Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 7052–7068
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An efficient hybrid MLFMA–FFT solver for the volume integral equation in case of sparse 3D inhomogeneous dielectric scatterers

J. De Zaeytijd *, I. Bogaert, A. Franchois

Department of Information Technology (INTEC), Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium

Received 15 June 2007; received in revised form 8 April 2008; accepted 11 April 2008
Available online 22 April 2008

Abstract

Electromagnetic scattering problems involving inhomogeneous objects can be numerically solved by applying a Method of Moments discretization to the volume integral equation. For electrically large problems, the iterative solution of the resulting linear system is expensive, both computationally and in memory use. In this paper, a hybrid MLFMA–FFT method is presented, which combines the fast Fourier transform (FFT) method and the High Frequency Multilevel Fast Multipole Algorithm (MLFMA) in order to reduce the cost of the matrix–vector multiplications needed in the iterative solver. The method represents the scatterers within a set of possibly disjoint identical cubic subdomains, which are meshed using a uniform cubic grid. This specific mesh allows for the application of FFTs to calculate the near interactions in the MLFMA and reduces the memory cost considerably, since the aggregation and disaggregation matrices of the MLFMA can be reused. Additional improvements to the general MLFMA framework, such as an extention of the FFT interpolation scheme of Sarvas et al. from the scalar to the vectorial case in combination with a more economical representation of the radiation patterns on the lowest level in vector spherical harmonics, are proposed and the choice of the subdomain size is discussed. The hybrid method performs better in terms of speed and memory use on large sparse configurations than both the FFT method and the HF MLFMA separately and it has lower memory requirements on general large problems. This is illustrated on a number of representative numerical test cases.

Keywords: Volume integral equation; Method of moments; FFT method; HF MLFMA; Vector spherical harmonics; Sparse geometries; Hybrid technique; Electromagnetic scattering

1. Introduction

In several research domains, it is necessary to simulate the full-vectorial, three-dimensional scattering of electromagnetic waves from inhomogeneous dielectric objects. A few examples are the calculation of radar cross

* Corresponding author.
E-mail address: jurgen.dezaeytijd@intec.ugent.be (J. De Zaeytijd).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.04.009

sections of composite structures, the homogenization of meta materials and the reconstruction of objects with inhomogeneous permittivity from scattering data. A volume integral equation (VIE) formulation provides a solid framework for such scattering simulations. Typically, a method of moments (MoM) discretization is used to obtain an N-dimensional linear system which is then solved numerically. However, if the electrical size of the scatterers increases, inverting directly this $N \times N$-system becomes very slow due to the $\mathcal{O}(N^3)$ computational complexity. Moreover, the memory needed to store the system's $N \times N$-matrix can easily exceed the available computer memory. Solving the system iteratively using, for example, a conjugate gradient (CG) or stabilized bi-conjugate gradient (BICGSTAB) solver reduces the computational complexity to $\mathcal{O}(N^{\mathrm{I}}N^2)$, with N^{I} the number of iterations, but the system's matrix still has to be stored in this approach.

Two classes of methods that reduce the storage requirements and speed up the matrix–vector multiplications needed in every step of the iterative solver are the Fast Fourier transform (FFT) based techniques and the Multilevel Fast Multipole algorithms (MLFMA's). The first class [1–5] uses the FFT to exploit the convolutional structure of the integral operator in the VIE and has a computational complexity (for one matrix–vector multiplication) of $\mathcal{O}(N \log N)$ and a memory use of $\mathcal{O}(N)$. The methods in the second class, such as the High Frequency (HF) MLFMA [6–9], the Low Frequency MLFMA [10] and the Inhomogeneous or Stable Plane wave method [11,12], are based on efficient decompositions of the Green function. The major advantages of the FFT methods are their speed and easy implementation, thanks to the fast, reliable and widespread codes for calculating FFTs [13]. The MLFMA's on the other hand allow a more flexible meshing of the scattering geometry, since they can be applied to arbitrary meshes. For moderate to large volumetric problems with densely distributed mesh elements, the FFT methods are usually faster, thanks to their small prefactor, despite the lower computational complexity of MLFMA's ($\mathcal{O}(N)$) on such dense geometries.

In this paper, a hybrid MLFMA–FFT method is proposed, which is particularly suited for large scattering configurations that show some sparsity. The method is a modification of the HF MLFMA that treats the interactions between nearby mesh elements using FFTs and the interactions between well separated elements as in a regular HF MLFMA. It can also be regarded as a hybridization of the subdomain FFT method, which is proposed here as an FFT method for a collection of cubic subdomains. The subdomain meshing avoids the extension of the FFT grid over empty space between scatterers, as is necessary in the classical FFT method. It will be shown that the MLFMA–FFT method outperforms both the regular HF MLFMA and the FFT method on large sparse geometries and that it can have lower memory requirements even on large dense geometries.

The outline of this paper is as follows. In Section 2 the 3D scattering problem is formulated using a VIE and discretized with a MoM scheme. Section 3 proposes a subdomain FFT method to speed up the matrix–vector products needed for the iterative solution of the linear system. Section 4 starts by shortly revisiting the HF MLFMA. For a more thorough treatment, the reader is referred to [9]. Next, some improvements to the general MLFMA framework are presented. Specifically, the exploitation of symmetries in the subdomain mesh allows for a reduction of the memory cost of the MLFMA and the application of an FFT interpolation scheme for the vectorial MLFMA and the use of vector spherical harmonics to represent the radiation patterns on the lowest level result in accurate and efficient aggregation and disaggregation stages. Section 5 presents the hybrid MLFMA–FFT and discusses its relation to the FFT method and the HF MLFMA. Finally, several numerical examples are given in Section 6 to validate the method and to demonstrate its accuracy and superior performance.

Throughout this paper, we will work in the frequency regime and the time dependency $e^{j\omega t}$, with ω the angular frequency, will be implicitly assumed.

2. Problem formulation

2.1. 3D volume integral equation

Consider a number of 3D inhomogeneous, possibly lossy, dielectric objects with arbitrary shape that are situated in a homogeneous background medium with complex permittivity ϵ_b (Fig. 1). All materials are non-magnetic and have permeability μ_0. The scatterers are characterized by the complex permittivity $\epsilon(\boldsymbol{r}) = \epsilon'(\boldsymbol{r}) - j\epsilon''(\boldsymbol{r})$,

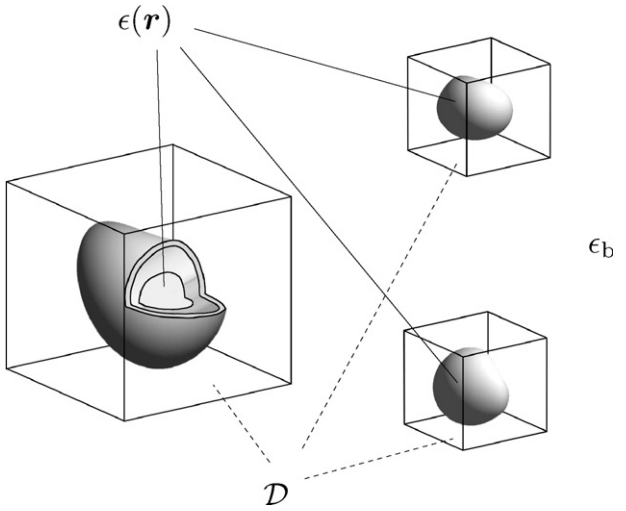

Fig. 1. 3D scattering configuration.

where $\boldsymbol{r} = x\hat{\boldsymbol{x}} + y\hat{\boldsymbol{y}} + z\hat{\boldsymbol{z}}$ denotes the position and $\hat{\boldsymbol{u}}$ is a unit vector in the u-direction. Only for $\boldsymbol{r}$ in one of the objects, the normalized contrast function

$$\chi(\boldsymbol{r}) = \frac{\epsilon(\boldsymbol{r}) - \epsilon_b}{\epsilon(\boldsymbol{r})}, \tag{1}$$

with respect to the background medium, is non-zero. A bounded, but not necessarily connected, domain $\mathcal{D}$ is defined that completely includes all the scatterers and on this domain the mixed potential formulation of the electrical field volume integral equation is considered [14]:

$$\boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) = \frac{\boldsymbol{D}(\boldsymbol{r})}{\epsilon(\boldsymbol{r})} + \mathrm{j}\omega\boldsymbol{A}^{\text{scat}}(\boldsymbol{r}) + \nabla\varphi^{\text{scat}}(\boldsymbol{r}), \tag{2}$$

where $\boldsymbol{E}^{\text{inc}}$ is a known time-harmonic incident field, $\boldsymbol{D}$ represents the electric flux density $\boldsymbol{D} = \epsilon\boldsymbol{E}$ with $\boldsymbol{E}$ the total electric field and where the scattered field $\boldsymbol{E}^{\text{scat}} = \boldsymbol{E} - \boldsymbol{E}^{\text{inc}}$ is expressed as

$$\boldsymbol{E}^{\text{scat}}(\boldsymbol{r}) = -\mathrm{j}\omega\boldsymbol{A}^{\text{scat}}(\boldsymbol{r}) - \nabla\varphi^{\text{scat}}(\boldsymbol{r}). \tag{3}$$

$\boldsymbol{A}^{\text{scat}}$ and φ^{scat}, respectively, represent the vector and scalar potentials corresponding to the contrast current density $\boldsymbol{J}^{\text{scat}}$ and the associated contrast charge density ρ^{scat}, defined by

$$\boldsymbol{J}^{\text{scat}}(\boldsymbol{r}) = \mathrm{j}\omega[\epsilon(\boldsymbol{r}) - \epsilon_b]\boldsymbol{E}(\boldsymbol{r}) = \mathrm{j}\omega\chi(\boldsymbol{r})\boldsymbol{D}(\boldsymbol{r}), \tag{4}$$

$$\rho^{\text{scat}}(\boldsymbol{r}) = \frac{-1}{\mathrm{j}\omega}\nabla\cdot\boldsymbol{J}^{\text{scat}}(\boldsymbol{r}). \tag{5}$$

The potentials are given by

$$\boldsymbol{A}^{\text{scat}}(\boldsymbol{r}) = \mu_0\int_{\mathcal{D}} G_b(\boldsymbol{r}-\boldsymbol{r}')\boldsymbol{J}^{\text{scat}}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}', \tag{6}$$

$$\varphi^{\text{scat}}(\boldsymbol{r}) = \frac{1}{\epsilon_b}\int_{\mathcal{D}} G_b(\boldsymbol{r}-\boldsymbol{r}')\rho^{\text{scat}}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}', \tag{7}$$

where

$$G_b(\boldsymbol{r}-\boldsymbol{r}') = \frac{\mathrm{e}^{-\mathrm{j}k_b\|\boldsymbol{r}-\boldsymbol{r}'\|}}{4\pi\|\boldsymbol{r}-\boldsymbol{r}'\|} \tag{8}$$

is the Green function of the background medium with propagation constant $k_b = \omega\sqrt{\epsilon_b\mu_0}$. Considering (4)–(7), it follows that the domain Eq. (2) is a Fredholm integral equation of the second kind for the unknown

electric flux density $\boldsymbol{D}$. Once (2) is solved for $\boldsymbol{D}$ in $\mathcal{D}$, (3) can be used to calculate the scattered field everywhere in space.

2.2. MoM discretization of the VIE

To solve (2) numerically, a Galerkin Method of Moments (MoM) is applied. The electric flux density is expanded as

$$\boldsymbol{D}(\boldsymbol{r}) \approx \sum_{\alpha=1}^{N} d_\alpha \boldsymbol{\Psi}_\alpha(\boldsymbol{r}) \quad \forall \boldsymbol{r} \in \mathcal{D}, \tag{9}$$

where $\boldsymbol{\Psi}_\alpha$ are vectorial basis functions and d_α are the unknown expansion coefficients. To define the functions $\boldsymbol{\Psi}_\alpha$, the domain $\mathcal{D}$ is built from a number of identical cubic subdomains D^a, $a = 1, \ldots, N^{\mathrm{D}}$, which belong to a uniform cubic grid with grid parameter Δ, and every subdomain D^a is divided in $N^{\mathrm{C}} = P^3$ cubic cells with side δ (Fig. 2). To every cell facet F_α in this grid, one basis function $\boldsymbol{\Psi}_\alpha = \Psi_\alpha \hat{\boldsymbol{u}}_\alpha$ is assigned, where $\hat{\boldsymbol{u}}_\alpha$ is the normal to F_α ($\hat{\boldsymbol{u}}_\alpha = \hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$ or $\hat{\boldsymbol{z}}$) and Ψ_α is a 3D rooftop function that assumes the value 1 on F_α and linearly tends to zero along the directions $\pm\hat{\boldsymbol{u}}_\alpha$ over the two cells S_α^+ and S_α^- that share F_α (Fig. 3). Such basis functions are also reported in [3,15] and a similar formulation on tetrahedral meshes is proposed in [14]. The support of Ψ_α is denoted $S_\alpha = S_\alpha^+ \cup S_\alpha^-$. With this choice for the basis functions, the normal component of $\boldsymbol{D}$ is continuous across all facets of the grid, as required by the boundary conditions. Furthermore, the contrast function χ is approximated by a piecewise constant function which assumes one value per cell. In cell S_α^+ (respectively, S_α^-) this value is denoted as χ_α^+ (respectively, χ_α^-). After substituting the expansion (9) in the domain Eq. (2), the same vectorial rooftop functions are used to test the equation. To illustrate the MoM procedure, consider $\boldsymbol{A}_\alpha^{\mathrm{scat}}$, $\phi_\alpha^{\mathrm{scat}}$

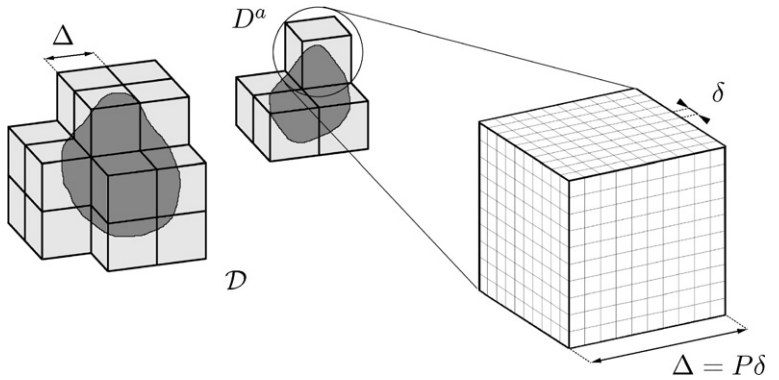

Fig. 2. The construction of the domain $\mathcal{D}$ from a number of identical cubic subdomains D^a, meshed with a uniform cubic grid.

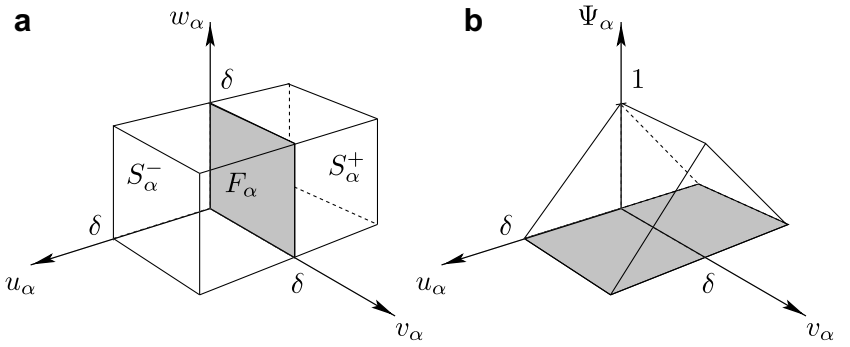

Fig. 3. Graphical representation of the rooftop function Ψ_α associated to facet F_α: (a) definition of the support cells S_α^+ and S_α^- and a local reference system $(\hat{\boldsymbol{u}}_\alpha, \hat{\boldsymbol{v}}_\alpha, \hat{\boldsymbol{w}}_\alpha)$, where $\hat{\boldsymbol{u}}_\alpha$ is the normal to F_α, and (b) Ψ_α as a function of u_α and v_α for every w_α: $0 \leqslant w_\alpha \leqslant \delta$.

and $\boldsymbol{E}_\alpha$, respectively, the potentials due to and the total electric field corresponding to one term $d_\alpha \boldsymbol{\Psi}_\alpha(\boldsymbol{r})$ in (9). Testing $\boldsymbol{A}_\alpha^{\text{scat}}$ and $\nabla\phi_\alpha^{\text{scat}}$ with $\boldsymbol{\Psi}_\beta$ yields

$$\int_{S_\beta} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \boldsymbol{A}_\alpha^{\text{scat}}(\boldsymbol{r}) \mathrm{d}\boldsymbol{r} = \mathrm{j}\omega\mu_0 d_\alpha \left[\chi_\alpha^+ \int_{S_\beta} \mathrm{d}\boldsymbol{r} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \int_{S_\alpha^+} \mathrm{d}\boldsymbol{r}' G_{\mathrm{b}}(\boldsymbol{r}-\boldsymbol{r}') \boldsymbol{\Psi}_\alpha(\boldsymbol{r}') + \chi_\alpha^- \int_{S_\beta} \mathrm{d}\boldsymbol{r} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \int_{S_\alpha^-} \mathrm{d}\boldsymbol{r}' G_{\mathrm{b}}(\boldsymbol{r}-\boldsymbol{r}') \boldsymbol{\Psi}_\alpha(\boldsymbol{r}') \right] \tag{10}$$

and

$$\int_{S_\beta} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \nabla\phi_\alpha^{\text{scat}}(\boldsymbol{r}) \mathrm{d}\boldsymbol{r} = -\int_{S_\beta} \phi_\alpha^{\text{scat}}(\boldsymbol{r}) \nabla \cdot \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \mathrm{d}\boldsymbol{r}, \tag{11}$$

where we have used Gauss' theorem and the fact that $\boldsymbol{\Psi}_\beta$ is zero outside S_β. The expression for $\phi_\alpha^{\text{scat}}$ in (11) is given by

$$\phi_\alpha^{\text{scat}}(\boldsymbol{r}) = -\frac{1}{\epsilon_{\mathrm{b}}} d_\alpha \left[\chi_\alpha^+ \int_{S_\alpha^+} G_{\mathrm{b}}(\boldsymbol{r}-\boldsymbol{r}') \nabla' \cdot \boldsymbol{\Psi}_\alpha(\boldsymbol{r}') \mathrm{d}\boldsymbol{r}' + \chi_\alpha^- \int_{S_\alpha^-} G_{\mathrm{b}}(\boldsymbol{r}-\boldsymbol{r}') \nabla' \cdot \boldsymbol{\Psi}_\alpha(\boldsymbol{r}') \mathrm{d}\boldsymbol{r}' + \left(\chi_\alpha^- - \chi_\alpha^+\right) \int_{F_\alpha} G_{\mathrm{b}}(\boldsymbol{r}-\boldsymbol{r}') \mathrm{d}\boldsymbol{r}' \right], \tag{12}$$

where the last term represents the contribution of the surface charges that arise from the discontinuities in the discretized contrast currents across the facets of the grid. The testing of $\boldsymbol{E}_\alpha$ with $\boldsymbol{\Psi}_\beta$ results in

$$\int_{S_\beta} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \boldsymbol{E}_\alpha(\boldsymbol{r}) \mathrm{d}\boldsymbol{r} = \frac{d_\alpha}{\epsilon_{\mathrm{b}}} \left[(1-\chi_\alpha^+) \int_{S_\beta \cap S_\alpha^+} \mathrm{d}\boldsymbol{r} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \boldsymbol{\Psi}_\alpha(\boldsymbol{r}) + (1-\chi_\alpha^-) \int_{S_\beta \cap S_\alpha^-} \mathrm{d}\boldsymbol{r} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \boldsymbol{\Psi}_\alpha(\boldsymbol{r}) \right]. \tag{13}$$

Note that we use complete rooftop functions $\boldsymbol{\Psi}_\beta$ in (10), (11) and (13) to test the VIE. Strictly speaking this generates a problem in (13) when the facets F_β and F_α coincide on the boundary of $\mathcal{D}$. In this case $S_\alpha^+ (= S_\beta^+)$ or $S_\alpha^- (= S_\beta^-)$ lies outside $\mathcal{D}$, where the expansion (9) is not defined. We therefore extend the validity of (9) over one cell-width exterior to $\mathcal{D}$ by replacing the half rooftop Ψ_α outside $\mathcal{D}$ with the constant unity function, i.e. $\boldsymbol{\Psi}_\alpha(\boldsymbol{r}) = \hat{\boldsymbol{u}}_\alpha$. This means we assume that keeping $\boldsymbol{D}(\boldsymbol{r}) \cdot \hat{\boldsymbol{u}}_\alpha$ constant and equal to its value on F_α is a good approximation over a distance δ outside $\mathcal{D}$.

Collecting the tested incident field in the N-dimensional vector $\boldsymbol{e}^{\text{inc}}$ and introducing N-dimensional vectors $\boldsymbol{d}_\chi^\pm$ with elements $\chi_\alpha^\pm d_\alpha$ and $\boldsymbol{d}$ with elements d_α, we finally obtain the following set of N linear equations in N unknowns d_α

$$\boldsymbol{e}^{\text{inc}} = \boldsymbol{W}_\chi \boldsymbol{d} + \boldsymbol{Z}^+ \boldsymbol{d}_\chi^+ + \boldsymbol{Z}^- \boldsymbol{d}_\chi^-. \tag{14}$$

The elements of the $N \times N$ matrices $\boldsymbol{Z}^\pm$ consist of double integrals with the Green function and at most two linear functions in the integrand, as appears from (10)–(12). These integrals are computed numerically using Gaussian quadrature and the $1/r$-singularity of the Green function is handled by singularity subtraction [16]. The sparse matrix $\boldsymbol{W}_\chi$ is derived from (13). Note that the elements of the dense matrices $\boldsymbol{Z}^\pm$ do not depend on the contrast. This way, they need to be calculated only once for a series of scattering simulations with varying contrast, for example in an inverse scattering problem, where the voxel permittivities are iteratively updated until the simulated forward scattering matches the measured data [17].

3. The subdomain FFT method

Since N is usually quite large, we solve (14) iteratively with a stabilized bi-conjugate gradient (BICGSTAB) routine [18]. To speed up the evaluations of the matrix–vector multiplications in (14), we focus on the most time consuming part, i.e. the multiplications with $\boldsymbol{Z}^\pm$. By denoting with $\boldsymbol{v}_a$ the part of an N-dimensional vector $\boldsymbol{v}$ that corresponds to basis or testing functions in the subdomain D^a, we can write

$$\left(\boldsymbol{Z}^+ \boldsymbol{d}_\chi^+ + \boldsymbol{Z}^- \boldsymbol{d}_\chi^-\right)_b = \sum_a \boldsymbol{Z}_{ba}^+ \boldsymbol{d}_{\chi,a}^+ + \sum_a \boldsymbol{Z}_{ba}^- \boldsymbol{d}_{\chi,a}^- \quad \forall b \in \{1, \ldots, N^D\}, \tag{15}$$

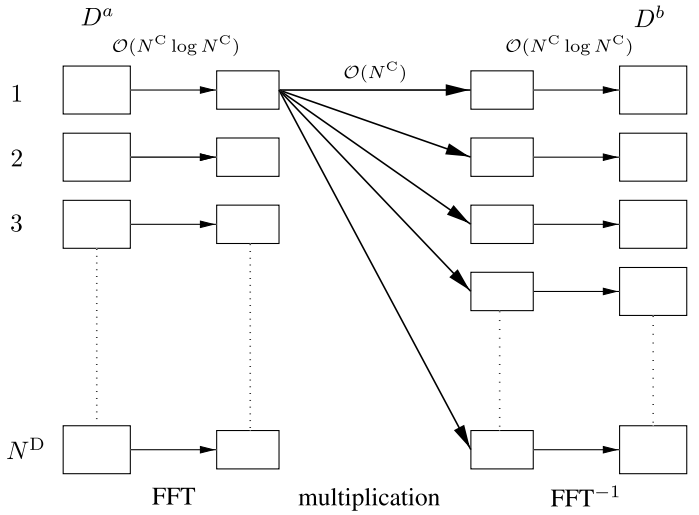

Fig. 4. Illustration of the computational complexity of the subdomain FFT method.

where the matrices $\boldsymbol{Z}^{\pm}_{ba}$ describe the interaction between two subdomains D^a and D^b. The matrix–vector products $\boldsymbol{Z}^{+}_{ba}\boldsymbol{d}^{+}_{\chi,a}$ and $\boldsymbol{Z}^{-}_{ba}\boldsymbol{d}^{-}_{\chi,a}$ can be cast as 3D discrete convolutions, since the convolutional symmetry in (6) and (7) is conserved thanks to the chosen discretization. Therefore, the index α and the notation F_α are replaced by the quintet $(p,q,r;a;u)$ and the notation $F^{a,u}_{p,q,r}$. The indices p,q and r determine the position of S^{-}_{α}, or equivalently $S^{a,-}_{p,q,r}$, in the x,y and z direction, respectively, within the subdomain D^a and the superscript $u = 1$, 2 or 3 discriminates between the three faces $F_{\alpha'}$ for which $S^{-}_{\alpha'} = S^{-}_{\alpha}$. With this labelling we write $[\boldsymbol{Z}^{+}_{ba}]_{\beta\alpha} = \mathcal{Z}^{+}_{ba}(p-p', q-q', r-r'; u; u')$ and

$$[\boldsymbol{Z}^{+}_{ba}\boldsymbol{d}^{+}_{\chi,a}]_{\beta} = \sum_{u'}\sum_{p'}\sum_{q'}\sum_{r'} \mathcal{Z}^{+}_{ba}(p-p', q-q', r-r'; u; u')\, \chi^{a,u',+}_{p',q',r'} d^{a,u'}_{p',q',r'}. \tag{16}$$

After performing a 3D FFT of size $(2P+1)\times(2P+1)\times(2P+1)$, the discrete convolution in this expression is transformed into a simple diagonal multiplication, as is well known. We denote this method, where the total mesh consists of cubic subdomains D^a, as the subdomain FFT method. The computational complexity of an evaluation of (14) in the subdomain FFT method (Fig. 4) is dominated by the calculation of the FFTs of $\boldsymbol{d}^{\pm}_{\chi,a}$ for every subdomain and the diagonal multiplications for every combination of subdomains. Thus it is $\mathcal{O}(c_1 N^{\mathrm{D}} N^{\mathrm{C}} \log N^{\mathrm{C}} + c_2 (N^{\mathrm{D}})^2 N^{\mathrm{C}})$ and the memory use is bounded above by $\mathcal{O}(d_1 (N^{\mathrm{D}})^2 N^{\mathrm{C}} + d_2 N^{\mathrm{D}} N^{\mathrm{C}})$ (for storing the spectra of $\mathcal{Z}^{+}_{ba}$, which are calculated in the setup time of the algorithm, and of $\boldsymbol{d}^{\pm}_{\chi,a}$), where c_1, c_2, d_1 and d_2 are constant prefactors. In practice the memory use is smaller, because some interactions $\mathcal{Z}^{+}_{ba}$ are identical due to translation symmetry.

4. The HF MLFMA

4.1. Basic equations

The Multilevel Fast Multipole Algorithm (MLFMA) is a multilevel extension of the Fast Multipole Method (FMM). In this paper, a vectorial FMM is employed. This means that the mixed potential formulation (3) is replaced by the electric field integral formulation

$$\boldsymbol{E}^{\mathrm{scat}}(\boldsymbol{r}) = -\mathrm{j}\omega\mu_0 \int_{\mathcal{D}} \left(\boldsymbol{I} + \frac{1}{k_b^2}\nabla\nabla\right) G_b(\boldsymbol{r}-\boldsymbol{r}') \cdot \boldsymbol{J}^{\mathrm{scat}}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}' \tag{17}$$

and the diagonal addition theorem for the Green dyadic is used (see [9], chapter 3, and [19]).

The FMM requires a division of the basis functions in a number of non-overlapping FMM-groups. In our implementation the FMM-groups conveniently coincide with the cubic subdomains D^a, which are introduced in Section 2.2. Let $\boldsymbol{E}_a^{\text{scat}}$ be the scattered electric field, caused by the contrast currents in subdomain D^a. The HF FMM computes $\boldsymbol{E}_a^{\text{scat}}$, tested with a basis function $\boldsymbol{\Psi}_\beta$ belonging to subdomain D^b as

$$\int_{\mathcal{D}} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \boldsymbol{E}_a^{\text{scat}}(\boldsymbol{r}) \mathrm{d}\boldsymbol{r} \approx -\frac{jk_b^3}{(4\pi)^2 \epsilon_{\text{b}}} \int_0^\pi \mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\phi \boldsymbol{V}_{\beta,b}(\theta,\phi) \cdot T_{ba}(\theta,\phi) \sin\theta \boldsymbol{U}_a(\theta,\phi). \tag{18}$$

Note that (18) represents one element of $-(\boldsymbol{Z}_{ba}^+ \boldsymbol{d}_{\chi,a}^+ + \boldsymbol{Z}_{ba}^- \boldsymbol{d}_{\chi,a}^-)$ in (15). The diagonal translation operator T_{ba} is calculated as

$$T_{ba}(\theta,\phi) = \sum_{l=0}^{L} (-j)^l (2l+1) h_l^{(2)}(k_{\text{b}} r_{ba}) P_l(\hat{\boldsymbol{k}}(\theta,\phi) \cdot \hat{\boldsymbol{r}}_{ba}), \tag{19}$$

where $h_l^{(2)}$ is the spherical Hankel function of the second kind and order l, P_l is the Legendre function of order l, $r_{ba} = \|\boldsymbol{r}_{ba}\| = \|\boldsymbol{r}_c^b - \boldsymbol{r}_c^a\|$ is the distance between the centers of the subdomains, $\hat{\boldsymbol{r}}_{ba} = \boldsymbol{r}_{ba}/r_{ba}$ and $\hat{\boldsymbol{k}}(\theta,\phi) = \hat{\boldsymbol{x}}\sin\theta\cos\phi + \hat{\boldsymbol{y}}\sin\theta\sin\phi + \hat{\boldsymbol{z}}\cos\theta$ is a vector on the unit sphere. The radiation patterns $\boldsymbol{U}_a$ and $\boldsymbol{V}_{\beta,b}$ are given by

$$\boldsymbol{U}_a(\theta,\phi) = \sum_{\alpha\in I_a^+} \chi_\alpha^+ d_\alpha \int_{S_\alpha^+} \mathrm{e}^{\mathrm{j}\boldsymbol{k}_{\text{b}}\cdot(\boldsymbol{r}'-\boldsymbol{r}_c^a)} (\boldsymbol{I} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \boldsymbol{\Psi}_\alpha(\boldsymbol{r}') \mathrm{d}\boldsymbol{r}' + \sum_{\alpha\in I_a^-} \chi_\alpha^- d_\alpha \int_{S_\alpha^-} \mathrm{e}^{\mathrm{j}\boldsymbol{k}_{\text{b}}\cdot(\boldsymbol{r}'-\boldsymbol{r}_c^a)} (\boldsymbol{I} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \boldsymbol{\Psi}_\alpha(\boldsymbol{r}') \mathrm{d}\boldsymbol{r}' \tag{20}$$

$$\boldsymbol{V}_{\beta,b}(\theta,\phi) = \int_{S_\beta} \mathrm{e}^{-\mathrm{j}\boldsymbol{k}_{\text{b}}\cdot(\boldsymbol{r}-\boldsymbol{r}_c^b)} (\boldsymbol{I} - \hat{\boldsymbol{k}}\hat{\boldsymbol{k}}) \cdot \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \mathrm{d}\boldsymbol{r}, \tag{21}$$

where $\boldsymbol{k}_{\text{b}} = k_{\text{b}}\hat{\boldsymbol{k}}$. The set $I_a^\pm$ in (20) contains indices α of basis functions $\boldsymbol{\Psi}_\alpha$ for which $S_\alpha^\pm$ lies in D^a. Note that these patterns only have transverse components $U_a^u = \boldsymbol{U}_a \cdot \hat{\boldsymbol{u}}$ and $V_{\beta,b}^u = \boldsymbol{V}_{\beta,b} \cdot \hat{\boldsymbol{u}}$ with $u = \phi$ or $u = \theta$. Eq. (18) can be shown to be valid up to arbitrary precision as long as r_{ba} is sufficiently large, i.e. D^a and D^b have to be well separated [9]. In general, this is expressed as $r_{ab} > \beta R$, where β is the separation parameter and R is the radius of the subdomains.

4.2. Numerical implementation

4.2.1. Integration and interpolation

To evaluate the integrals in (18) numerically and to perform the interpolations needed to extend the FMM to the MLFMA, we follow an approach, similar to that of Sarvas [20]. In this approach, the radiation patterns are presented in a Fourier basis rather than with the usual spherical harmonics and interpolations are done with FFTs. The difference is that the FMM in [20] is scalar, while ours is vectorial. We use the transverse components of the radiation patterns, which are not bandlimited in terms of spherical harmonics, in contrast to the cartesian components. However, the functions U_a^u and $V_{\beta,b}^u$ do have exponentially decaying Fourier spectra when their definition domain is extended from $[0,\pi]\otimes[0,2\pi]$ to $[0,2\pi]\otimes[0,2\pi]$ using the formula

$$F(\theta,\phi) = -F(2\pi-\theta,\phi+\pi) \tag{22}$$

for $F = U_a^u$ or $F = V_{\beta,b}^u$, as can easily be verified. The integral in (18) then is replaced by

$$\int_{\mathcal{D}} \boldsymbol{\Psi}_\beta(\boldsymbol{r}) \cdot \boldsymbol{E}_a^{\text{scat}}(\boldsymbol{r}) \mathrm{d}\boldsymbol{r} \approx -\frac{1}{2}\frac{jk_b^3}{(4\pi)^2 \epsilon_{\text{b}}} \int_0^{2\pi} \mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\phi \boldsymbol{V}_{\beta,b}(\theta,\phi) \cdot T_{ba}(\theta,\phi) |\sin\theta| \boldsymbol{U}_a(\theta,\phi), \tag{23}$$

where the definition domain of the spherical function T_{ba} is extended with

$$T_{ba}(\theta,\phi) = T_{ba}(2\pi-\theta,\phi+\pi). \tag{24}$$

Formula (22) thus allows the extension the FFT approach of [20] to the vectorial case. In the Fourier representation, $N_0(2M_0+1)$ samples of the radiation patterns, uniformly spaced in θ and ϕ within the domain $[0,\pi]\otimes[0,2\pi]$, are needed to perform the integration in (23). The numbers M_0 and N_0 depend on the subdomain radius R and the desired accuracy. We chose $M_0 = L_0, N_0 = L_0 + 1$ where L_0 is determined numerically

together with L in (19) on a worst case scenario in such a way that the desired accuracy is achieved with the minimum number of samples. L_0 turns out to be much smaller than L, usually somewhat larger than $\frac{L}{2}$. Further differences with the implementation of [20] involve the recursive calculation of the truncated Fourier spectrum of $T_{ba}(\theta,\phi)\mid\sin\theta\mid$ using a recursion formula for the Legendre function and the use of a 2D extension of Theorem 4.1 in [20] to anterpolate the product $T_{ba}(\theta,\phi)\mid\sin\theta\mid \boldsymbol{U}_a(\theta,\phi)$.

4.2.2. More efficient aggregation and disaggregation

The evaluation of (20) in $N_0(2M_0+1)$ uniformly spaced sample points, referred to as aggregation towards the lowest level, can be cast into a matrix operation:

$$\boldsymbol{\mathcal{U}}_a^u = \boldsymbol{\mathcal{A}}^{+,u}\boldsymbol{d}_{\chi,a}^{+} + \boldsymbol{\mathcal{A}}^{-,u}\boldsymbol{d}_{\chi,a}^{-}, \quad u=\theta \text{ or } \phi. \tag{25}$$

$\boldsymbol{\mathcal{U}}_a^u$ is a $N_0(2M_0+1)$-dimensional vector containing the samples of U_a^u and $\boldsymbol{\mathcal{A}}^{+,u}$ and $\boldsymbol{\mathcal{A}}^{-,u}$ are the aggregation matrices. After interpolation, translation and anterpolation, the incoming pattern $\boldsymbol{R}_b(\theta,\phi)=\sum_a T_{ba}(\theta,\phi)\ \boldsymbol{U}_a(\theta,\phi)\mid\sin\theta\mid$ of every group D^b is multiplied in (23) with $\boldsymbol{V}_{\beta,b}$ and integrated for every $\boldsymbol{\Psi}_\beta$ in D^b. Since the integration is performed numerically by a summation over samples, this can also be written as a matrix operation:

$$\boldsymbol{e}_{b,\mathrm{FMM}}^{\mathrm{scat}} = \boldsymbol{\mathcal{D}}^{\theta}\boldsymbol{\mathcal{R}}_b^{\theta} + \boldsymbol{\mathcal{D}}^{\phi}\boldsymbol{\mathcal{R}}_b^{\phi}, \tag{26}$$

where $\boldsymbol{e}_{b,\mathrm{FMM}}^{\mathrm{scat}}$ contains the weighted scattered field in subdomain D^b, due to all well separated subdomains D^a and $\boldsymbol{\mathcal{R}}_b^u$ contains the samples of $\boldsymbol{R}_b(\theta,\phi)\cdot\hat{\boldsymbol{u}}$. The matrices $\boldsymbol{\mathcal{D}}^\theta$ and $\boldsymbol{\mathcal{D}}^\phi$ are the disaggregation matrices and this step in the algorithm is denoted as disaggregation from the lowest level.

Again the contributions of S_α^+ and S_α^- are separated in (25) in order to make the aggregation matrices independent of the contrast. Because of this and because of the identical geometry of all subdomains, the aggregation and disaggregation matrices are the same for every subdomain and have to be stored only once. This saves a lot of memory compared to a HF MLFMA applied to arbitrary meshes, where these matrices have to be stored per FMM-group or subdomain.

Although using uniform samples in θ and ϕ allows for an elegant FFT interpolation, combining global exact interpolation with efficiency, it is suboptimal with respect to the aggregation toward and disaggregation from the lowest level. The cost of these stages can be reduced by choosing the samples more optimally [21] or by temporarily switching to another, more economic representation of the radiation patterns. The latter approach has been employed in [22], where the cartesian components of the radiation patterns are represented in a spherical harmonics basis on the lowest level. After aggregation to this basis, the uniform samples are still needed to proceed with the diagonal translations and the interpolations towards higher levels, where the patterns are stored in the usual k-space representation. In this paper, a similar strategy is adopted, but we use vector spherical harmonics to represent only the transverse components of the radiation patterns. For example:

$$\boldsymbol{U}_a(\theta,\phi) = \sum_{l=0}^{L_0}\sum_{m=-l}^{l}(\gamma_a)_{lm}\boldsymbol{X}_{lm}(\theta,\phi) + \sum_{l=0}^{L_0}\sum_{m=-l}^{l}(\kappa_a)_{lm}\boldsymbol{\Phi}_{lm}(\theta,\phi). \tag{27}$$

Here, $\boldsymbol{X}_{lm}$ and $\boldsymbol{\Phi}_{lm}=\hat{\boldsymbol{k}}\times\boldsymbol{X}_{lm}$ are vector spherical harmonics as defined in [23]. The multipole coefficients $(\gamma_a)_{lm}$ and $(\kappa_a)_{lm}$ are determined as

$$(\gamma_a)_{lm} = 4\pi j^l\sum_{\alpha\in I_a^+}\chi_\alpha^+ d_\alpha\int_{S_\alpha^+}\mathrm{d}\boldsymbol{r}\boldsymbol{\Psi}_\alpha(\boldsymbol{r})\cdot\boldsymbol{m}_{lm}^{(1)*}(\boldsymbol{r}-\boldsymbol{r}_c^a) + 4\pi j^l\sum_{\alpha\in I_a^-}\chi_\alpha^- d_\alpha\int_{S_\alpha^-}\mathrm{d}\boldsymbol{r}\boldsymbol{\Psi}_\alpha(\boldsymbol{r})\cdot\boldsymbol{m}_{lm}^{(1)*}(\boldsymbol{r}-\boldsymbol{r}_c^a) \tag{28}$$

$$(\kappa_a)_{lm} = 4\pi j^{l-1}\sum_{\alpha\in I_a^+}\chi_\alpha^+ d_\alpha\int_{S_\alpha^+}\mathrm{d}\boldsymbol{r}\boldsymbol{\Psi}_\alpha(\boldsymbol{r})\cdot\boldsymbol{n}_{lm}^{(1)*}(\boldsymbol{r}-\boldsymbol{r}_c^a) + 4\pi j^{l-1}\sum_{\alpha\in I_a^-}\chi_\alpha^- d_\alpha\int_{S_\alpha^-}\mathrm{d}\boldsymbol{r}\boldsymbol{\Psi}_\alpha(\boldsymbol{r})\cdot\boldsymbol{n}_{lm}^{(1)*}(\boldsymbol{r}-\boldsymbol{r}_c^a) \tag{29}$$

where $\boldsymbol{m}_{lm}^{(1)}(\boldsymbol{r}) = j_n(k_0 r)\boldsymbol{X}_{lm}(\hat{\boldsymbol{r}})$ and $\boldsymbol{n}_{lm}^{(1)}(\boldsymbol{r}) = \frac{1}{k_0}\nabla\times\boldsymbol{m}_{lm}^{(1)}(\boldsymbol{r})$, with j_n the spherical Bessel function, are the standing wave vector solutions with zero divergence of the Helmholtz equation as defined in [23]. $\boldsymbol{m}_{lm}^{(1)*}$ and $\boldsymbol{n}_{lm}^{(1)*}$ denote the complex conjugate of $\boldsymbol{m}_{lm}^{(1)}$ and $\boldsymbol{n}_{lm}^{(1)}$ respectively. The advantage of this approach is that we only need

to calculate $2 \times (L_0 + 1)^2$ multipole coefficients to represent $\boldsymbol{U}_a$, which is half the amount of uniform samples that are needed, $2 \times (2L_0 + 1)(L_0 + 1)$. The matrix version of (28) and (29) which now replaces (25) is

$$\boldsymbol{\gamma}_a = \boldsymbol{\mathcal{M}}^+ \boldsymbol{d}^+_{\chi,a} + \boldsymbol{\mathcal{M}}^- \boldsymbol{d}^-_{\chi,a}, \tag{30}$$

$$\boldsymbol{\kappa}_a = \boldsymbol{\mathcal{N}}^+ \boldsymbol{d}^+_{\chi,a} + \boldsymbol{\mathcal{N}}^- \boldsymbol{d}^-_{\chi,a}, \tag{31}$$

where $\boldsymbol{\gamma}_a$ and $\boldsymbol{\kappa}_a$ are vectors containing the multipole coefficients of $\boldsymbol{U}_a$. $\boldsymbol{\mathcal{M}}^{\pm}$ and $\boldsymbol{\mathcal{N}}^{\pm}$ are the new aggregation matrices, which are still independent of the domain index a. Since these matrices are only half as large as the original aggregation matrices in (25), the aggregation to multipole coefficients will be twice as fast. The overall gain factor, however, will be smaller than 2, because we still have to evaluate the multipole expressions for the radiation patterns in the $N_0(2M_0 + 1)$ uniform sample points with (27). However, this can be done efficiently. First of all note that $\boldsymbol{\Phi}_{lm} \cdot \hat{\boldsymbol{\theta}} = -\boldsymbol{X}_{lm} \cdot \hat{\boldsymbol{\phi}}$ and $\boldsymbol{\Phi}_{lm} \cdot \hat{\boldsymbol{\phi}} = \boldsymbol{X}_{lm} \cdot \hat{\boldsymbol{\theta}}$. Then let us rewrite (27) using a block matrix notation:

$$\begin{bmatrix} \boldsymbol{\mathcal{U}}^{\theta}_a \\ \boldsymbol{\mathcal{U}}^{\phi}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mathcal{X}}_{\theta} & -\boldsymbol{\mathcal{X}}_{\phi} \\ \boldsymbol{\mathcal{X}}_{\phi} & \boldsymbol{\mathcal{X}}_{\theta} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_a \\ \boldsymbol{\kappa}_a \end{bmatrix}. \tag{32}$$

It appears that four matrix–vector multiplications (with matrices of much smaller dimension than the aggregation matrices) have to be carried out. This can be avoided, however, since (32) can be diagonalized:

$$\begin{bmatrix} \boldsymbol{\mathcal{U}}^{\theta}_a \\ \boldsymbol{\mathcal{U}}^{\phi}_a \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ j\boldsymbol{I} & -j\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{\theta} - j\boldsymbol{\mathcal{X}}_{\phi} & 0 \\ 0 & \boldsymbol{\mathcal{X}}_{\theta} + j\boldsymbol{\mathcal{X}}_{\phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_a - j\boldsymbol{\kappa}_a \\ \boldsymbol{\gamma}_a + j\boldsymbol{\kappa}_a \end{bmatrix}, \tag{33}$$

where $\boldsymbol{I}$ represents the unit matrix of dimension $N_0(2M_0 + 1)$. This way, only two matrix–vector products and some simple recombinations remain. Note that in [22] three matrix–vector products are required to calculate the three cartesian components of the radiation patterns. Furthermore the summations over m in (27) can be carried out efficiently by using FFTs. It can be shown that $\boldsymbol{X}_{lm} \cdot \hat{\boldsymbol{\theta}}$ and $\boldsymbol{X}_{lm} \cdot \hat{\boldsymbol{\phi}}$ depend on ϕ only through a factor $\mathrm{e}^{\mathrm{j}m\phi}$, such that the typical summation is of the form

$$\sum_{l=0}^{L_0} \sum_{m=-l}^{l} F_{lm}(\theta) \mathrm{e}^{\mathrm{j}m\phi_q} \gamma_{lm} = \sum_{m=-L_0}^{L_0} \sum_{l=|m|}^{L_0} F_{lm}(\theta) \mathrm{e}^{\mathrm{j}m\phi_q} \gamma_{lm} \tag{34}$$

$$= \sum_{m=-L_0}^{L_0} \mathrm{e}^{\mathrm{j}m\frac{2\pi}{2L_0+1}q} \left(\sum_{l=|m|}^{L_0} F_{lm}(\theta) \gamma_{lm} \right), \tag{35}$$

where we have assumed the form $\phi_q = q\frac{2\pi}{2L_0+1} = q\frac{2\pi}{2M_0+1}$ for the samples in the ϕ-direction. The outer summation in (35) is a discrete Fourier transform and hence can be calculated by an FFT.

$\boldsymbol{V}_{\beta,b}$ is also expanded in multipoles and by substituting this expansion in (18) and by interchanging the integration and the summation in the multipole expansion, a disaggregation procedure with multipoles is readily obtained similar to the aggregation procedure outlined above. The gain in speed obtained by the aggregation and disaggregation via multipoles will be discussed in Section 6.2.

A last improvement to the aggregation towards and disaggregation from the lowest level is a purely technical one. The matrix operations (30) and (31) have to be carried out for every subdomain. Since the aggregation and disaggregation matrices are the same for all the groups, the vectors $\boldsymbol{d}^+_{\chi,a}$ for all a, for example, can be stored columnwise in a large matrix with N^{D} columns. This matrix can be multiplied as a whole with the appropriate aggregation matrix using Level 3 Basic Linear Algebra Subprograms (BLAS) [24,25], which reduces the CPU time considerably. This will also be illustrated in Section 6.2.

5. The hybrid MLFMA–FFT method

The hybrid MLFMA–FFT method consists of applying the HF MLFMA of Section 4 to the cubic mesh of Section 2.2 and treating the interactions (15) between subdomains that are not well separated – the near interactions – with the use of FFTs (16). The resulting method is an improvement with respect to both the HF

MLFMA and the subdomain FFT method. With regard to the MLFMA, treating the near interactions with FFTs is rewarding, because the $\mathcal{O}((N^C)^2)$ near interactions of the HF MLFMA are replaced by more efficient $\mathcal{O}(N^C)$ diagonal multiplications (Fig. 5). With regard to the subdomain FFT method, the hybrid MLFMA–FFT method replaces the diagonal multiplication in the Fourier domain with the diagonal translation of the FMM for well separated subdomains. Because the dimension of the former is proportional to N^C and thus to the volume of the subdomains, while the dimension of the latter is generally smaller (only proportional to the surface area of the subdomains when these are large enough), this can reduce the computation time and memory use, even in a two-level FMM. Furthermore, in the multilevel scheme, the computational complexity of the MLFMA–FFT method is $\mathcal{O}(N)$ to $\mathcal{O}(N\log^2 N)$, while the subdomain FFT method scales as $(N^D)^2 N^C = N^2/N^C$ for a fixed size of the subdomains.

For a more in depth study of the method, the subdomain size is the most important parameter. For a fixed separation parameter β (which determines the reachable accuracy) it determines the height of the MLFMA tree (we add levels until there are no more far interactions on the level to add) and the efficiency of the multilevel scheme. To investigate the influence of the subdomain size, consider first a very large dense scattering configuration. In such a situation, the number of near interactions for a given subdomain (i.e. the number of subdomains with a center that lies within a radius βR from the center of the considered subdomain) is independent of the subdomain size. To see what happens when the subdomain size is increased, consider a doubling of this size:

- The number of cells N^C per subdomain is multiplied by 8.
- The number of subdomains is divided by 8.
- The cost of calculating the FFTs for every subdomain increases slightly from $N^C \ln(N^C)$ to $N^C \ln(8N^C)$.
- The cost of the diagonal multiplications in the Fourier domain for the near interactions is unaltered, since the increase in cells per subdomain and the decrease in number of subdomains balance each other.
- The cost of the translations and interpolations/anterpolations is seriously reduced, because we lose a level and nothing changes for the remaining levels.
- The cost of the aggregation toward and disaggregation from the lowest level is increased, because L_0 is increased. For subdomain sizes of about one wavelength or less, the increase is practically negligible, but for larger subdomains, L_0 increases linearly with the subdomain size.

One can conclude that unless the subdomains become very large ($k_b R \gg 1$), an increase in subdomain size reduces the cost and memory usage. For sparse configurations, the behaviour is less predictable, but apart from some exotic configurations the same conclusion stands. This is an important difference with the original MLFMA, where the cost of the near interactions is multiplied by 8 when the subdomain size is doubled. Therefore, in the MLFMA there generally is an optimal subdomain size, smaller than the background wave-

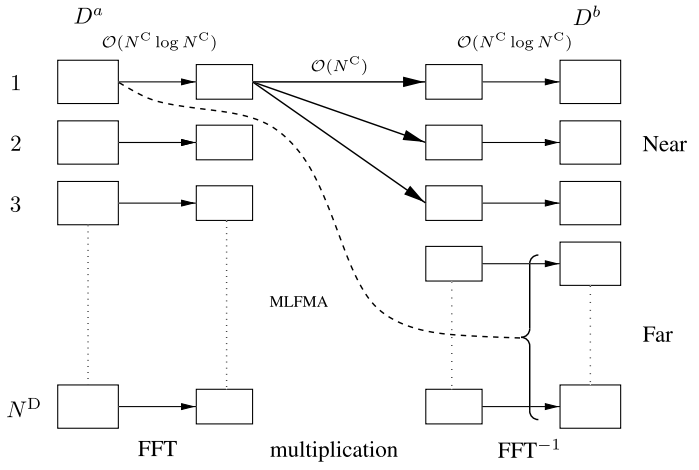

Fig. 5. Illustration of the computational complexity of the MLFMA–FFT method.

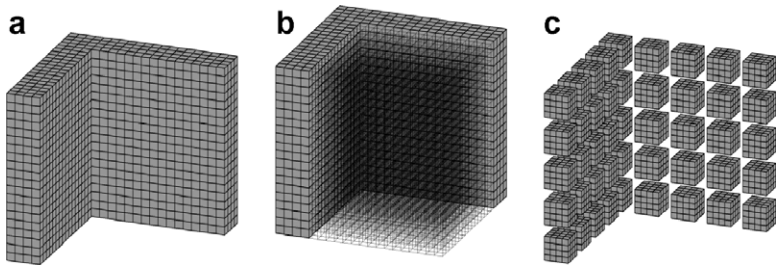

Fig. 6. Two ways of handling a sparse cubic mesh (a): extension to the bounding box D^C (b), and division in identical cubic subdomains (c).

length, which balances the load between near and far interactions optimally, but the use of FFTs shifts this optimum in the MLFMA–FFT method to much larger subdomains.

Our main goal is to examine for which type of configurations the MLFMA–FFT method outperforms the FFT method in terms of CPU time or memory consumption. On dense problems it is well known that the MLFMA, and thus also the MLFMA–FFT method, is $\mathcal{O}(N)$. However, it appears from numerical tests that despite this lower computational complexity, the MLFMA–FFT method is slower than the FFT method even for fairly large problems, due to the small prefactor of the FFT method. It will be shown in Section 6, though, that it uses substantially less memory. Another situation is encountered when considering sparse scattering configurations, such as the one in Fig. 6(a). When the FFT method is used to calculate the scattering from this geometry, the cubic grid has to be extended to the bounding box $\mathcal{D}^C$ of the domain $\mathcal{D}$ as in Fig. 6(b). This implies that the CPU time and the memory needed for the calculation of (14) are the same as for a dense configuration in $\mathcal{D}^C$. In this case the MLFMA–FFT or even the subdomain FFT method can yield a faster matrix–vector multiplication because they discretize the geometry more economically (Fig. 6(c)). If we do not want the grid to extend over regions of empty space, the maximal subdomain size is dictated by the sparsity of the configuration. Combining this with the main conclusion of the previous paragraph, we state the following rule of thumb: if the maximal subdomain size, determined by the sparsity of the geometry, is not much larger then a wavelength, use this maximal subdomain size and the MLFMA–FFT method. For very large maximal subdomain sizes, the aggregation and disaggregation steps become unwieldy and it might be appropriate to use the subdomain FFT on the maximal subdomains or the MLFMA–FFT method with smaller subdomains, depending on the specific geometry. In general, however, the MLFMA–FFT method will consume less memory on any large problem, as will be demonstrated in the next section.

6. Validation and performance analysis

In this section, the proposed MLFMA–FFT method is validated and its performance is investigated and compared to that of the FFT-method and the HF MLFMA on a number of test cases. All computations are carried out in double precision arithmetics on a 64 bit computer with 2 GHz Dual Core AMD Opteron processor and 8 GB RAM. All FFTs are computed using FFTW, the Fastest Fourier Transform in the West, a collection of fast C routines for computing the discrete Fourier transform [13]. No parallelizing or multithreading of any kind are used.

6.1. Validation

The scattering from a homogeneous sphere is considered to first validate the subdomain FFT method. The sphere has a radius $R = \lambda_b$ of one background wavelength and a permittivity $\epsilon = (2 - 2j)\epsilon_b$ and is illuminated with an x-polarized plane wave traveling in the $+z$-direction:

$$\boldsymbol{E}^{\text{inc}}(\boldsymbol{r}) = \mathrm{e}^{-\mathrm{j}k_b z}\hat{\boldsymbol{x}}. \tag{36}$$

The sphere is contained in a cubic domain $\mathcal{D}$ with side $2\lambda_b$. The grid on this domain has a cell size $\delta = 0.05\lambda_b$, which results in 196 800 unknowns. Such a fine grid is chosen to reduce the staircasing error. We solve the VIE

in two ways: firstly we consider $\mathcal{D}$ as one cubic subdomain, which means that we employ a classical FFT method, and secondly we divide $\mathcal{D}$ in 64 subdomains and apply the subdomain FFT method. Both methods needed 39 BICGSTAB iterations to converge to an accuracy of 10^{-6}. Fig. 7 compares the scattered fields with the analytical solution provided by the MIE series [26]. The agreement is very good.

We now compare the scattered fields computed by the MLFMA–FFT method and the subdomain FFT-method for the sparse scattering configuration of Fig. 8. A homogeneous sphere with a radius $R = 2\lambda_b$ and

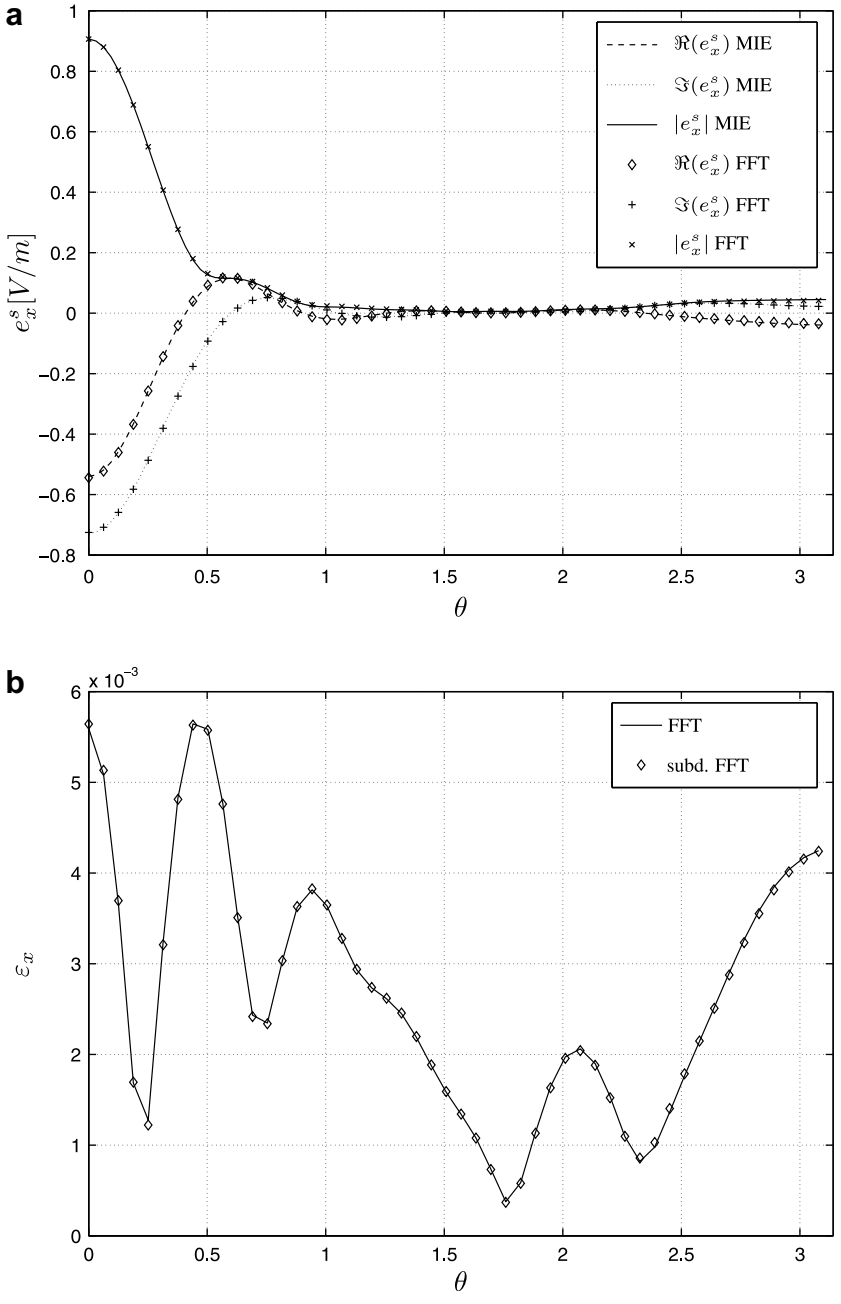

Fig. 7. Scattering from a homogeneous sphere with radius $R = \lambda_b$ and permittivity $\epsilon = (2 - 2j)\epsilon_b$: comparison between MIE series and results from the FFT and subdomain FFT methods. Figure (a) shows the x-component of the scattered field on a semicircle $S = \{\boldsymbol{r} : \boldsymbol{r} = R_m \sin\theta\hat{\boldsymbol{x}} + R_m \cos\theta\hat{\boldsymbol{z}}\}$, with $R_m = 4\lambda_b$ and $\theta \in [0, \pi[$. Figure (b) shows the error defined as $\varepsilon_x(\boldsymbol{r}) = \|\mathrm{e}_x^s(\boldsymbol{r}) - \mathrm{e}_{x,\mathrm{MIE}}^s(\boldsymbol{r})\|/\max_{r\in S}\|\mathrm{e}_{x,\mathrm{MIE}}^s(\boldsymbol{r})\|$ for both the FFT and subdomain FFT methods.

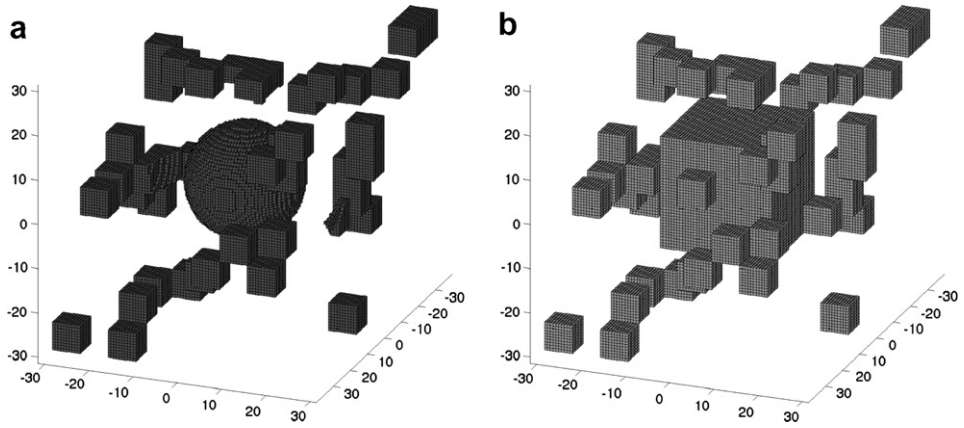

Fig. 8. A homogeneous sphere with $R = 2\lambda_b$ and $\epsilon = (2 - 2j)\epsilon_b$, surrounded by 50 particles with side λ_b and $\epsilon = 1.5\epsilon_b$, randomly distributed over a cubic domain $\mathcal{D}^C$ with side $10\lambda_b$. Figure (a) shows the actual scatterers and figure (b) represents the mesh used by the MLFMA–FFT and subdomain FFT methods. The cell size is $\delta = 0.1\lambda_b$.

a permittivity $\epsilon = (2 - 2j)\epsilon_b$ is surrounded by 50 cubes with side λ_b and permittivity $\epsilon = 1.5\epsilon_b$, that are randomly distributed in a cubic domain $\mathcal{D}$ with side $10\lambda_b$. The cell size is $0.1\lambda_b$. This problem yields 344 100 unknowns and is solved by both methods in 67 iterations to an accuracy of 10^{-6}. The results are shown in Fig. 9. The parameters for the MLFMA are chosen such that the relative error on the FMM-formula (18) is less than 10^{-5} and Fig. 9(b) shows that the relative difference between both solutions stays below this value.

6.2. Performance analysis

We first demonstrate the acceleration of the aggregation towards and the disaggregation from the lowest level, introduced by the techniques of Section 4.2.2, in Table 1. Three sizes of the subdomains ($0.5\lambda_b$, $0.8\lambda_b$ and λ_b) are considered. The values of L_0 and N^a determine the size of the aggregation and disaggregation matrices. N^a is the number of basis functions in a subdomain if it is filled with a uniform cubical grid with $\delta = 0.1\lambda_b$. L_0 is determined together with L (not shown in Table 1) such that the relative error on the FMM-formula (18) does not exceed 10^{-5} when the separation parameter β is set to 4. It can be seen that by performing the aggregation towards and disaggregation from the lowest level via multipoles (M) instead of with uniform samples (US), the CPU time for these stages can be reduced by 40–45%. Moreover, with the use of Level 3 BLAS routines (MB) the total reduction factor ranges from 3 for a subdomain size of $0.5\lambda_b$ to 5 for a subdomain size of λ_b. The number of subdomains N^D, once it is large enough, does not seem to have a significant influence on these gain factors.

Next, the performances in terms of CPU time and storage requirements of all the methods described in this paper are investigated and compared. Table 2 shows the CPU time for one evaluation of (14) and the memory needed to solve the VIE for a number of test geometries. These geometries are all contained in a cubic bounding box $\mathcal{D}^C$ with side $10\lambda_b$ and are meshed with a uniform cubic grid with cell size $\delta = 0.1\lambda_b$. For the FFT method, the complete bounding box $\mathcal{D}^C$ has to be discretized, irrespective of the actual permittivity profile and geometry inside $\mathcal{D}^C$. This yields the test grid "full". The subdomain FFT method and the MLFMA–FFT method can be used on sparse subdomain grids. The test grid "clutN^P" thus refers to grids like the one of Fig. 8(b), where N^P is the number of particles that surround a central cube of side $4\lambda_b$. All particles are cubes with side λ_b and coincide with one subdomain each in all test grids, except for "clut45/2". The test grid "clut45/2" is identical to "clut45", but the size of its subdomains is twice as small ($0.5\lambda_b$) and hence their number is eight times larger than with "clut45". Finally the test grid "corner" indicates a mesh like the one in Fig. 6(a) with a wall thickness of λ_b and outer dimensions of $10\lambda_b$. The subdomains for the test grids "full" and "corner", when using MLFMA–FFT or subdomain FFT, also have a size λ_b.

We first note that the subdomain FFT method is less efficient than the FFT method in case of dense geometries like the test grid "full", due to a higher complexity. In fact, on the test grid "full" and on all other test

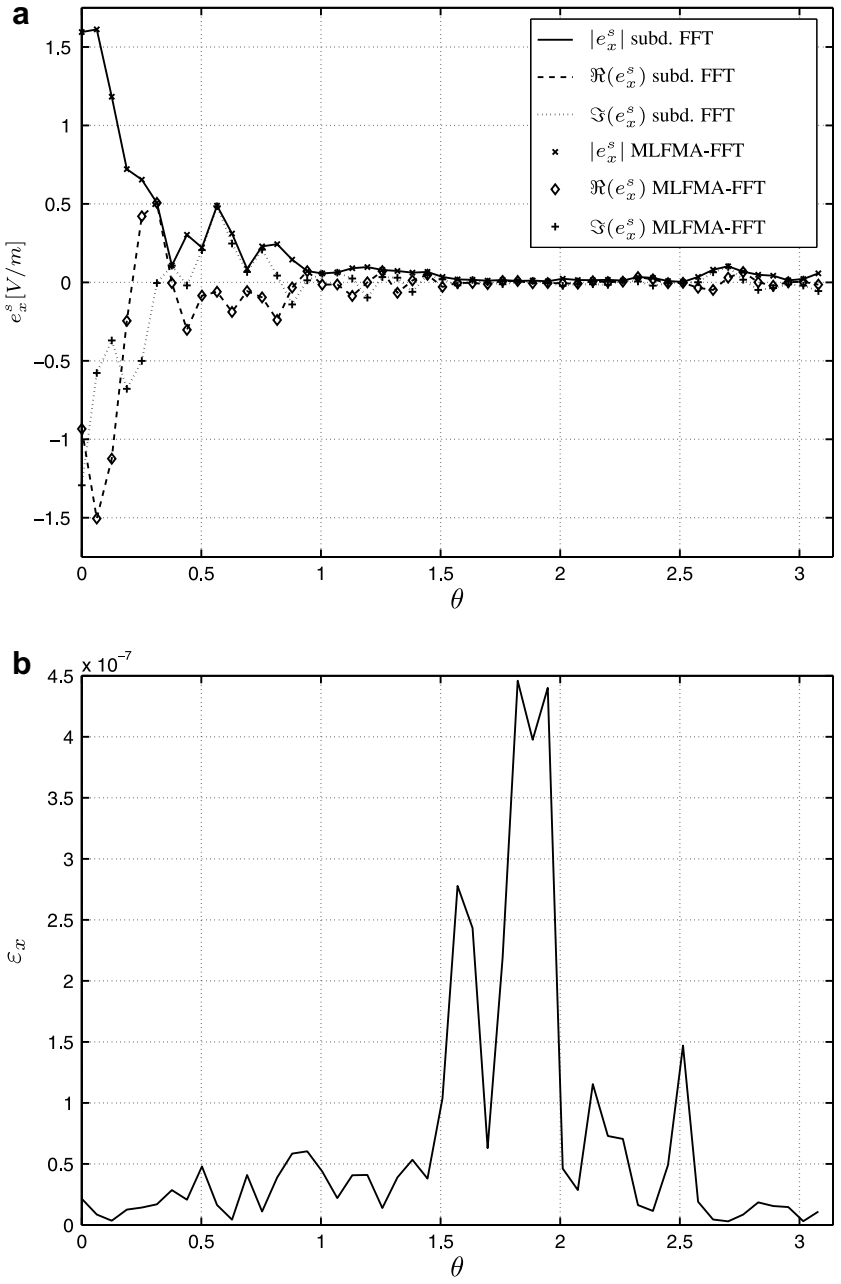

Fig. 9. Scattering from the scatterers depicted in Fig. 8. Figure (a) shows the x-component of the scattered field on a semicircle $S = \{\boldsymbol{r} : \boldsymbol{r} = R_m \sin\theta\hat{\boldsymbol{x}} + R_m \cos\theta\tilde{\boldsymbol{z}}\}$, with $R_m = 10\lambda_b$ and $\theta \in [0, \pi[$. Figure (b) shows the error defined as $\varepsilon_x(\boldsymbol{r}) = \|\mathrm{e}^s_{x,2}(\boldsymbol{r}) - \mathrm{e}^s_{x,1}(\boldsymbol{r})\|/\|\mathrm{e}^s_{x,1}(\boldsymbol{r})\|$ between the subdomain FFT method ($\mathrm{e}^s_{x,1}$) and the hybrid MLFMA–FFT method ($\mathrm{e}^s_{x,2}$).

grids except for "clut45", the subdomain FFT method could not even be used, because it required more memory than the available 8 GB. On the sparse test grid "clut45", the subdomain FFT method yields a faster matrix–vector product than the FFT method, but even here the memory requirements are close to the limit.

When we compare the MLFMA–FFT method with the subdomain FFT method on the test grid "clut45", the gain in CPU time and especially in memory use of the hybrid method is obvious. In comparison to the FFT method (test grid "full"), the MLFMA–FFT method clearly performs better on all sparse test grids. The matrix–vector multiplication is faster and the reduction of the storage needs is even more explicit. Even on the dense problem of test grid "full" the MLFMA–FFT method requires less memory than the FFT-

Table 1
Comparison of CPU times (in seconds) for the aggregation and disaggregation towards and from the lowest level using uniform samples (US), multipoles (M) and multipoles and Level 3 BLAS (MB)

Subdomain size (in λ_b)	L_0	N^a	N^D	US	M	MB
0.5	11	450	1000	5.00	3.16	1.66
0.5	11	450	500	2.62	1.61	0.85
0.8	15	1728	1000	45.88	24.64	9.32
0.8	15	1728	500	17.74	10.03	4.70
1.0	17	3300	1000	112.34	61.25	21.29
1.0	17	3300	500	56.26	30.77	10.79

Results are given for different subdomain sizes and for two values of N^D, the number of subdomains. The values of L_0 yield an FMM-accuracy of 10^{-5}.

Table 2
Comparison between the different methods in terms of memory requirements and CPU-time per evaluation of (14)

Problem	Method	f_V	N	CPU-time	Memory
Full	FFT	1.0	3,030,000	38.92 s	+7 GB
	MLFMA–FFT	1.0	3,030,000	310.88 s	2951 MB
clut45	MLFMA–FFT	0.07	343,900	12.91 s	940 MB
	subdomain FFT	0.07	343,900	29.87 s	>7.3 GB
clut45/2	MLFMA–FFT	0.07	343,900	32.04 s	720 MB
	MLFMA	0.07	343,900	142.89 s	1570 MB
clut87	MLFMA–FFT	0.11	487,300	20.83 s	1059 MB
clut127	MLFMA–FFT	0.15	610,700	25.12 s	1156 MB
clut173	MLFMA–FFT	0.19	758,400	34.38 s	1267 MB
corner	MLFMA–FFT	0.18	591,900	19.33 s	943 MB

f_V is the volume fraction of the scatterers in the surrounding cube $\mathcal{D}^C$ with side $10\lambda_b$. Whenever the MLFMA or the subdomain FFT method is not mentioned for one of the test cases, this means that the available memory was not sufficient for that method.

method, but it is defeated in terms of CPU time. Note that most of the +7 GB used by the FFT-method is needed to store the FFT-vectors and is therefore not affected by how sparse the actual permittivity profile on the grid is. Because of the $\mathcal{O}(N)$ storage complexity of both the MLFMA–FFT method and the FFT method on dense problems like the test grid "full", the hybrid method will always require less memory than the FFT method on electrically large problems.

Finally the MLFMA–FFT method and the MLFMA are compared on the same cubic mesh and using the same parameters. The MLFMA exploits the symmetry to reuse the aggregation and disaggregation matrices. When not doing so, the aggregation and disaggregation matrices of only 78 subdomains with side λ_b could be stored in the available 8 GB of RAM. However, even the symmetry adjusted MLFMA could only be used on the sparsest test grid "clut45/2" with the smallest subdomains, since the other test grids required too much memory to store the near interaction matrices. For this test grid, the hybrid MLFMA–FFT is almost 5 times as fast as the MLFMA and memory requirements are less than half. It follows that the hybrid MLFMA–FFT method is applicable to a much wider range of volumetric problems than the MLFMA and that it also performs better. The test case "clut45/2" illustrates that dividing the mesh in smaller subdomains beyond what is necessary to account for the sparsity of the scatterers is not beneficial in terms of CPU time of the MLFMA–FFT method. On this example it does reduce the memory use, but this behaviour strongly depends on the geometry and is not a fundamental property.

7. Conclusions

In this paper, a new hybrid MLFMA–FFT method was presented. The method combines the advantages of the HF MLFMA and the FFT method on volumetric scattering problems which can be meshed using a

uniform grid. A flexible subdomain meshing of sparse scatterers was introduced and the symmetry in this mesh was exploited to make some of the more time and memory consuming stages in the HF MLFMA more efficient. Also, a novel approach employing vector spherical harmonics to represent the radiation patterns on the lowest level in combination with an FFT interpolation scheme for the vectorial MLFMA was introduced. It was finally shown that the resulting hybrid method is a valuable supplement to the existing fast methods, because it is more efficient on sparse scattering configurations and it can easily be used as a regular FFT solver on dense problems. Furthermore, when memory is an issue, the method's low storage requirements provide a means to tackle very large problems which would otherwise be out of reach.

Acknowledgment

Jürgen de Zaeytijd is a Research Assistant of the Fund for Scientific Research – Flanders (FWO-Vlaanderen). The work of Ignace Bogaert is supported by a doctoral grant from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

[1] T.K. Sarkar, E. Sarvas, S.M. Rao, Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies, IEEE Trans. Antennas Propagat. 34 (5) (1986) 635–640.
[2] C.C. Su, Electromagnetic scattering by a dielectric body with arbitrary inhomogeneity and anisotropy, IEEE Trans. Antennas Propagat. 37 (3) (1989) 384–389.
[3] P. Zwamborn, P.M. van den Berg, The three-dimensional weak form of the conjugate gradient FFT method for solving scattering problems, IEEE Trans. Microw. Theory Tech. 40 (9) (1992) 1757–1766.
[4] C.C. Su, The three-dimensional algorithm of solving the electric field integral equation using face-centered node points, conjugate gradient method, and FFT, IEEE Trans. Microw. Theory Tech. 41 (6) (1993) 510–515.
[5] H. Gan, W.C. Chew, A discrete BCG–FFT algorithm for solving 3D inhomogeneous scatterer problems, IEEE Trans. Antennas Propagat. 9 (10) (1995) 1339–1357.
[6] R. Coifman, V. Rokhlin, S. Wandzura, The fast multipole method for the wave equation: a pedestrian prescription, IEEE Antennas Propagat. Mag. 35 (1993) 7–12.
[7] C.C. Lu, W.C. Chew, A multilevel algorithm forsolving boundary integral equations of wave scattering, Microw. Opt. Technol. Lett. 7 (10) (1994) 466–470.
[8] J.M. Song, C.C. Lu, W.C. Chew, Multilevel fast-multipole algorithm for electromagnetic scattering by large complex objects, IEEE Trans. Antennas Propagat. 45 (10) (1997) 1488–1493.
[9] W.C. Chew, J. Jin, E. Michielssen, J. Song, Fast and Efficient Algorithms in Computational Electromagnetics, Artech House, Boston, 2001.
[10] J.S. Zhao, W.C. Chew, Three dimensional multilevel fast multipole algorithm from static to electrodynamic, Microw. Opt. Technol. Lett. 26 (1) (2000) 43–48.
[11] B. Hu, W.C. Chew, Fast inhomogeneous plane wave algorithm for scattering from objects above the multilayered medium, IEEE Trans. Geosci. Remote Sens. 39 (5) (2001).
[12] E. Darve, P. Havé, Afast multipole method formaxwell equations stable at all frequencies, Phil. Trans. Roy. Soc. Lond. A 362 (1816) (2004) 603–628.
[13] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2) (2005) 216–231.
[14] D.H. Schaubert, D.R. Wilton, A.W. Glisson, A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies, IEEE Trans. Antennas Propagat. 32 (1) (1984) 77–85.
[15] A.F. Peterson, S.L. Ray, R. Mittra, Computational Methods for Electromagnetics, Wiley IEEE Press, 1997.
[16] D.R. Wilton, S.M. Rao, A.W. Glisson, D.H. Schaubert, O.M. Al-Bundak, C.M. Butler, Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains, IEEE Trans. Antennas Propagat. 32 (3) (1984) 276–281.
[17] J. De Zaeytijd, A. Franchois, C. Eyraud, J.M. Geffrin, Full-wave three-dimensional microwave imaging with a regularized Gauss–Newton method theory and experiment, IEEE Trans. Antennas Propagat. 55 (11) (2007) 3279–3292.
[18] R.D. da Cunha, T. Hopkins, The parallel iterative methods (PIM) package for the solution of systems of linear equations on parallel computers, Appl. Numer. Math. 19 (1–2) (1995) 33–50.
[19] W.C. Chew, S. Koc, J.M. Song, C.C. Lu, E. Michielssen, A succinct way to diagonalize the translation matrix in three dimensions, Microw. Opt. Technol. Lett. 15 (3) (1997) 144–146.
[20] J. Sarvas, Performing interpolation and anterpolation entirely by fast Fourier transform in the 3-D multilevel fast multipole algorithm, SIAM J. Numer. Anal. 41 (6) (2003) 2180–2196.
[21] O.M. Bucci, C. Gennarelli, C. Savarese, Optimal interpolation of radiated fields over a sphere, IEEE Trans. Antennas Propagat. 39 (11) (1991) 1633–1643.
[22] T.F. Eibert, A diagonalized multilevel fast multipole method with spherical harmonics expansion of the k-space integrals, IEEE Trans. Antennas Propagat. 53 (2) (2005) 814–817.

[23] R.C. Wittman, Spherical wave operators and the translation formulas, IEEE Trans. Antennas Propagat. 36 (8) (1988) 1078–1087.
[24] C.L. Lawson, R.J. Hanson, D. Kincaid, F.T. Krogh, Basic linear algebra subprograms for FORTRAN usage, ACM Trans. Math. Softw. 5 (1979) 308–323.
[25] L.S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, R.C. Whaley, An updated set of basic linear algebra subprograms (BLAS), ACM Trans. Math. Softw. 28 (2) (2002) 135–151.
[26] J. Van Bladel, Electromagnetic Fields, Hemisphere Publishing Corporation, New York, 1985.