incoming field where the subscript 't' indicates vectors in the yz-plane. For dielectric objects, this

scattering problem is solved using a boundary integral equation (BIE) that is a two-dimensional version of the PMCHWT integral equation. It was first proposed to analyze waveguides in [31], [32], [33] and for the TM problem it takes the following form:

$$\begin{aligned} e_x^i &- \lim_{\mathbf{r}\to C+} \oint_C [e_x \frac{\partial G_0}{\partial n'} - \frac{jk_0^2}{\omega\epsilon_0} G_0 h_t] dc' \\ &= \lim_{\mathbf{r}\to C-} \oint_C [e_x \frac{\partial G}{\partial n'} - \frac{jk^2}{\omega\epsilon} G h_t] dc' \qquad (1) \\ h_t^i &- \lim_{\mathbf{r}\to C+} \oint_C [-\frac{j\omega\epsilon}{k_0^2} e_x \frac{\partial^2 G_0}{\partial n \partial n'} - \frac{\partial G_0}{\partial n} h_t] dc' \\ &= \lim_{\mathbf{r}\to C-} \oint_C [-\frac{j\omega\epsilon}{k^2} e_x \frac{\partial^2 G}{\partial n \partial n'} - \frac{\partial G}{\partial n} h_t] dc' \qquad (2) \end{aligned}$$

with $k^2 = \omega^2\epsilon\mu$ and the Green function

$$G(\mathbf{r}|\mathbf{r}') = \frac{j}{4} H_0^{(2)}(k|\mathbf{r} - \mathbf{r}'|), \qquad (3)$$

and similarly for k_0^2 and G_0. C^- and C^+ denote that the contour C is approached from the inside and outside respectively. The unknowns are the tangential electric field component e_x and the tangential magnetic field component h_t to the contour C. The first equation imposes the continuity of the total tangential electric field and the second equation the continuity of the total tangential magnetic field at the contour C. An extension to multiple object scattering is trivial and the TE problem follows from duality. A further extension to problems where the fields have an $e^{-j\beta x}$ dependence, leading to a coupling between TE and TM, has been studied in [34]. For PEC objects, the vanishing of the tangential component of the electric field is expressed at the boundary.

The BIE is discretized using a Galerkin MoM scheme that was first proposed in [31], [32], [33]. The boundary C is divided into segments and for the h_t unknown a piecewise constant approximation is chosen using pulse functions and for the e_x unknown, a piecewise linear approximation is chosen using overlapping triangular functions. The length of the segments typically is one tenth of a wavelength. The first equation (1) is tested using the pulse functions and the second equation (2) is tested using the triangular functions. The pulse functions have a height equal to their inverse lengths and the triangular functions have a height equal to the inverse of the average lengths of its two supporting segments.

In order to obtain a high accuracy, the singular part from the Green function and its derivatives are extracted in some cases. The basis and test function integrations for this singular part are evaluated analytically for self-patch integrations (i.e. the basis and test functions are defined over the same segment) and for neighbor-patch integrations (i.e. the basis and test functions are defined over adjacent segments). We extract the following singular parts

$$\begin{aligned}
\left[G(\mathbf{r}|\mathbf{r}')\right]_{\text{sing}} &= \frac{1}{2\pi}\log(k|\mathbf{r}-\mathbf{r}'|), && (4)\\
\left[\frac{\partial}{\partial n'}G(\mathbf{r}|\mathbf{r}')\right]_{\text{sing}} &= -\frac{1}{2\pi}\frac{\mathbf{n}'\cdot(\mathbf{r}-\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|^2}, && (5)\\
\left[\frac{\partial^2}{\partial n\partial n'}G(\mathbf{r}|\mathbf{r}')\right]_{\text{sing}} &= \frac{k^2}{4\pi}(\mathbf{n}\cdot\mathbf{n}')\log(k|\mathbf{r}-\mathbf{r}'|)\\
&+\frac{1}{\pi}\left[\frac{(\mathbf{n}\cdot(\mathbf{r}-\mathbf{r}'))(\mathbf{n}'\cdot(\mathbf{r}-\mathbf{r}'))}{|\mathbf{r}-\mathbf{r}'|^2}-\frac{\mathbf{n}\cdot\mathbf{n}'}{2}\right]\frac{1}{|\mathbf{r}-\mathbf{r}'|^2}. && (6)
\end{aligned}$$

The analytical integrations over the singular part encountered when considering a pulse basis and pulse test function were treated in [23] for self- and neighbor-patches. Here, we will give a compact overview of all analytical integrations. For a pulse basis function and a pulse test function as shown in Fig. 2a, we have the following singular integral

$$\begin{aligned}
&\frac{1}{l_1 l_2}\int_0^{l_2}\int_0^{l_1}\left[G(\mathbf{r}(s)|\mathbf{r}'(s'))\right]_{\text{sing}}\mathrm{d}s'\mathrm{d}s =\\
&\frac{1}{2\pi}\left[\log k + f_1(l_1,l_2)+f_1(l_2,l_1)\right], && (7)
\end{aligned}$$

with f_i, $i=1,2,...,11$, defined in Appendix A. Carefully taking the limits $\phi \to 0$ and $l_1 \to l_2$ leads to the self-patch contributions. For a pulse basis function and a test triangular function (or vice versa), we obtain for the situation shown in Fig. 2b the following integrals

$$\begin{aligned}
&\frac{1}{l_1 l_2}\int_0^{l_2}\int_0^{l_1}(l_1-s')\left[\frac{\partial}{\partial n'}G(\mathbf{r}(s)|\mathbf{r}'(s'))\right]_{\text{sing}}\mathrm{d}s'\mathrm{d}s =\\
&\frac{1}{2\pi l_1 l_2}\left[f_2(l_1,l_2)+f_3(l_1,l_2)-f_3(l_2,l_1)\right], && (8)
\end{aligned}$$

and for the other situation shown in Fig. 2c

$$\frac{1}{l_1 l_2}\int_0^{l_2}\int_0^{l_1}s'\left[\frac{\partial}{\partial n'}G(\mathbf{r}(s)|\mathbf{r}'(s'))\right]_{\text{sing}}\mathrm{d}s'\mathrm{d}s =$$

$$\frac{1}{2\pi l_1 l_2}\left[f_4(l_1,l_2)+f_4(l_2,l_1)\right]. \tag{9}$$

The situation of a triangular basis and test function is more complicated since the involved derivatives of the Green function (6) now both have a logarithmic and one over distance squared singularity. Three different situations need to be distinguished. The situation of Fig. 2d leads to

$$\begin{aligned}\frac{1}{l_1 l_2}\int_0^{l_2}\int_0^{l_1}(l_1-s')(l_2-s)[\frac{\partial^2}{\partial n\partial n'}G(\mathbf{r}(s)|\mathbf{r}'(s'))]_{\text{sing}}\mathrm{d}s'\mathrm{d}s=\\ -\frac{k^2\cos\phi}{4\pi l_1 l_2}\left[\frac{1}{4}l_1^2 l_2^2\log k+f_5(l_1,l_2)+f_5(l_2,l_1)\right]\\ +\frac{1}{2\pi l_1 l_2}\left[f_6(l_1,l_2)+f_6(l_2,l_1)\right].\end{aligned} \tag{10}$$

The self-patch case is found by again taking the limit for $\phi\to 0$, together with a change of sign of the result. In equation (10), a singular contribution was omitted that will drop out when combining the contributions of a self-patch and neighbor-patch contribution of the form Fig. 2d. The integral needed for the situation of Fig. 2f can be cast in the same form as (10) by replacing f_5 and f_6 by f_7 and f_8 respectively. Finally, the situation of Fig. 2e requires the following integral

$$\begin{aligned}\frac{1}{l_1 l_2}\int_0^{l_2}\int_0^{l_1}s'(l_2-s)\left[\frac{\partial^2}{\partial n\partial n'}G(\mathbf{r}(s)|\mathbf{r}'(s'))\right]_{\text{sing}}\mathrm{d}s'\mathrm{d}s=\\ -\frac{k^2\cos\phi}{4\pi l_1 l_2}\left[\frac{1}{4}l_1^2 l_2^2\log k+f_9(l_1,l_2)+f_{10}(l_1,l_2)-\frac{1}{3}f_{10}(l_2,l_1)\right]\\ +\frac{1}{2\pi l_1 l_2}\left[-\frac{l_1 l_2}{2}+f_{11}(l_1,l_2)-f_{11}(l_2,l_1)\right].\end{aligned} \tag{11}$$

The remaining regular part of the Green function and its derivatives for self-patch and neighbor-patch integrations, as well as all other basis and test function integrations are evaluated using Gaussian quadrature rules.

The high accuracy of the basis and test function integrations is paramount to assure the convergence of the iterative solution of the MoM system when considering extremely large scale problems.

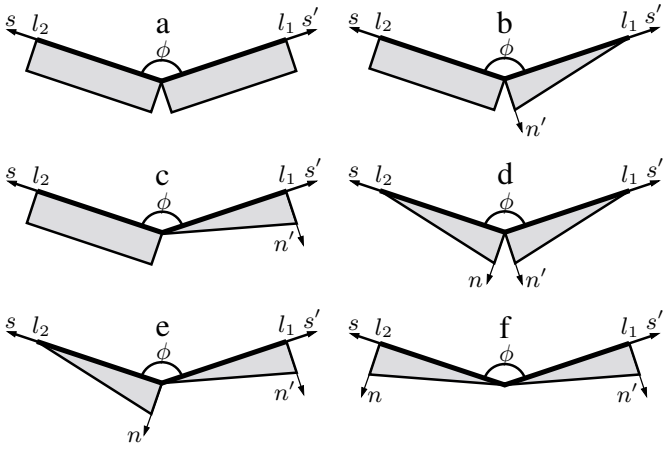


Fig. 2. Different cases of neighbor-patch evaluation.

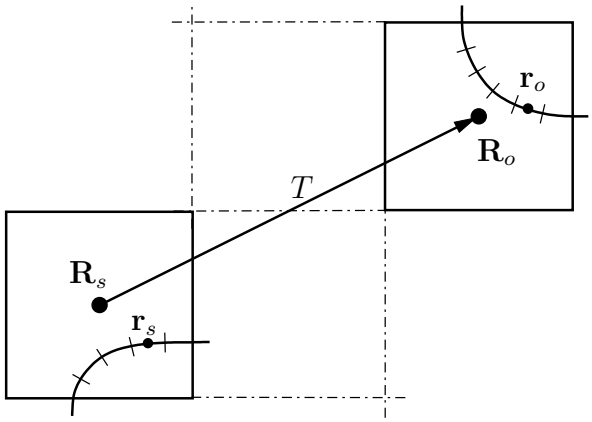


Fig. 3. Two interacting boxes in the MLFMA.

III. MLFMA

Since we consider scattering at objects that are large compared to the wavelength, we can suffice with the classical MLFMA that uses a plane wave decomposition [10]. However, when objects are considered with substantial sub-wavelength detail, it might be necessary to use a low-frequency extension of the MLFMA as described for example in [11]. The MLFMA evaluates the fields due to N sources at the position of these N sources. This corresponds to a matrix-vector multiplication in the iterative solution of the MoM linear system. First, the segments are grouped into an equidistant lattice of square boxes. The interaction between a source and an observation point is not calculated directly on an individual basis but indirectly by considering

the interaction of the source and observation box to which the source and observation points belong. This situation is depicted in Fig. 3. In order to accomplish this, the Green function is decomposed in a plane wave summation

$$G(\mathbf{r}_0|\mathbf{r}_s) = -\frac{j}{4}\sum_{n=-Q}^{n=Q}\left[e^{j\mathbf{k}_n\cdot(\mathbf{r}_s-\mathbf{R}_s)}T_n(\mathbf{R}_o-\mathbf{R}_s)e^{-j\mathbf{k}_n\cdot(\mathbf{r}_o-\mathbf{R}_o)}\right], \tag{12}$$

with $\mathbf{R}_o$ and $\mathbf{R}_s$ the centers of the observation and source boxes respectively, $\mathbf{r}_o$ and $\mathbf{r}_s$ the positions of the observation and source points respectively, the wave-vectors $\mathbf{k}_n = k(\cos\phi_n\mathbf{u}_y + \sin\phi_n\mathbf{u}_z)$ where $\phi_n = 2\pi n/(2Q+1)$, $n = -Q, ..., 0, ...Q$ and finally, $\mathbf{u}_y$, $\mathbf{u}_z$ the unit vectors in the y and z direction respectively. The translation operator is given by

$$T_n(\mathbf{R}) = \frac{1}{2Q+1}\sum_{n'=-Q}^{n'=Q} H_{n'}^{(2)}(kR)e^{jn'(\Phi-\phi_n-\frac{\pi}{2})}, \tag{13}$$

with R and Φ the polar coordinates of $\mathbf{R}$. The value of Q that determines the number of plane waves depends on the size of the box and on the requested accuracy. For small box sizes, an optimal value can be numerically searched; for higher levels, a good choice is given by the excess bandwidth formula [10].

The aggregation, that corresponds to the first part in (12), is an expansion in plane waves of the source field along the directions defined by $\mathbf{k}_n$ and depends only on the position of the source point in the source box. The translation, that corresponds to the second part in (12), depends only on the center positions of the source and observation box. Finally, the disaggregation, that corresponds to the third part in (12), adds the contribution of all plane waves at the observation point. The disaggregation only depends on the position of the observation point in the observation box. The interaction between a source and observation point is only calculated using this approach when the source and the observation box(es) are at least separated by one box.

To increase the efficiency of the method, boxes are hierarchically grouped in larger boxes by taking up to four non-empty boxes together and so on until one large box encompasses the entire structure. In this way a multilevel tree structure is obtained. Interactions between a source and observation point are evaluated using the largest possible boxes, i.e. as high as possible in the tree, without violating the rule that a buffer of one box between the observation and source box is needed. The value for Q increases when we go up in the tree because the size of the boxes increases. This means that the plane wave directions need to be sampled finer, i.e. interpolation

is necessary. The hierarchical multilevel scheme reduces the computational complexity of the matrix-vector product to $O(N \log N)$.

Note that at the lowest levels, the box-buffer criterion that is used to determine whether two boxes are sufficiently separated to use the plane wave decomposition (12) is stricter than for the higher levels. This is to assure a high accuracy, even when segments partly protrude from their box. A segment and all its discretization points belong to a box when the middle of that segment is contained by the box. In practice, a two-box-buffer limit is used for the lowest level. This decreases gradually to a one-box-buffer limit that is used for the fourth and higher levels. Also, note that only the plane wave expansion for the e_x field component is needed. Indeed, the magnetic field follows from $\mathbf{h}_t = (\mathbf{u}_x \times \nabla e_x)/(j\omega\mu)$.

To reduce the number of iterations, a block-Jacobi preconditioner is utilized. The blocks are chosen as the self-interactions of the boxes at a certain level in the tree. If a higher level is taken, the size of the blocks increases. Consequently, a more powerful preconditioner is obtained. However, the computational cost for setting up and applying the preconditioner is also higher. These self-interaction blocks are stored in LU-decomposed form to allow for a fast forward and inverse multiplication scheme, without any extra memory requirements.

IV. Parallelization

A. Scalability: hierarchical scheme

One of the most challenging problems related to the parallelization of the MLFMA is that of the scalability. A parallel algorithm is said to be scalable when a larger simulation can be handled on a proportionally larger parallel machine, without loss of efficiency. In other words, if the computational complexity of a serial implementation is $O(N \log N)$ and $P = O(N)$ denotes the number of processors, the memory, communication and computational requirements of each node should not exceed $O(\log N)$. It can be shown that the issue of scalability is closely related to the distribution of the workload across the different processes. In the case of a two-dimensional, high-frequency, boundary integral equation problem, the number of boxes decreases on average with a factor of two at every next level, while the number of sampling points roughly doubles. Therefore, each level in the MLFMA tree contains $O(N)$ sampling points in total. The lowest levels have $O(N)$ boxes each containing $O(1)$ (i.e. a constant number, independent from N) sampling points whereas the highest levels have $O(1)$ boxes with $O(N)$ sampling points.

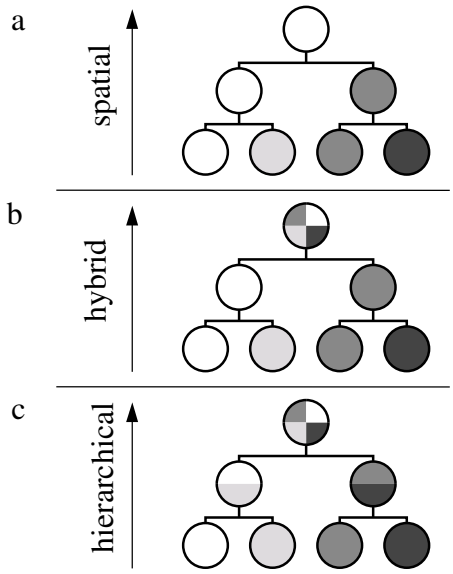


Fig. 4. Spatial (a), hybrid (b) and hierarchical (c) partitioning scheme for a three-level tree and four processes. The different shades of gray represent the distribution of data among the different processes.

The most simple partitioning approach is the distribution of the boxes containing their radiation patterns as whole (see Fig. 4a). Because this approach requires $O(N)$ memory and calculation time on the nodes that are attributed a top-level radiation pattern, spatial partitioning is not scalable. The actual number of processes that can be used depends on the speed of the interconnection network, the speed of the processors, the implementation details and the desired parallel efficiency that is to be obtained, but can never increase proportionally to the problem size.

A hybrid scheme was later proposed in [18] and is illustrated in Fig. 4b. At the lower levels, spatial partitioning is used, whereas at the top levels, the radiation pattern samples themselves are distributed among *all* processes. This is called *k*-space partitioning. When $P = O(N)$, only a constant number of top levels can be partitioned in this way, because at least $O(N)$ sampling points are needed. This means that spatial partitioning is still required for the $O(\log N)$ other levels, and that the hybrid approach does therefore not improve the scaling behavior over spatial partitioning, although the parallel efficiencies benefit [18], [35].

The hierarchical partitioning technique was recently introduced in [21]. At every level, $O(N/P)$ sampling points are allocated to each node. This is accomplished by a gradual transition from

spatial to *k*-space partitioning, as illustrated in Fig. 4c. Suppose that $P = O(N)$ with $P = 2^i$, $i \in \mathbb{N}$. At the lowest level, each node is allocated $O(N/P) = O(1)$ boxes with their $O(1)$ sampling points as a whole (spatial partitioning). At the next level, the boxes are shared between an even and its neighboring odd process, however, each node now holds only half of the radiation pattern samples. At the next level, the boxes are shared by four nodes, each containing only one fourth of the sampling points. This process continues until at the top level, the box is shared by all nodes, each node containing $1/P^{th}$ of the radiation pattern. The transition from spatial to *k*-space partitioning requires $\log_2(P)$ levels, which means that only the remaining $O(1)$ levels are used for spatial and *k*-space partitioning. For more information regarding this repartitioning process, we refer to [21], [22]. It is clear that every node contains $O(1)$ boxes and $O(1)$ samples at each level. Because the calculation time is proportional to the number of sampling points, it is also $O(1)$ per node and per level. It can be shown that the amount of communication per node and per level is also $O(1)$ [26], which means that this approach indeed allows for a number of processes $P = O(N)$. The hierarchical partitioning technique not only strongly reduces the amount of communication [22], it also reduces the number of communication events. Indeed, a certain node only has to communicate to $O(\log N)$ other nodes (i.e. $O(1)$ nodes per level).

These ideas can be extended to a three dimensional, high-frequency boundary integral equation problem. In that case, each box has on average four children and the number of sampling points increases by a factor of four at every next level. This means that the radiation patterns now need to be repartitioned into four sections instead of only two as was the case in two dimensions. This must be accomplished by a repartitioning in both θ and ϕ direction [26]. It is clear that such a two-dimensional partitioning leads to a complex implementation, however, one can easily prove that when using only a one-dimensional partitioning, in e.g. the ϕ-direction only, the number of processes is limited to only $P = O(\sqrt{N})$.

The use of the hierarchical partitioning technique requires a local interpolator. In this case, a Dirichlet kernel with a Gaussian taper is used [36] although an interpolator based on periodic approximate prolate spheroidal (APS) functions [37] is slightly more performant. In a practical situation, the number of processes P is much smaller than 2^L, with L the number of levels in the tree. This means that typically spatial partitioning is used for the first 6 or 7 levels, after which the hierarchical partitioning scheme is deployed. For these lowest levels, a global interpolator based on Fast Fourier Transforms (using the FFTW [38] package) is used because it is more

accurate and faster. For the remaining top levels, if any, full k-space partitioning is used.

B. Asynchronous algorithm

When considering geometries with multiple dielectric objects, each homogeneous region requires a different MLFMA tree for the evaluation of the interactions. It is favorable to perform a global distribution of the geometry. This means that large MLFMA trees (such as the background medium) are distributed among many nodes, whereas smaller MLFMA trees are distributed among fewer nodes, possibly even a single node. The recently introduced asynchronous algorithm can handle this kind of problems in a flexible way [25].

The workload for each node is divided into small homogeneous 'work packets'. Such a work packet can be a portion of the (dis)aggregations, translations or near interactions that need to be evaluated during each iteration. Work packets can be blocked because they rely on data that is to be received from another node or because another work packet needs to be evaluated first. All the work packets that are no longer blocked are contained in a priority queue which selects the most urgent one that is to be evaluated. The priority queue can contain work packets that belong to different homogeneous regions and hence allows the simultaneous processing of different MLFMA trees. The evaluation of a work packet and/or the receiving of data from another node can in turn add new work packets to the queue. The introduction of the hierarchical approach requires only small modifications to this scheme.

After the aggregation packets to a certain level are completed, a repartitioning occurs and half of the sampling points are scheduled for sending. Also, the sampling points near the edges of the local partition that are needed by the nodes that handle the adjacent partitions are scheduled for transmitting. Similarly, the receiving of these data is prepared. While waiting for it, the node can go ahead with local calculations because the work packets for the local translations and local aggregations can be added to the priority queue. By local aggregations, we mean this portion of the aggregations that depends upon sampling points that were calculated locally. In order to perform the aggregations accurately, the node also requires boundary radiation pattern sampling points from the nodes. These are handled in different work packets which are scheduled when they receive the necessary data. In other words, the (dis)aggregation at each level is split into three work packets: one for the local sampling points, one for the left boundary points and one for the right boundary points.

C. Implementation and limitations

The two parallelization techniques (hierarchical partitioning and asynchronous algorithm) have been combined into the existing C++ implementation. The AMOS [39] library is used for the calculation of special functions and the Parallel Iterative Methods (PIM) library provides for the conjugate gradient based routines. The Message Passing Interface (MPI) [40] was used for internode communication. In the current implementation, it is required that the number of processors used is a power of two when the hierarchical partitioning technique is used.

The first step of the implementation is a sequential step: each node constructs the global MLFMA tree for the geometry under consideration. The boxes at each level are ordered according to a Hilbert space filling curve, which is subsequently partitioned among the nodes. Also, the number of k-space partitions at each level are determined. This determines the load balancing between the nodes. At this point, the memory associated with the portion of the MLFMA tree that is not local to the node is released. After this first step, the further processing (i.e. setup of communication maps, aggregations, translations, disaggregations, iterative solution process and output calculation) is fully parallelized. This first sequential step requires only a few minutes, even for the very large problems with are shown in the next section. However, for extremely large-scale simulations, the memory per node might be limiting. Some implementations also provide a parallelized construction of the MLFMA tree, e.g. [18].

When considering simulations that contain smaller dielectric objects, an important remark is to be made. At the lowest levels, the MLFMA trees corresponding to these objects are allocated to only a few nodes. Using the hierarchical approach, more and more nodes get involved in the handling of the higher levels, resulting in a poor data locality and an increase of communication. By reducing the maximum number of partitions for these smaller trees, this problem can be minimized. However, a more flexible asynchronous implementation with hierarchical partitioning that involves only the same nodes that contain the tree at the lowest level remains to be investigated.

V. Examples

For the calculation of the results in this Section, two different computational clusters were used. On the one hand, a cluster consisting of 8 machines, each containing 2 quad-core AMD Opteron 2350 processors and 32 Gigabyte RAM (64 cores and 256 GByte RAM in total). A

simple Gigabit Ethernet network was used as an interconnection between the machines and the open LAM [41] implementation of the MPI was used as a communication library.

On the other hand, a much larger and more recent cluster consisting of 32 machines, each containing 2 quad-core Intel Xeon L5420 processors and 16 Gigabyte RAM (512 cores and 512 GByte RAM in total). The machines are connected through a fast 20 Gigabit/s Infiniband network and the proprietary 'Intel MPI' implementation of the MPI was used as communication library. In what follows, we will refer to the clusters by their interconnection network, i.e. the 'Ethernet cluster' and the 'Infiniband cluster' respectively. Double precision calculations were used for all examples.

A. *Parallel efficiency*

Besides the scalability, another important property of a parallel algorithm is the parallel efficiency η, defined as the ratio of the speedup and the number of processors when considering a *fixed-size* problem.

$$\eta = \frac{T_1}{pT_p}, \tag{14}$$

where T_p indicates the runtime using p processors. In [25], using an asynchronous algorithm and *spatial* partitioning, this property was investigated for different geometries containing multiple dielectric objects up to 24 nodes. Using the hierarchical partitioning strategy and the same geometries as in [25], the parallel efficiencies were again determined on both the 'Ethernet' and 'Infiniband' clusters. Each test geometry was discretized in 5 million unknowns using a segment length of $\lambda/10$. This corresponds to the maximum problem size that could be held in the memory of a single machine. The results are shown in Fig. 5. Compared to the implementation using only spatial partitioning, the efficiencies clearly benefit from the hierarchical approach, in accordance to observations made in [21], [22]. The efficiencies on the 'Ethernet cluster' are between 40% to 50% for 64 nodes and between 47% and 72% for 512 nodes on the 'Infiniband cluster'. A parallel efficiency of 72% on 512 cores corresponds to a speedup of about 370. For the PEC simulation, this means that the duration for one matrix-vector multiplication is reduced from 182.74s using a single core to only 0.49s using 512 cores. It should be noted that these parallel efficiencies increase when handling larger problems.

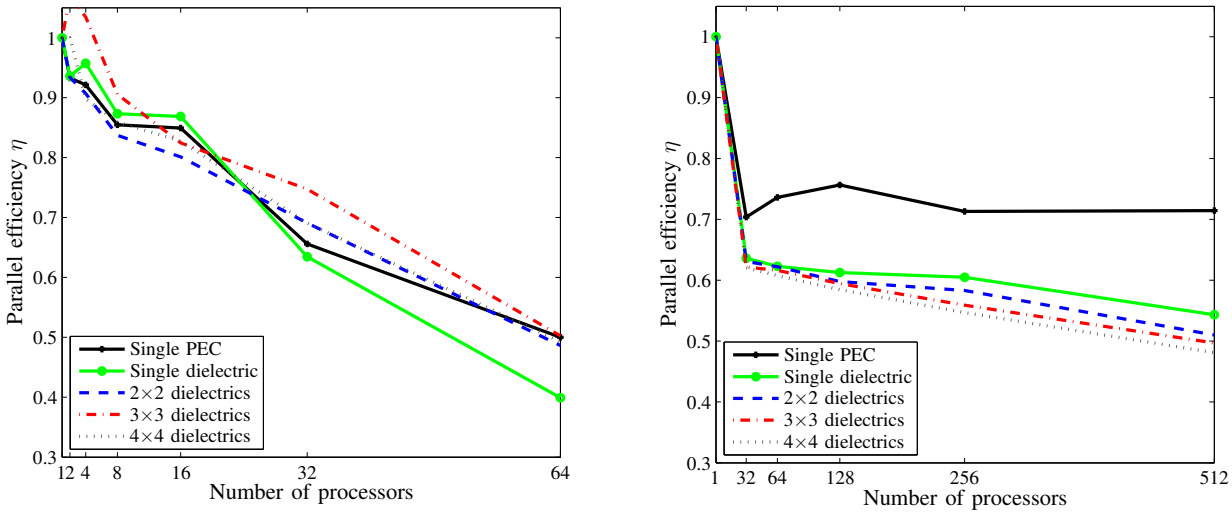


Fig. 5. Parallel efficiency η as a function of the number of processors on two clusters: the Gigabit Ethernet cluster (left) and the Infiniband cluster (right).

When comparing the two clusters, it is clear that a faster interconnection network is beneficial. The Infiniband network allows for data rates that are roughly 20 times higher than the Gigabit Ethernet network. Furthermore, the latency (time delay for communication) in the Ethernet network is in the order of milliseconds, whereas the Infiniband network has a latency in the order of microseconds. For comparison, the latency for accessing data which are stored in the local RAM memory is in the order of nanoseconds.

Besides the raw network speed, there is another important aspect when trying to compare the two clusters. The Ethernet interconnect requires many processor cycles in order to progress its communication. These processor cycles are spent within the MPI library. This extra work is attributed to a single core within the 8-core machines, hence, it appears that one core is running 'slower', resulting in waiting cycles for the other seven cores. This effect is also severely limiting the parallel efficiency.

B. Canonical examples

To test the accuracy of the implementation for very large-scale problems, we consider two canonical examples: the plane wave TM scattering at a PEC cylinder with a diameter of $2\,000\,000\lambda$ and at a PEC cylinder with a diameter of $325\,000\lambda$ embedded into a dielectric cylinder (relative

TABLE I

SIMULATION PARAMETERS FOR THE CANONICAL EXAMPLES

	PEC	PEC/diel
# unknowns	62 831 854	72 196 850
# processors	256	256
smallest box size	0.25 λ	0.20 λ
# MLFMA levels	24	23
precision FMM interactions	10^{-5}	10^{-4}
# iterations	971	1 201
preconditioner	16 $\lambda\times$ 16 λ	3.2 $\lambda\times$ 3.2 λ
setup time	34 min.	18 min.
solution time	4h 48min.	7h 16min.
time for matrix-vector product	8.47s	10.37s
# RCS output points	37 273 600	14 926 275

permittivity $\epsilon_r = 2$) with a diameter of $650\,000\lambda$. Using a $\lambda/10$ discretization, the former is discretized in $62\,831\,854$ unknowns, the latter in $72\,196\,850$ unknowns. Using the TFQMR [42] iterative method, the problems were solved to a relative residual error of 10^{-3}, using 256 cores and almost 512 GByte of RAM memory in total on the 'Infiniband cluster'. Other simulation parameters of interest are listed in table I.

For both cases, the bistatic radar cross-section (RCS) σ_c was numerically calculated in N equidistant angles θ_i and compared to the analytical solution σ_a. The evaluation of the analytical solution can be accelerated using the fast cosine transform, allowing for an evaluation in only a few minutes. We determine the root mean square (RMS) error as follows

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |\sigma_a(\theta_i) - \sigma_c(\theta_i)|^2} \tag{15}$$

For the full $[0^\circ \ldots 360^\circ]$ region, the RMS error is only 0.129 dB for the PEC simulation and 0.523 dB for the PEC/dielectric case, indeed yielding very accurate results. Fig. 6 both shows the full bistatic RCS and two small regions around 0° and 60° for both cases.

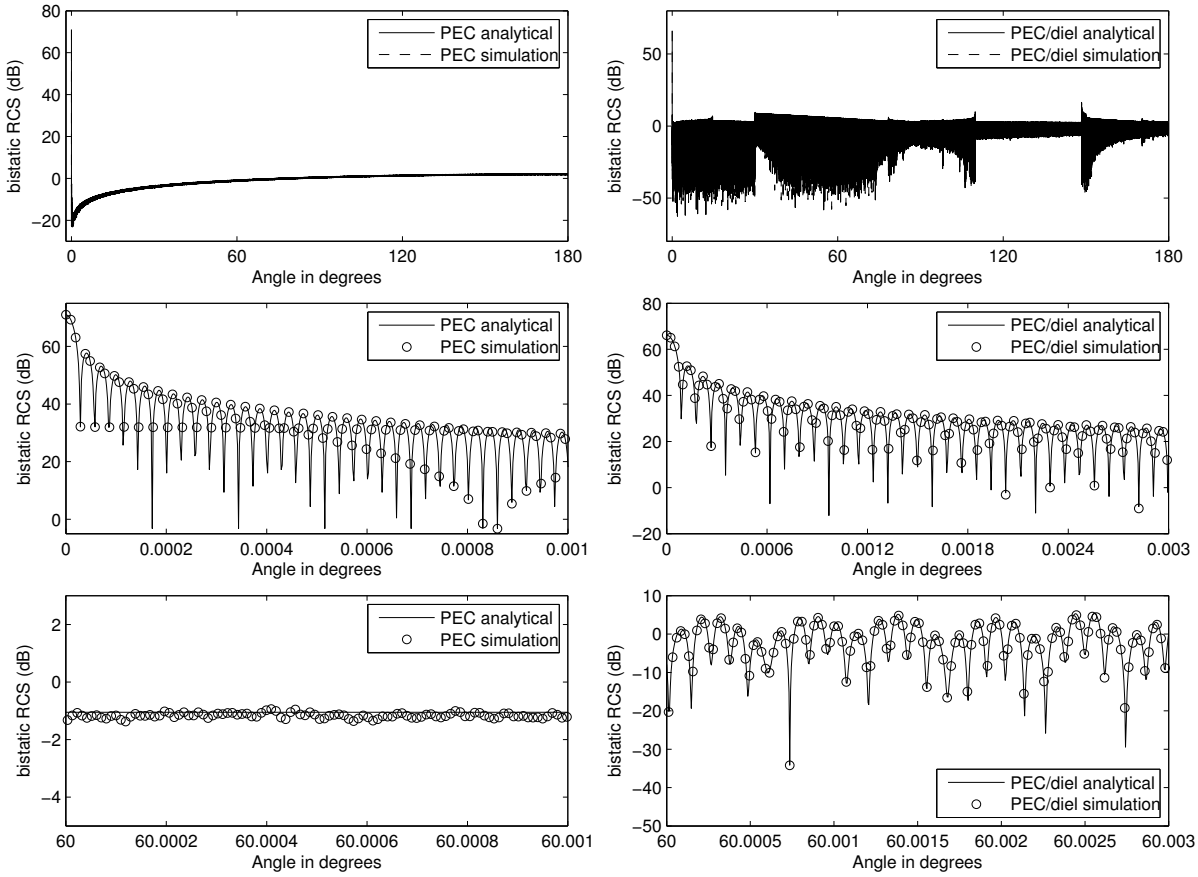


Fig. 6. Comparison between the analytical and simulated bistatic RCS for the scattering at a PEC cylinder with a diameter of $2\,000\,000\lambda$ (left) and a PEC cylinder with a diameter of $325\,000\lambda$ embedded into a dielectric ($\epsilon_r = 2$) cylinder with a diameter of $650\,000\lambda$. The full RCS $\theta \in [0^\circ \dots 180^\circ]$ (top) and details for θ around 0° (middle) and θ around 60° (bottom) is given.

C. Cassegrain reflector antenna

As a final example we consider a beam-waveguide antenna as depicted in Fig. 7. Although the dimensions were taken arbitrarily, the design is inspired by ESA's deep space antennas at New Norcia/Western Australia and Cebreros/Spain [43]. It features a 35-meter parabolic main reflector and a 6-meter hyperbolic sub-reflector. The lens system consists of a number of flat mirrors (M_1, M_2 and M_3), parabolic mirrors (P_1 and P_2) and an elliptical mirror (E). The highest frequency band of operation is the Ka-band (31.8-32.3 GHz downlink, 34.2-34.7 GHz uplink).

First, we consider the case where the structure operates as a transmitting antenna. A TM polarized Gaussian bundle with a beam waist of 2 meter illuminates a 4 meter lens with has a focal point f_1. The lens has a quarter-wavelength coating to eliminate reflections. This is achieved

by embedding a lens with a relative permittivity ϵ_r = 4 into a slightly larger lens with a relative permittivity ϵ_r = 2. The behavior of such a coated lens has already been studied in [23]. The image from the lens is reflected to the elliptical mirror E which has its focal points in f_1' and f_2 and is further reflected by means of mirror M_2 to the identical parabolic mirrors P_1 and P_2 with a focal point in f_2' and f_3' respectively. Finally, mirror M_3 feeds the image to the hyperbolic sub-reflector and finally the parabolic main reflector. Note that every mirror is modeled as a closed PEC object with a thickness of 5mm.

Second, we consider the case where the structure is illuminated with a TM polarized plane wave incident from top to bottom and acts as a receiving antenna. In this case, the lens is omitted. In both cases (transmitting/receiving), the frequency was 32 GHz, hence the structures are approximately 4 000 λ in diameter. The simulations were carried out on 16 cores on the 'Ethernet cluster'. TFQMR was used to calculate the iterative solution until a relative residual error of 10^{-4} was obtained. Table II lists the simulation parameters in both cases.

The electrical field densities of the two simulations are illustrated in Fig. 8(a) and Fig. 8(b). One can clearly see the behavior of each mirror as predicted by the laws of optics. However, the full-wave model incorporates all kinds of possible artifacts. For instance, the vertical interference pattern emerging in the field that is reflected by the main parabolic dish in 8(b) is likely due to the lens aberrations.

Note that these 2D problems can be considered as small, because their total simulation time is well below 10 minutes. However, it allows for many runs that are usually required during a geometry optimization process in the design phase.

VI. Conclusions

We presented a general-purpose two-dimensional boundary integral equation solver as an open-source package to the electromagnetic community (http://www.openfmm.net). The software is capable of handling complex geometries with multiple dielectric and/or conducting objects, and is accelerated using an asynchronous implementation of the parallel MLFMA. We have demonstrated how to obtain a high accuracy of the Method of Moments matrix elements by handling both the singular part of both self- and neighbor-patches analytically. Furthermore, we have shown how the use of the hierarchical partitioning technique can yield a scalable algorithm with good parallel efficiencies, even on slower interconnection networks. Several canonical and

TABLE II

SIMULATION PARAMETERS FOR THE CASSEGRAIN ANTENNA

	transmitting	receiving
# unknowns	288 485	216 485
smallest box size	0.25λ	0.25λ
preconditioner	$8\lambda \times 8\lambda$	$8\lambda \times 8\lambda$
# iterations	171	111
memory per node	104 MByte	16.89 MByte
setup time	274s	197s
solution time	184s	89s

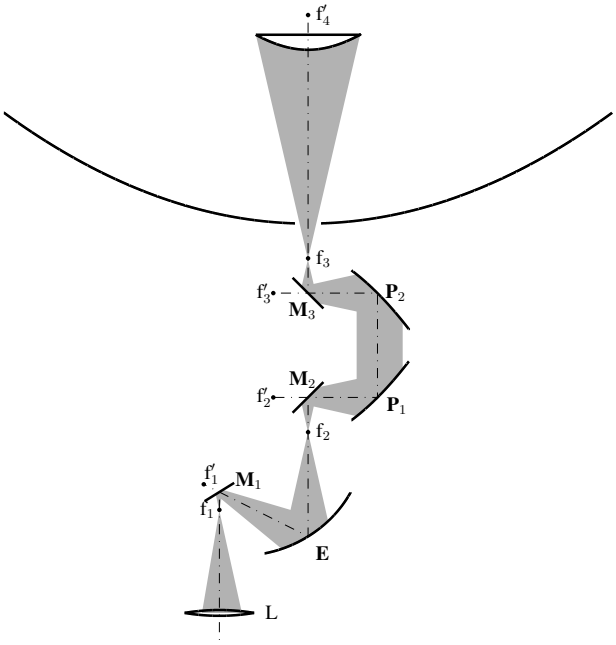
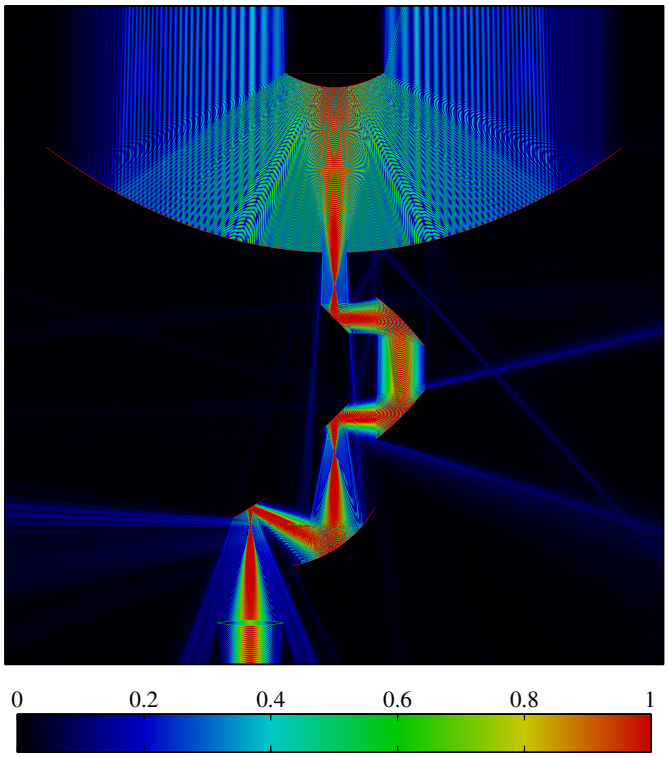


Fig. 7. Schematic of a Cassegrain reflector antenna

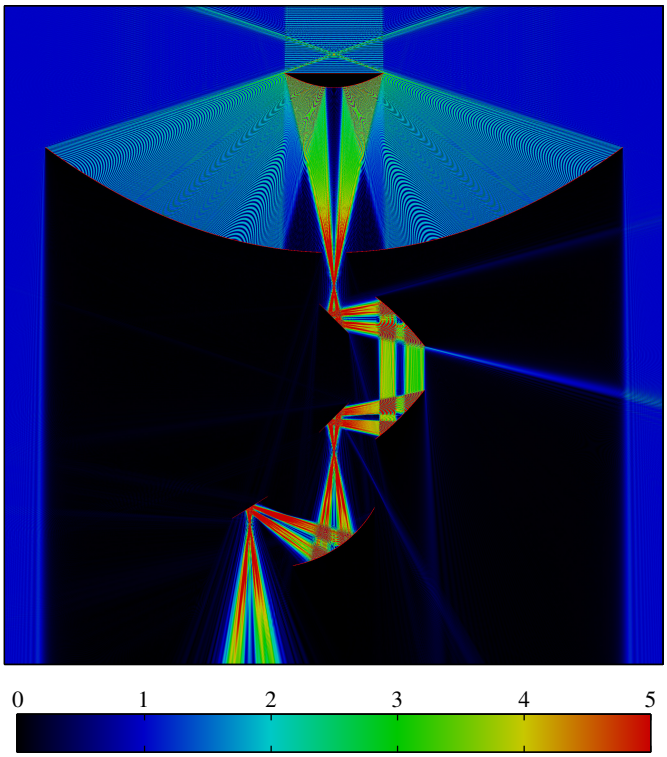
more realistic examples were presented to demonstrate the high accuracies that can be obtained.

ACKNOWLEDGMENT

Some of the computational resources and services used in this work were provided by Ghent University. In particular, the authors would like to acknowledge Stijn De Weirdt for his technical assistance and expertise.



(a) Electrical field density $\|\mathrm{e}_z\|$ (V/m) of the Cassegrain antenna illuminated with a Gaussian bundle incident from the bottom.



(b) Electrical field density $\|\mathrm{e}_z\|$ (V/m) of the Cassegrain antenna illuminated with a plane wave bundle incident from the top.

APPENDIX A

$$f_1(l_1,l_2)=\frac{1}{4}\left(1-\frac{l_1\cos\phi}{l_2}\right)\beta-\frac{3}{4}+\frac{1}{2}\frac{l_1}{l_2}\left[\gamma(l_1)\cos\phi+\nu(l_1,l_2)\right], \tag{16}$$

$$f_2(l_1,l_2)=\frac{1}{2}\left(l_1^2\cos\phi+2l_1l_2-l_2^2\cos\phi\right)\delta+l_1l_2\alpha(l_2,l_1) \tag{17}$$

$$f_3(l_1,l_2)=\frac{l_1^2}{2}\alpha(l_1,l_2)\cos\phi+\frac{l_1^2}{4}\left[\beta-\gamma(l_1)\right]\sin\phi, \tag{18}$$

$$f_4(l_1,l_2)=f_3(l_1,l_2)+\frac{l_1^2}{2}\delta\cos\phi, \tag{19}$$

$$f_5(l_1,l_2)=\left(\frac{l_1l_2}{3}-\frac{l_1^2}{6}\cos\phi\right)f_{11}(l_1,l_2)$$

$$+\frac{l_1^4}{48}\left[2\gamma(l_1)-\beta\right]+\frac{1}{16}l_1^2l_2^2\beta+\frac{l_1^3l_2}{24}\cos\phi-\frac{25}{96}l_1^2l_2^2, \tag{20}$$

$$f_6(l_1,l_2)=\frac{l_1l_2}{4}\left[1+\beta-4\gamma(l_1)\right]+f_{11}(l_1,l_2), \tag{21}$$

$$f_7(l_1,l_2)=\frac{l_1^4}{4}\nu(l_1,l_2)\cos\phi+\frac{l_1^4}{16}\left[2\gamma(l_1)-\beta\right]\cos 2\phi$$

$$-\frac{l_1^3l_2}{8}\cos\phi+\frac{l_1^2l_2^2}{32}(2\beta-3), \tag{22}$$

$$f_8(l_1,l_2)=-f_{11}(l_1,l_2)+\frac{l_1l_2}{4}, \tag{23}$$

$$f_9(l_1,l_2)=\left(\frac{l_1^2l_2^2}{8}-\frac{l_1^3l_2}{6}\cos\phi\right)\beta$$

$$+\frac{l_1^3l_2}{3}(\nu(l_1,l_2)+\gamma(l_1)\cos\phi)-\frac{19}{48}l_1^2l_2^2, \tag{24}$$

$$f_{10}(l_1,l_2)=\frac{l_1^4}{16}\beta\cos 2\phi+\frac{l_1^3l_2}{8}\cos\phi$$

$$-\frac{l_1^4}{8}\cos\phi\left[2\nu(l_1,l_2)+\gamma(l_1)(2\cos\phi-1)\right], \tag{25}$$

$$f_{11}(l_1,l_2)=\frac{l_1^2}{4}[2\nu(l_1,l_2)+(2\gamma(l_1)-\beta)\cos\phi], \tag{26}$$

with

$$\alpha(l_1,l_2)=\arctan\left(\frac{l_2-l_1\cos\phi}{l_1\sin\phi}\right),\beta=\log(l_1^2-2l_1l_2\cos\phi+l_2^2),$$

$$\gamma(l)=\log l,\delta=\arctan(\cot\phi),\nu(l_1,l_2)=[\alpha(l_1,l_2)+\delta(l_1,l_2)]\sin\phi. \tag{27}$$

REFERENCES

[1] J. Van Bladel and K. Mei, "On the capacitance of a cube," *Applied Scientific Research*, vol. B9, no. 4-5, pp. 267–270, Aug. 1962.

[2] R. Harrington, "Origin and development of the method of moments for field computation," *IEEE Antennas Propag. Mag.*, vol. 32, no. 3, pp. 31–35, Jun. 1990.

[3] E. Yamashita and R. Mittra, "Variational methods for the analysis of microstrip lines," *IEEE Trans. Microw. Theory Tech.*, vol. 16, no. 4, pp. 251–256, Apr. 1968.

[4] K. Blomme and D. De Zutter, "Multivoice subsectional wavelets creating sparse matrices in the moment method applied to coupled microstrips," *Electromagnetics*, vol. 17, no. 5, pp. 483–503, Sep. 1997.

[5] F. Canning, "The impedance matrix localization method for moment-method calculations," *IEEE Antennas Propag. Mag.*, vol. 32, no. 5, pp. 18–30, Oct. 1990.

[6] P. Zwamborn and P. van den Berg, "The three-dimensional weak form of the conjugate gradient FFT method for solving scattering problems," *IEEE Trans. Microw. Theory Tech.*, vol. 40, no. 9, pp. 1757–1766, Sep. 1992.

[7] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Comput. Phys.*, vol. 73, no. 2, pp. 325–348, Dec. 1987.

[8] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *J. Comput. Phys.*, vol. 60, no. 2, pp. 187–207, Sep. 1985.

[9] C. Lu and W. Chew, "A multilevel algorithm for solving a boundary integral equation of wave scattering," *Microwave and Optical Technology Letters*, vol. 7, no. 10, pp. 466–470, july 1994.

[10] W. C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*. Boston: Artech House, 2001.

[11] I. Bogaert, D. Pissoort, and F. Olyslager, "A normalized plane wave method for 2-D Helmholtz problems," *Microwave and Optical Technology Letters*, vol. 48, no. 2, pp. 237–243, Feb. 2006.

[12] I. Bogaert, J. Peeters, and F. Olyslager, "A nondirective plane wave MLFMA stable at low frequencies," *IEEE Trans. Antennas Propag.*, vol. 56, no. 12, pp. 3752–3767, Dec. 2008.

[13] I. Bogaert and F. Olyslager, "A low frequency stable plane wave addition theorem," *J. Comput. Phys.*, vol. 228, no. 4, pp. 1000–1016, Mar. 2009.

[14] T. Malas and L. Gürel, "Incomplete LU preconditioning with the multilevel fast multipole algorithm for electromagnetic scattering," *SIAM Journal on Scientific Computing*, vol. 29, no. 4, pp. 1476–1494, Jun. 2007.

[15] F. Andriulli, K. Cools, H. Bağci, F. Olyslager, A. Buffa, S. Christiansen, and E. Michielssen, "A multiplicative Calderón preconditioner for the electric field integral equation," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2398–2412, Aug. 2008.

[16] H. Bağci, F. Andriulli, K. Cools, F. Olyslager, and E. Michielssen, "A Calderón multiplicative preconditioner for the combined field integral equation," *IEEE Trans. Antennas Propag.*, vol. 57, no. 10, pp. 3387–3392, Oct. 2009.

[17] P. F. Gorder, "Multicore processors for science and engineering," *IEEE Comput. Sci. Eng.*, vol. 9, no. 2, pp. 3–7, Mar. 2007.

[18] 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.

[19] X.-M. Pan and X.-Q. Sheng, "A sophisticated parallel MLFMA for scattering by extremely large targets," *IEEE Antennas Propag. Mag.*, vol. 50, no. 3, pp. 129–138, Jun. 2008.

[20] Ö. Ergül and L. Gürel, "Efficient parallelization of the multilevel fast multipole algorithm for the solution of large-scale scattering problems," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2235–2345, Aug. 2008.

[21] ——, "Hierarchical parallelisation strategy for multilevel fast multipole algorithm in computational electromagnetics," *Electronics Letters*, vol. 44, no. 1, pp. 3–5, Jan. 2008.

[22] ——, "A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 57, no. 6, pp. 1740–1750, Jun. 2009.

[23] J. Fostier and F. Olyslager, "A GRID computer implementation of the multilevel fast multipole algorithm for full-wave analysis of optical devices," *IEICE Trans. Comm.*, vol. E90-B, no. 9, pp. 2430–2438, Sep. 2007.

[24] ——, "An asynchronous parallel multilevel fast multipole algorithm," in *Proc. IEEE Int. Symp. Antennas Propag.*, Hawaii, USA, Jun. 2007, pp. 3424–3427.

[25] ——, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2346–2355, Aug. 2008.

[26] ——, "Provably scalable parallel multilevel fast multipole algorithm," *IET Electronics Letters*, vol. 44, no. 19, pp. 1111–1113, Sep. 2008.

[27] A. Poggio and E. Miller, "Integral equation solution of three-dimensional scattering problems," in *Computer Techniques for Electromagnetics*, R. Mittra, Ed. Elmsford, N.Y.: Pergamon, 1973, ch. 4.

[28] Y. Chang and R. Harrington, "A surface formulation for characteristic modes of material bodies," *IEEE Trans. Antennas Propag.*, vol. 25, no. 6, pp. 789–795, Nov. 1977.

[29] T. Wu and L. Tsai, "Scattering from arbitrarily-shaped lossy dielectric bodies of revolution," *Radio Science*, vol. 12, no. 5, pp. 709–718, Sep. 1977.

[30] J. Fostier and F. Olyslager, "Full-wave electromagnetic scattering at extremely large 2D objects," *IET Electronics Letters*, vol. 45, no. 5, pp. 245–246, Feb. 2009.

[31] F. Olyslager, D. De Zutter, and K. Blomme, "Rigourous analysis of the propagation characteristics of general lossless and lossy multiconductor transmission lines in multilayered media," *IEEE Trans. Microw. Theory Tech.*, vol. 41, no. 1, pp. 79–88, Jan. 1993.

[32] F. Olyslager and D. De Zutter, "Rigourous full-wave analysis of electric and dielectric waveguides embedded in a multilayered bianisotropic medium," *Radio Science*, vol. 28, no. 5, pp. 937–946, Sep. 1993.

[33] ——, "Rigourous boundary integral equation solution for general isotropic and uniaxial anisotropic dielectric waveguides in multilayered media including losses, gain and leakage," *IEEE Trans. Microw. Theory Tech.*, vol. 41, no. 8, pp. 1385–1392, Aug. 1993.

[34] J. Fostier and F. Olyslager, "A fast parallel 2.5D MLFMA solver," in *Proceedings of the XXIX General Assembly of the International Union of Radio Science*, Chicago, USA, Aug. 2008.

[35] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2719–2727, Aug. 2005.

[36] S. Ohnuki and W. C. Chew, "Numerical analysis of local interpolation error for 2D-MLFMA," *Microw. and Opt. Techn. Lett.*, vol. 36, no. 1, pp. 8–12, Jan. 2003.

[37] M. Lu and E. Michielssen, "A local filtering scheme for FMM/PWTD algorithms," in *Proc. IEEE Int. Symp. Antennas Propag.*, vol. 4, Jun. 2004, pp. 4523–4526.

[38] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005, www.fftw.org.

[39] D. E. Amos, "A portable package for bessel functions of a complex argument and nonnegative order," *ACM Trans. Math. Soft. (TOMS)*, vol. 12, no. 3, pp. 265–273, Sep. 1986.

[40] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. Cambridge, MA: MIT Press, 1994.

[41] G. Burns, R. Daoud, and J. Vaigl, "LAM: An Open Cluster Environment for MPI," in *Proceedings of Supercomputing Symposium*, 1994, pp. 379–386. [Online]. Available: http://www.lam-mpi.org/download/files/lam-papers.tar.gz

[42] R. W. Freund, "A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems," *SIAM J. Sci. Comput.*, vol. 14, no. 2, pp. 470–482, Mar. 1993.

[43] R. Martin and M. Warhaut, "ESA's 35-meter deep space antennas at New Norcia/Western Australia and Cebreros/Spain," in *Proc. Aerospace Conf.*, vol. 2, Mar. 2004, pp. 1124–1133.