Progress In Electromagnetics Research, PIER 55, 47–78, 2005

TRANSLATION PROCEDURES FOR BROADBAND MLFMA

H. Wallén and J. Sarvas

Electromagnetics Laboratory
Helsinki University of Technology
P.O. Box 3000, FI-02015 TKK, Finland

Abstract—The multilevel fast multipole algorithm (MLFMA) is used in computing acoustic and electromagnetic fields with integral equation methods. The traditional MLFMA, however, suffers from a low-frequency breakdown that effectively limits the minimum division cube side length to approximately one wavelength. To overcome this low-frequency breakdown and get a broadband MLFMA, we propose an efficient and relatively straightforward implementation of the field translations based on the spectral representation of the Green's function. As an alternative we also consider the so called uniform MLFMA, which has a lower computational cost but limited accuracy. We consider the essential implementation details and finally provide numerical examples to demonstrate the error controllability of the translations.

1 Introduction

2 Representations for the Green's Function

3 Implementation of the Translation Procedures
- 3.1 Propagating Part of the Field
- 3.2 Evanescent Part of the Field
  - 3.2.1 Quadrature Rules and Outer-to-Inner Translation
  - 3.2.2 Interpolation and Anterpolation
- 3.3 UMLFMA
  - 3.3.1 Integral Equation for the Translation Function
  - 3.3.2 Interpolation and Anterpolation
- 3.4 Broadband Combinations

4 Numerical Tests for Error Control

5 Conclusions

References

1. INTRODUCTION

The multilevel fast multipole algorithm (MLFMA) [1, 2] is used in computing acoustic and electromagnetic fields with integral equation methods. More specifically, assuming that we are solving an integral equation using an iterative solver, we can use the MLFMA to compute the needed matrix-vector products efficiently without assembling the system matrix. In the matrix-vector product we essentially compute the field due to N weighted basis functions or sources. By grouping the sources in $\mathcal{O}(\log N)$ levels in a tree-like manner, and always computing the interactions between the groups at the coarsest possible level, the number of interactions can be reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. We still need to compute nearby interactions in the normal direct way, but most of the interactions are computed faster, and for high-frequency applications the computational cost of the matrix-vector product decreases from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$.

To implement the MLFMA, we begin by enclosing the computational domain in a large cube, which is then recursively divided into eight sub-cubes until the smallest division cubes contain a small number of sources each. Only the non-empty cubes are stored in a tree-like data-structure. Then we compute the matrix-vector product in two sweeps. First, in the *aggregation* steps we combine the outgoing field representations from smaller division cubes to larger ones. Then, in the *disaggregation* steps we distribute the incoming or local field representations from larger division cubes to smaller ones. The interactions are also computed in the disaggregation steps by transforming outgoing representations into incoming representations in non-nearby division cubes.

The key procedures in a 3-D MLFMA are the field translations. In the aggregation steps the outgoing fields from smaller division cubes are interpolated and shifted to outgoing fields in a larger parental cube; this is the *outer-to-outer* translation. In the disaggregation steps the incoming field in a parental division cube is anterpolated and shifted to incoming fields in the sub-cubes; this the *inner-to-inner* translation. In the *outer-to-inner* translation an outgoing field in a division cube is transformed to an incoming field in a well-separated cube of the same level.

We assume that the reader is familiar with the traditional MLFMA, as described in detail in the book by Chew et al. [2]. In this

paper we mainly consider the needed translation procedures, and point out the essential implementation details, when replacing the traditional translation procedures with broadband ones. We only treat the scalar case, but all results are easily carried over to the vector case, too.

In this paper we number the levels relative to the wavelength instead of relative to the entire object under consideration, although the latter convention seems more common. In particular, the division cubes at level l have side length $a_l = 2^l a_0$, with a chosen a_0 and a suitable range of integers l. Thus, by higher levels or larger l we always mean larger division cubes compared with the wavelength.

Efficient MLFMA methods are based on plane-wave expansions of the Green's function

$$G(\mathbf{r} - \mathbf{r}') = \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{4\pi\,|\mathbf{r} - \mathbf{r}'|}. \tag{1}$$

In the original high-frequency MLFMA such a presentation is given by Rokhlin's translation formula [3, 4], which, however, has the drawback that the resulting outer-to-inner translation looses its accuracy if the side length of the division cube is essentially less than the wavelength. This is the well-known *low-frequency breakdown*. To avoid this problem and to get a *broadband MLFMA*, which is valid for all division levels and frequencies, one has to use outer-to-inner translation methods based on the *spectral representation*, also called the *inhomogeneous plane-wave expansion*, of the Green's function [5–8]. This representation is direction-dependent and can be split into two parts: a propagating part and an evanescent part. The spectral representation avoids the low-frequency breakdown but has the drawback that six expansions must be used to cover all translation directions for the evanescent part.

The need to develop an efficient broadband MLFMA has given arise to much of recent research on that problem. In the approach of Greengard, Huang, Rokhlin and Wadzura [6] the outer-to-inner translation is based on the spectral representation of the Green's function and on an efficient generalized Gaussian quadrature rule for the evanescent part by Yarvin and Rokhlin [9]; they perform other translations by multipole series, which, however, become rather elaborate with non-static fields. Additionally, the multipole series representations become inefficient for large division cubes with respect to the wavelength, and thus one needs to switch to the traditional MLFMA, or some other representation, at higher levels to get a broadband method.

Jiang and Chew [7] present a broadband MLFMA, where the outer-to-inner translation is performed by the spectral representation of the Green's function, but the sampling of the evanescent part is

based on a less efficient quadrature than the generalized Gaussian one; other translations are performed by an polynomial interpolation method. To improve the efficiency at higher levels, they also partially extrapolate the evanescent part from the propagating part, and eventually when the division cube side length is large enough compared with the wavelength they extrapolate the evanescent part completely from the propagating part.

The approach of Darve and Havé [8] is similar to that of Greengard et al. but they improve the efficiency by using the spectral representation also for the outer-to-outer and inner-to-inner translations. For the propagating part, they embed the direction-dependency into the propagating translation function and smooth the translation function using spherical harmonics. For the evanescent part they represent the fields using truncated series of singular functions arising from the singular value decomposition of the evanescent exponential translation function. The resulting method is rather elaborate.

Xuan, Zhu, Adams and Gedney [10] propose a broadband method, called the uniform MLFMA (UMLFMA), which is essentially different from the above ones; the outer-to-inner translation is performed by a modification of the Rokhlin translation formula and other translations are carried out efficiently by interpolation matrices. The method is, unlike the previous approaches, direction-independent and, therefore, of low cost. However, its accuracy is limited.

In this paper we propose a broadband method, which is based on the spectral representation of the Green's function and is designed to combine the best features of the above approaches. For the propagating part (real θ) we use an entirely FFT-based approach, and embed the direction dependency into the translation function, thus needing only one far-field pattern and plane-wave expansion per division cube. For the evanescent part we need six expansions to cover all directions, and we use a highly efficient generalized Gaussian quadrature rule for the complex θ-direction. A novel feature is the use of a simple least-squares interpolation procedure in the complex θ-direction leading to efficient interpolation and anterpolation matrices. Our method is error controllable but at low accuracy levels it is more costly than the UMLFMA.

In the proposed form, the UMLFMA appears not to include the static case with $k = 0$. In this paper we also fill that gap by presenting a modification of the UMLFMA which more explicitly includes the static case. We complete the paper by presenting numerical examples where we test and compare the accuracy and error controllability of our method with those of the UMLFMA.

2. REPRESENTATIONS FOR THE GREEN'S FUNCTION

The high-frequency fast multipole method (FMM) [3, 4] and the original multilevel fast multipole algorithm (MLFMA) [1, 2] are based on Rokhlin's translation formula, representing the Green's function (1) as a plane-wave expansion

$$G(\mathbf{D}+\mathbf{d}) \approx \frac{ik}{(4\pi)^2} \int_{-\pi}^{\pi} \int_{0}^{\pi} T_L(\theta,\varphi) e^{i\mathbf{k}(\theta,\varphi)\cdot\mathbf{d}} \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\varphi, \quad D > d, \tag{2}$$

where $\mathbf{k}$ is the wave-vector

$$\mathbf{k}(\theta,\varphi) = k\big((\widehat{\mathbf{x}}\cos\varphi + \widehat{\mathbf{y}}\sin\varphi)\sin\theta + \widehat{\mathbf{z}}\cos\theta\big) \tag{3}$$

and

$$T_L(\theta,\varphi) = T_L\left(\widehat{\mathbf{k}}\cdot\widehat{\mathbf{D}}\right) = \sum_{n=0}^{L} i^n (2n+1) h_n^{(1)}(kD) P_n\left(\widehat{\mathbf{k}}\cdot\widehat{\mathbf{D}}\right) \tag{4}$$

is the Rokhlin translation function. Here, we use the notation $\widehat{\mathbf{k}} = \mathbf{k}/k$ and $k = |\mathbf{k}|$ for $\mathbf{k}$ and other vectors. The truncation order or degree L controls the error in the representation (2). For d_0 digits of accuracy the usual choice is [2]

$$L \approx kd + 1.8(d_0)^{2/3}(kd)^{1/3}, \quad \text{for } L < kD. \tag{5}$$

At low frequencies, or equivalently with small kD, the error is, however, not controllable. By increasing the number of buffer boxes, i.e., by increasing D, one could improve the accuracy, but at the same time the computational cost is increasing because more outer-to-inner translations need to be performed, and more nearby interactions need to be calculated directly. The error control can be improved somewhat at lower frequencies or levels by choosing a more optimal truncation order L, as described in [11], but ultimately this so called *low-frequency breakdown* limits the minimum division cube side length to about $\lambda/10 \ldots \lambda$ depending on the desired accuracy.

To completely avoid the low-frequency breakdown, we have to use another representation of the Green's function. In this paper we study the following two approaches. The first one is based on the spectral representation of the Green's function, also called the inhomogeneous plane-wave expansion [5–8], while the second one, called the uniform MLFMA (UMLFMA) [10], is based on a modification of the representation (2) of the Green's function.

The spectral representation of the Green's function can be written as

$$G(\mathbf{r}) = \frac{ik}{8\pi^2} \int_{-\pi}^{\pi} \int_{\Gamma} e^{i\mathbf{k}(\theta,\varphi)\cdot\mathbf{r}} \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\varphi, \qquad z = \widehat{\mathbf{z}} \cdot \mathbf{r} > 0 \tag{6}$$

where $\mathbf{k}(\theta, \varphi)$ is defined as before (3), but now the angle θ is integrated over the path $\Gamma = \Gamma_p + \Gamma_e$ in the complex plane shown in Fig. 1. The wave-vector $\mathbf{k}$ is real for the propagating part Γ_p, while $\mathbf{k}$ is imaginary for the evanescent part Γ_e. The propagating part G_p and the evanescent part G_e of the Green's function are obtained from (6) by restricting the integration to Γ_p or Γ_e, respectively.

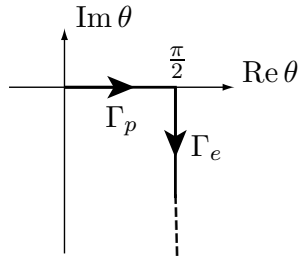

Figure 1. The integration path $\Gamma = \Gamma_p + \Gamma_e$ in the complex plane.

The representation (6) is valid only for $z > 0$, and therefore, we need to split the translation into six directions: $+z$, $-z$, $+x$, $-x$, $+y$ and $-y$, and consider each direction separately. Otherwise, the formula is fairly similar to (2) used in the original MLFMA; instead of the Rokhlin translation function (4) we just have a simpler exponential translation function $e^{i\mathbf{k}\cdot\mathbf{D}}$, but on the other hand, the integration path is now more complicated.

The recently proposed UMLFMA [10] is based on the representation

$$G(\mathbf{D} + \mathbf{d}) \approx \frac{ik}{(4\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi-i\alpha}^{\pi-i\alpha} T(\theta, \varphi) e^{i\mathbf{k}(\theta,\varphi)\cdot\mathbf{d}} \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\varphi, \tag{7}$$

where the parameter α is related to the division cube side length a as

$$\alpha = \max\left(-\ln\left(ka\sqrt{3}\right), 0\right). \tag{8}$$

This representation is direction-independent as far as the related far-field and incoming wave patterns are concerned, but we have

to numerically find a different translation function $T(\theta, \varphi)$ for each different translation-vector $\mathbf{D}$ at each level. Note also that the parameter α in (8) varies from one level to another because a does that.

3. IMPLEMENTATION OF THE TRANSLATION PROCEDURES

All the above representations for the Green's function can be expressed in the general form

$$G(\mathbf{D}+\mathbf{d}) = \int_S T(\mathbf{k}, \mathbf{D})\, e^{i\mathbf{k}\cdot\mathbf{d}}\, \mathrm{d}S(\mathbf{k}). \tag{9}$$

Let $\mathbf{F}$ be a field due to a source distribution $q(\mathbf{r}')$ in a division cube Q_1, and consider $\mathbf{F}(\mathbf{r})$ as an incoming field in a well-separated cube Q_2 of the same division level, see Fig. 2. Using (9), we can express $\mathbf{F}(\mathbf{r})$ as

$$\begin{aligned} F(\mathbf{r}) &= \int_{Q_1} G(\mathbf{D}+\mathbf{r}-\mathbf{r}')\, q(\mathbf{r}')\, \mathrm{d}V' \\ &= \int_S \left(\int_{Q_1} e^{-i\mathbf{k}\cdot\mathbf{r}'} q(\mathbf{r}')\, \mathrm{d}V' \right) T(\mathbf{k}, \mathbf{D})\, e^{i\mathbf{k}\cdot\mathbf{r}}\, \mathrm{d}S(\mathbf{k}), \end{aligned} \tag{10}$$

where $\mathbf{D}$ is the translation vector from the origin of Q_1 to that of Q_2. If we represent the outgoing field from the cube Q_1 by using the *far-field pattern*

$$F_\infty(\mathbf{k}) = \int_{Q_1} e^{-i\mathbf{k}\cdot\mathbf{r}'} q(\mathbf{r}')\, \mathrm{d}V', \tag{11}$$

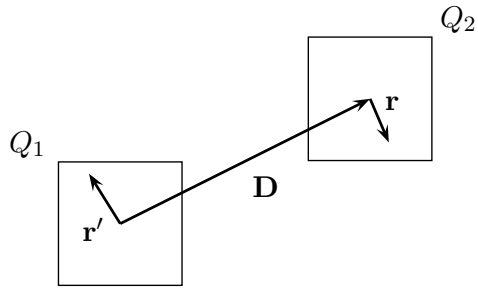

Figure 2. Outer-to-inner translation from cube Q_1 to cube Q_2. The cube Q_1 contains a source $q(\mathbf{r}')$ and we want to compute the incoming field $F(\mathbf{r})$ inside the cube Q_2. The cubes are assumed to be separated by at least one empty cube in each direction.

we get the local incoming field inside Q_2 as a *plane-wave expansion*

$$F(\mathbf{r}) = \int_S V(\mathbf{k})\, e^{i\mathbf{k}\cdot\mathbf{r}}\, \mathrm{d}S(\mathbf{k}), \tag{12}$$

where

$$V(\mathbf{k}) = F_\infty(\mathbf{k})\, T(\mathbf{k}, \mathbf{D}) \tag{13}$$

is the *incoming wave pattern*. The *translation function* $T(\mathbf{k}, \mathbf{D})$ is diagonal in the sense that it only depends on the translation vector $\mathbf{D}$ but not on $\mathbf{r}$ or $\mathbf{r}'$.

The most straightforward way to implement the translation procedures in the MLFMA is to choose an appropriate quadrature rule for the field integral (10) and use the sample points $\mathbf{k}_n$ of the quadrature when representing the far-field patterns and incoming wave patterns in terms of sample matrices. A shift of origin is particularly simple for this representation; we just multiply the far-field or incoming wave pattern samples point-wise with $e^{i\mathbf{k}_n\cdot\mathbf{p}}$, where $\mathbf{p}$ is a vector from the old origin to the new one.

In the MLFMA the grid of sample points $\mathbf{k}_n$ of the far-field and incoming wave patterns must usually vary from one level to another in order to maintain the wanted representation accuracy using a minimal number of sample points. Therefore, in the aggregation steps the far-field sample matrices must be interpolated from a lower level sampling grid to a higher level one, and similarly, in the disaggregation steps, the incoming wave pattern sample matrices must be anterpolated from a higher level to a lower level.

In the following subsections we in detail present the various translation procedures. We first consider the translations of the propagating and evanescent parts of the fields in the MLFMA based on the spectral representation (6), and thereafter, the translations in the UMLFMA. Finally, we summarize the different combinations that can be used to implement a broadband MLFMA.

3.1. Propagating Part of the Field

The propagating part of the spectral representation (6) of the Green's function

$$G_p(\mathbf{D}+\mathbf{d}) = \frac{ik}{8\pi^2} \int_{-\pi}^{\pi} \int_{0}^{\pi/2} e^{i\mathbf{k}(\theta,\varphi)\cdot(\mathbf{D}+\mathbf{d})} \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\varphi, \tag{14}$$

is almost of the same form as Rokhlin's translation formula (2). The translations of the propagating part of the field can therefore be

handled almost in the same way as in the original MLFMA. The main difference is that the integration in (14) is over only one half of the unit sphere. However, the interpolations and anterpolations between levels are much more efficient if we sample the far-field and incoming wave patterns on the entire sphere. Furthermore, for all six directions we anyway need the samples on the entire sphere.

The fact that the integral is over only a half-sphere can be treated in the translations from one cube to another in the following two ways. We can sample the translation function using a suitable quadrature rule over the half-sphere and then interpolate to the sample points over the entire unit sphere. This approach is used by Jiang and Chew [7]. Another approach is to multiply the translation function by a a suitably smoothed characteristic function, as outlined by Darve and Havé [8], after which the integrals again can be taken over the entire sphere. Here, we will use the second approach, based on 2-D trigonometric polynomials and Fourier-interpolation, essentially in the same way as in the FFT-based MLFMA of Sarvas [12].

First, we rewrite the integral in (14) over the entire sphere and get

$$G_p(\mathbf{D}+\mathbf{d}) = \frac{ik}{8\pi^2} \int_{-\pi}^{\pi} \int_{0}^{\pi} e^{i\mathbf{k}(\theta,\varphi)\cdot(\mathbf{D}+\mathbf{d})} \chi_{\left[0,\frac{\pi}{2}\right]}(\theta) \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\varphi, \tag{15}$$

where $\chi_{[a,b]}$ is the characteristic function of an interval $[a,b]$ defined by

$$\chi_{[a,b]}(x) = \begin{cases} 1, & a \le x \le b \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Then, to be able to use the FFT method, as described in detail in [12], we extend all functions $U(\theta,\varphi)$, defined on the unit sphere and 2π-periodic in φ, in the θ-variable from $0 \le \theta \le \pi$ to $-\pi \le \theta \le \pi$ in the natural way by setting

$$U(-\theta,\varphi) = U(\theta,\varphi+\pi), \quad \text{for } 0 \le \theta \le \pi, \tag{17}$$

which makes them 2π-periodic also in θ. After this extension we can write the integral (15) as

$$G_p(\mathbf{D}+\mathbf{d}) = \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} T_p(\theta,\varphi) \, e^{i\mathbf{k}(\theta,\varphi)\cdot\mathbf{d}} \, \mathrm{d}\theta \, \mathrm{d}\varphi, \tag{18}$$

where T_p is the translation function in the $+z$-direction

$$T_p(\theta,\varphi) = \frac{ik}{(4\pi)^2}\, e^{i\mathbf{k}(\theta,\varphi)\cdot\mathbf{D}}\, S(\theta), \quad \text{where} \tag{19}$$

$$S(\theta) = \chi_{\left[-\frac{\pi}{2},\frac{\pi}{2}\right]}(\theta)\,|\sin\theta|\,. \tag{20}$$

For translations in other directions, we also need characteristic functions in φ so that the integration is over the appropriate half-sphere. For instance, for the translation in the $+y$-direction, $\chi_{\left[0,\frac{\pi}{2}\right]}(\theta)$ in the integral (15) must be replaced by $\chi_{[0,\pi]}(\varphi)$, and after the extension the translation function in the $+y$-direction is of the form (19) where $S(\theta)$ is replaced by

$$S(\theta,\varphi) = \left[\chi_{[0,\pi]}(\theta)\,\chi_{[0,\pi]}(\varphi) - \chi_{[-\pi,0]}(\theta)\,\chi_{[-\pi,0]}(\varphi)\right]\sin\theta. \tag{21}$$

Now, the representation (18) is of the form (9) and we can efficiently compute the integrals using a 2D-trapezoidal rule if we use the orthogonality properties of the trigonometric polynomials, as we describe next, instead of directly sampling the translation function (19). Except for the outer-to-inner translation, which is not a simple point-wise multiplication, the implementation follows the general outline above in Section 3. In particular, we represent the outgoing field from a cube Q due to the source $q(\mathbf{r}')$ using the usual far-field pattern

$$F_\infty(\theta,\varphi) = \int_Q e^{-i\mathbf{k}(\theta,\varphi)\cdot\mathbf{r}'} q(\mathbf{r}')\,\mathrm{d}V', \tag{22}$$

which is a smooth $2\pi\times 2\pi$-periodic function after the extension (17), and we represent the propagating part F_p of the incoming or local field using the incoming wave pattern V of the plane-wave expansion

$$F_p(\mathbf{r}) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} V(\theta,\varphi)\, e^{i\mathbf{k}(\theta,\varphi)\cdot\mathbf{r}}\,\mathrm{d}\theta\,\mathrm{d}\varphi. \tag{23}$$

Smooth $2\pi\times 2\pi$-periodic functions can be efficiently approximated by truncated Fourier series, or *trigonometric polynomials*, and in particular, for the far-field pattern F_∞ we get,

$$F_\infty(\theta,\varphi) \approx \sum_{m=-M}^{M}\sum_{n=-N}^{N-1} a_{m,n} e^{i(m\theta+n\varphi)}. \tag{24}$$

In storing F_∞, instead of the above coefficients $a_{m,n}$, we store the samples

$$F^\infty_{m,n} = F_\infty \left(\frac{2\pi}{2M+1} m, \frac{\pi}{N} n \right), \tag{25}$$

where only the $(M+1) \times 2N$ samples with $0 \leq m \leq M$ and $-N \leq n < N$ need to be stored. For notational convenience, we assume that $M = N$ and call N the degree of the trigonometric polynomial (24). The sampling rate is also N.

The far-field pattern F_∞ has an effective spatial bandwidth of approximately $kd/2$, e.g., see [13], where $d = a\sqrt{3}$ is the diameter of the cube. Therefore, the trigonometric polynomial representation can be made arbitrarily accurate by choosing $N > kd/2$. The choice

$$N = \frac{kd}{2} + 1.8(d_0)^{2/3} \left(\frac{kd}{2} \right)^{1/3}, \tag{26}$$

which is approximately *half of the sampling rate* of the original MLFMA, seems to be sufficient for d_0 digits of accuracy.

To perform the outer-to-inner translation, we make use of the orthogonality of the trigonometric polynomials so that we can evaluate the field integral (23) accurately using the trapezoidal rule.

First, we observe that the function $e^{i\mathbf{k}\cdot\mathbf{r}}$ in (23) is of the same degree N as the far-field pattern F_∞. Therefore, due to the orthogonality of the terms $e^{i(m\theta+n\varphi)}$ over the integration domain $-\pi \leq \theta, \varphi \leq \pi$, it is sufficient to anterpolate the incoming wave pattern $V = F_\infty T_p$ to the same degree N. This anterpolated $\widetilde{V}$ is the wanted end-product of the outer-to-inner translation. To achieve this, as shown in [12], we need to interpolate the far-field pattern F^∞ to degree $2N$ and smooth, or anterpolate, the translation function T_p to degree $2N$. Thereafter, we multiply the samples point-wise and anterpolate the product to degree N.

To form the smoothed translation function $\widetilde{T}_p$, we note that a sufficient degree for representing the function $e^{i\mathbf{k}\cdot\mathbf{D}}$ is

$$N' = kD + 1.8(d_0)^{2/3} (kD)^{1/3}, \tag{27}$$

while the degree of the function $S(\theta)$ or $S(\theta, \varphi)$ is infinite. Therefore, we sample the function $e^{i\mathbf{k}\cdot\mathbf{D}}$ at degree $2N + N'$ and anterpolate the function S to degree $2N + N'$ by truncating its Fourier series representation. After multiplying the samples point-wise we anterpolate the product to degree $2N$ to obtain the samples of $\widetilde{T}_p$.

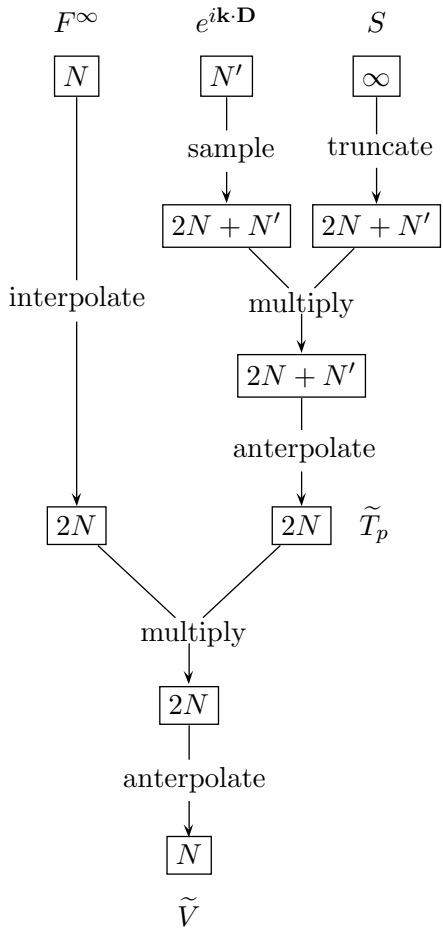

Figure 3. Outer-to-inner translation procedure to form the anterpolated incoming wave pattern $\widetilde{V}$. The values in the boxes are the degrees or sample rates of the relevant quantities. The scalar factor $ik(4\pi)^{-2}$ is omitted for brevity.

For a flowchart for the translation procedure, see Fig. 3. The needed intermediate degree $2N + N'$ can be quite large, and thus it may be preferable to precompute $\widetilde{T}_p$ to speed up the outer-to-inner translations.

In forming the smoothed $\widetilde{T}_p$ for different translation directions, as

(20) and (21) indicate, we need the Fourier series

$$S(\theta) = \chi_{\left[-\frac{\pi}{2},\frac{\pi}{2}\right]}(\theta)\,|\sin\theta| = \sum_{n=-\infty}^{\infty} a_n e^{in\theta}, \tag{28}$$

$$u(\theta) = \chi_{[0,\pi]}(\theta)\sin\theta = \sum_{n=-\infty}^{\infty} b_n e^{in\theta} \quad \text{and} \tag{29}$$

$$v(\varphi) = \chi_{[0,\pi]}(\varphi) = \sum_{n=-\infty}^{\infty} c_n e^{in\varphi}, \tag{30}$$

where the coefficients are

$$a_{-1} = a_1 = \frac{1}{2\pi}, \qquad a_n = \frac{2 + n\,i^{n+1}\left(1-(-1)^n\right)}{2\pi\left(1-n^2\right)}, \tag{31}$$

$$b_{-1} = \frac{i}{4},\ b_1 = \frac{-i}{4}, \quad b_n = \frac{1+(-1)^n}{2\pi\left(1-n^2\right)} \quad \text{and} \tag{32}$$

$$c_0 = \frac{1}{2}, \qquad c_n = \frac{i\left((-1)^n-1\right)}{2\pi n}. \tag{33}$$

Other needed Fourier series are those of the functions $S(\theta+\pi)$, $u(-\theta)$, $v(-\varphi)$ and $v(\varphi\pm\pi/2)$, which are easily obtained from the above series.

The outer-to-outer and inner-to-inner translations can be performed in the same way as in the original MLFMA, using shifts and interpolations or anterpolations as appropriate. For the propagating part, we perform all interpolations and anterpolations using FFT as described in [12].

Finally, we observe that at higher levels, i.e., when the division cube side length is larger than about one wavelength, we can represent the whole field using the above defined far-field and incoming wave patterns. Then, we can simply omit the evanescent representations and replace the propagating translation function (19) with the function

$$T(\theta,\varphi) = \frac{ik}{32\pi^2}\,T_L(\theta,\varphi)\,|\sin\theta|\,, \tag{34}$$

where T_L is the Rokhlin translation function (4) of the appropriate degree $L \approx 2N$. This translation function for the whole field can then be handled in the same way as the propagating translation function above, or slightly differently as described in detail in [12].

3.2. Evanescent Part of the Field

We make the substitution $\sigma = -ik\cos\theta$ in the integral (6) and get the evanescent part of the Green's function as the integral

$$G_e(\mathbf{D}+\mathbf{d}) = \int_{-\pi}^{\pi}\int_{0}^{\infty} T_e(\sigma,\varphi)\, e^{i\mathbf{k}(\sigma,\varphi)\cdot\mathbf{d}}\,\mathrm{d}\sigma\,\mathrm{d}\varphi, \tag{35}$$

where T_e is the evanescent translation function

$$T_e(\sigma,\varphi) = \frac{1}{8\pi^2} e^{i\mathbf{k}(\sigma,\varphi)\cdot\mathbf{D}} \tag{36}$$

and $\mathbf{k}(\sigma,\varphi)$ is the complex wave-vector

$$\mathbf{k}(\sigma,\varphi) = \sqrt{\sigma^2+k^2}\,(\hat{\mathbf{x}}\cos\varphi + \hat{\mathbf{y}}\sin\varphi) + i\sigma\hat{\mathbf{z}}. \tag{37}$$

Furthermore, we assume that $\hat{\mathbf{z}}\cdot(\mathbf{D}+\mathbf{d}) > 0$, i.e., we only consider translations in the $+z$-direction. Other directions follow in a similar fashion by simply rotating the coordinate system: e.g., for translations in the $+x$-direction, the wave-vector is

$$\mathbf{k}^{+x}(\sigma,\varphi) = \sqrt{\sigma^2+k^2}\,(-\hat{\mathbf{z}}\cos\varphi + \hat{\mathbf{y}}\sin\varphi) + i\sigma\hat{\mathbf{x}}. \tag{38}$$

This representation (35) is exactly of the form (9), and so we immediately get the formulas needed for the outer-to-inner translation shown in Fig. 2. The evanescent part of the outgoing field from the cube Q_1 due to the source $q(\mathbf{r}')$ is represented using the *evanescent far-field pattern*

$$F_\infty^e(\sigma,\varphi) = \int_{Q_1} e^{-i\mathbf{k}(\sigma,\varphi)\cdot\mathbf{r}'} q(\mathbf{r}')\,\mathrm{d}V', \tag{39}$$

while the evanescent part F_e of the local incoming field inside the cube Q_2 due to the source inside Q_1 is expressed as an *inhomogeneous plane-wave expansion*

$$F_e(\mathbf{r}) = \int_{-\pi}^{\pi}\int_{o}^{\infty} V_e(\sigma,\varphi)\, e^{i\mathbf{k}(\sigma,\varphi)\cdot\mathbf{r}}\,\mathrm{d}\sigma\,\mathrm{d}\varphi, \tag{40}$$

where

$$V_e(\sigma,\varphi) = F_\infty^e(\sigma,\varphi)\, T_e(\sigma,\varphi) \tag{41}$$

is the *evanescent incoming wave pattern* and T_e is the translation function (36).

To implement the MLFMA we need an efficient quadrature rule for the integral (40), or equivalently for the integral (35), and also interpolation and anterpolation schemes for the far-field patterns and incoming wave patterns between the different levels.

3.2.1. Quadrature Rules and Outer-to-Inner Translation

The integrand in (35) is a smooth 2π-periodic function in φ and so we can use a trapezoidal rule for the φ integral, and the interpolation and anterpolation in φ can be done using FFT as for the propagating part. We use $2N$ quadrature points

$$\varphi_n = \frac{\pi}{N} n, \quad -N \le n \le N-1, \tag{42}$$

and the weights are π/N. An optimal degree N can be found numerically once the σ quadrature points are known.

The difficult part is the σ integration, for which we find a generalized Gaussian quadrature by using the algorithm of Yarvin and Rokhlin [9]. We start by transforming (35) using the integral representation

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \cos\alpha)\, \mathrm{d}\alpha = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{ix\cos\alpha}\, \mathrm{d}\alpha, \tag{43}$$

for the Bessel function of the first kind and the substitution $s = \sigma a$, where a is the division cube side length, and get

$$G_e(\mathbf{D}+\mathbf{d}) = \frac{1}{4\pi a} \int_0^\infty e^{-\frac{z}{a}s} J_0\left(\frac{\rho}{a}\sqrt{s^2+(ka)^2}\right) \mathrm{d}s, \tag{44}$$

where $\rho = \sqrt{x^2+y^2}$ with $(x,y,z) = \mathbf{D}+\mathbf{d}$.

This is the integral for which we want to find an efficient quadrature. For $ka \to 0$, this reduces to the same integral as in [9, section 6.3], and so we could use the quadrature for $k = 0$ available at `http://www.netlib.org/pdes/multipole/vwts.f`, which is rather good also for small ka. However, for $ka \approx 1$ or larger it is more efficient to find new quadrature rules.

The algorithm for finding the quadrature points and weights is described in detail in the article by Yarvin and Rokhlin [9], and thus we here only provide some comments on the implementation. First we

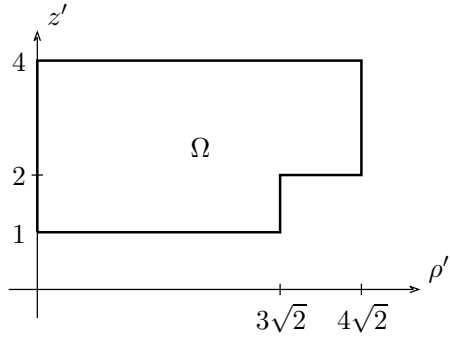

Figure 4. The region Ω.

observe that due to (43) the kernel whose singular functions are to be computed can be taken to be

$$K(s;\rho',z') = e^{-z's}\cos\left(\rho'\sqrt{s^2+(ka)^2}\right), \tag{45}$$

with (ρ', z') varying over the boundary $\partial\Omega$ of the region Ω in Fig. 4; this domain covers all the points $(x, y, z) = \mathbf{D} + \mathbf{d}$ where $\mathbf{d} = \mathbf{r} - \mathbf{r}'$ and $\mathbf{r} \in Q_1$, $\mathbf{r}' \in Q_2$, while Q_1, Q_2 run through all pairs in the interaction list in any of the six directions. (Note, however, that this requires us to split the interaction list in six approximately equal length lists, somewhat differently than in e.g. [6].) For the approximate SVD of (45) we use a high order Gauss-Legendre quadrature for $s \in [0, s_{max}]$, $s_{max} \approx 36$, while we discretize $\partial\Omega$ piecewise using six high order Gauss-Legendre quadratures.

After deriving the quadrature points and weights of different orders for different levels (i.e., different ka), we test the accuracy against the actual integral (44) for $(\rho/a, z/a)$ in the whole region Ω to verify that the quadrature rules work as expected. Somewhat surprisingly, no problems due to the possibility of interior Helmholtz resonances of Ω were met by the authors when testing the algorithm for $ka = 2^l\pi$ with $l = -10, \ldots, 10$.

In this way, quadrature rules can be obtained for any ka and accuracies up to at least 10 significant digits if needed. For practical purposes, however, it is more convenient to tabulate the points and weights once and for all. Numerical tests suggest that a quadrature rule derived for some ka actually works at the same accuracy for any larger ka. On the other hand, for increasing ka, we need fewer quadrature points to obtain the same accuracy if we derive new quadrature rules. A limited number of precomputed quadrature rules is thus sufficient to cover all frequencies and levels for a specific accuracy.

Table 1. Quadrature points s_m and weights w_m for the integral (44) for $ka = \pi$ and four digits accuracy.

m	s_m	w_m
1	0.1177386084	0.3030688291
2	0.6238529768	0.7048731895
3	1.4869765464	0.9868793874
4	2.5383212196	1.0993614501
5	3.6732075130	1.1691224893
6	4.8755665557	1.2349243363
7	6.1416349409	1.2967438942
8	7.4677829649	1.3567820556
9	8.8522946903	1.4218293927

The points and weights for one representative quadrature rule is shown in Table 1, while the points and weights for other quadrature rules covering all frequencies or levels and at least two or four digits accuracy are available at `http://www.hut.fi/~kwallen/equad/`. The needed number of sample points are also found in Table 2 in Section 4.

Finally, the M quadrature points and weights for the σ-integral in (35) are

$$\sigma_m = \frac{s_m}{a}, \quad w_m = \frac{v_m}{a}, \quad 1 \leq m \leq M, \tag{46}$$

where (s_m, v_m) are the quadrature points and weights for the integral (44) with respect to s.

We now have efficient quadratures for the field integral (40) both in φ and σ, which also yield efficient sampling points for the far-field patterns and incoming wave patterns. Accordingly, for the integral (40) we get

$$F_e(\mathbf{r}) \simeq \sum_{m=1}^{M} \sum_{n=-N}^{N-1} \frac{\pi}{N} w_m \, V_e(\sigma_m, \varphi_n) \, e^{i\mathbf{k}(\sigma_m, \varphi_n)\cdot\mathbf{r}} \tag{47}$$

where the incoming wave pattern V_e is stored as a $M \times 2N$ sample matrix, with elements $(V_e)_{m,n} = V_e(\sigma_m, \varphi_n)$. Similarly, the far-field pattern is stored as a $M \times 2N$ sample matrix F^e_∞, and the outer-to-inner translation is computed as a point-wise multiplication

$$(V_e)_{m,n} = (F^e_\infty)_{m,n} \, (T_e)_{m,n}, \tag{48}$$

where $(T_e)_{m,n} = T_e(\sigma_m, \varphi_n)$ is the discretized translation function.

We finish this section by noting that the field integral (40) is convergent in σ, since $\widehat{\mathbf{z}}\cdot(\mathbf{D}+\mathbf{r}-\mathbf{r}') > 0$, but the far-field $F^e_\infty(\sigma,\varphi)$ and the function $e^{i\mathbf{k}(\sigma,\varphi)\cdot\mathbf{r}}$ are not bounded as $\sigma \to \infty$. This is, however, not a problem in practise as the σ sample points are small enough so that the needed samples do not become too large. We could add and subtract a suitably defined shift-vector, as in [8], to make the functions bounded, but this is unnecessary in a sample point based implementation.

3.2.2. Interpolation and Anterpolation

As we represent the evanescent far-field patterns and incoming wave patterns using their sample points, the needed shifts of origins are simple: essentially a point-wise multiplication with an exponential function. The needed interpolation and anterpolation operations are, however, less obvious. The problematic part is the σ-direction. The number of sample points does not change rapidly between levels, but due to the change in scaling, the sample points are approximately scaled by a factor of two between levels. At lower levels, the normalized sample points $s_m = a\sigma_m$ are nearly the same for adjacent levels, but the division cube side length $a = a_l$ is changing by a factor of two from level to level.

Since we have derived the sample points for the field integral (40), the far-field patterns and incoming wave patterns become oversampled, and it turns out that the following straightforward interpolation scheme works sufficiently well.

The interpolation from level l to level $l+1$ in the φ-direction can be handled using FFT as for the propagating part, and we may assume that the number of sample points in the φ-direction is the same $2N$ at both level l and level $l+1$. Otherwise, we can first interpolate in the φ-direction and make them the same.

For the interpolation in the σ-direction let F^l and F^{l+1} be the level l and $l+1$ evanescent far-field sample matrices representing the same far-field, due to a source in a level l division cube Q, and with matrix elements

$$F^l_{mn} = F^e_\infty\left(\sigma^l_m, \varphi_n\right), \quad m = 1, \ldots, M^l,\ n = -N, \ldots, N-1. \tag{49}$$

We want to find an $M^{l+1} \times M^l$ interpolation matrix E^l so that

$$F^{l+1} = E^l F^l. \tag{50}$$

It suffices to find E^l which is valid for all point sources at points $\mathbf{r}$ in

Q. The far-field due to a point source at $\mathbf{r} = (x, y, z)$ is of the form

$$F = \exp(-i\sqrt{\sigma^2 + k^2}(x\cos\varphi + y\sin\varphi) + \sigma\, z), \tag{51}$$

and we easily see that if E^l satisfies (50) for all point sources in Q, then it is independent of φ. Therefore, we can integrate away the φ-dependence from (50) using (43), and enforce the resulting equation at a sufficient number of test-points (x_n, y_n, z_n). We get a matrix-equation

$$\widetilde{F}^{l+1} = E^l \widetilde{F}^l, \tag{52}$$

$$\widetilde{F}^j_{mn} = e^{z_n \sigma^j_m} J_0\left(\rho_n \sqrt{\left(\sigma^j_m\right)^2 + k^2}\right), \quad j = l,\, l+1, \tag{53}$$

where $\rho_n = \sqrt{x_n^2 + y_n^2}$. The test points are chosen inside the level l division cube Q, of side length a_l, i.e.

$$0 \le \rho_n \le \frac{a_l}{\sqrt{2}}, \quad -\frac{a_l}{2} \le z_n \le \frac{a_l}{2}. \tag{54}$$

Using enough test points, we get an over-determined linear system with full rank, which we solve for E^l in the least squares sense.

Numerical tests show that the accuracy of this interpolation scheme is more than sufficient for levels with $a_l \le 2\lambda$, i.e., that the error in the σ-interpolation is much smaller that the error in the corresponding quadrature rule. At higher levels the accuracy degrades somewhat due to the small number of sample points M_l. Also, when increasing the number of sample points in σ, the accuracy of the quadrature rule increases faster than the accuracy of the interpolation scheme. However, for up to at least four digits precision the interpolation scheme works well.

The σ-anterpolation can be carried out using the transpose of the interpolation matrix as in the original MLFMA. The anterpolation from level $l+1$ to level l can be done using a $M^l \times M^{l+1}$ -matrix H^l defined by

$$H^l = \left(W^l\right)^{-1} \left(E^l\right)^T W^{l+1}, \tag{55}$$

where the diagonal matrices W^l and W^{l+1} contains the σ-quadrature weights for the respective level.

We need to compute and store only one interpolation-matrix E^l for each pair of adjacent levels, so the setup time and memory consumption is negligible compared to the rest of the computations. Furthermore,

since the interpolation and anterpolation matrices are small, these operations are efficient. For instance, for 4 digits of accuracy, $M_l \leq 11$.

Finally, we note that, when storing the far-field patterns and incoming wave patterns, the oversampling in the φ-direction is unnecessary. We can significantly reduce the memory-consumption by using a smaller sampling rate $N' < N$ in the φ-direction when storing the far-field and incoming wave patterns, and only when translating a far-field pattern to an incoming wave pattern increase the sampling rate to N by interpolating in the φ-direction, and finally anterpolate back to the rate N' after the translation. Actually, it appears that we could also perform the outer-to-inner translations using the smaller number of samples, $M \times 2N'$, if we numerically compute a suitable translation function $\widetilde{T}_e$ in a similar fashion as in the UMLFMA.

3.3. UMLFMA

When the division cubes are small compared with the wavelength, the evanescent part of the spectral representation (6) is dominant. Based on this fact Xuan et al. [10] proposed the UMLFMA representation where the θ-integral is shifted a distance α away from the real axis at lower levels. However, as $k \to 0$, the parameter $\alpha \to \infty$, and the representation appears not to reach the static case $k = 0$ as a limit.

We here present a modification of the UMLFMA, which more explicitly contains the static case. We start by embedding the term $ik(4\pi)^{-2}\sin\theta$ into the translation function $T(\theta, \phi)$ in (7). Using the substitution $\theta = t - i\alpha$ we get an equivalent representation

$$G(\mathbf{D} + \mathbf{d}) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} T(t, \varphi)\, e^{i\mathbf{k}(t,\varphi)\cdot\mathbf{r}}\, \mathrm{d}t\, \mathrm{d}\varphi, \tag{56}$$

where

$$\mathbf{k}(t, \varphi) = (\widehat{\mathbf{x}}\cos\varphi + \widehat{\mathbf{y}}\sin\varphi)\, k\sin(t - i\alpha) + \widehat{\mathbf{z}}\, k\cos(t - i\alpha). \tag{57}$$

As a further modification, we express α using a new parameter γ as

$$\alpha = \max\bigl(-\ln(ka/\gamma), 0\bigr), \tag{58}$$

where a is the division cube side length. The choice $\gamma = 1/\sqrt{3}$ gives (8), which is the value of α used in [10]. However, it seems that a larger γ is needed to improve the accuracy, as we will see below.

For $ka < \gamma$, the choice of $\alpha = -\ln(ka/\gamma)$ gives

$$k\sin(t - i\alpha) = \frac{\gamma}{2ia}\left(e^{it} - (ka/\gamma)^2 e^{-it}\right) \quad \text{and} \tag{59}$$

$$k\cos(t - i\alpha) = \frac{\gamma}{2a}\left(e^{it} + (ka/\gamma)^2 e^{-it}\right). \tag{60}$$

Substituting these equations into (57) we get

$$\begin{aligned}\mathbf{k}(t,\varphi) = \frac{\gamma}{2ia}\Big(&\left(e^{it} - (ka/\gamma)^2 e^{-it}\right)\left(\widehat{\mathbf{x}}\cos\varphi + \widehat{\mathbf{y}}\sin\varphi\right) \\ &+ i\left(e^{it} + (ka/\gamma)^2 e^{-it}\right)\widehat{\mathbf{z}}\Big).\end{aligned} \tag{61}$$

We see that with this form of the wave-vector $\mathbf{k}(t,\varphi)$ the representation (56) is valid even in the static case $k = 0$.

On the other hand, for $ka \geq \gamma$, we get $\alpha = 0$ or equivalently $t = \theta$, and the representation (56) takes exactly the same form as the propagating part of the spectral representation (18). Then, we can use (34) as the translation function and perform the translations as described above for the propagating part of the field. In the following we, therefore, mainly consider the case $ka < \gamma$.

Again, the representation (56) is of the form (9), and thus the implementation follows the same general outline as described above in Section 3. In particular, for the outer-to-inner translation in Fig. 2, we represent the outgoing field from the cube Q_1 due to the source q using the far-field pattern

$$F_\infty(t,\varphi) = \int_{Q_1} e^{-i\mathbf{k}(t,\varphi)\cdot\mathbf{r}'} q(\mathbf{r}')\,\mathrm{d}V', \tag{62}$$

while the local incoming field is expressed as a plane-wave expansion

$$F(\mathbf{r}) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} V(t,\varphi)\, e^{i\mathbf{k}(t,\varphi)\cdot\mathbf{r}}\,\mathrm{d}t\,\mathrm{d}\varphi \tag{63}$$

using the incoming wave pattern V. For discretizing the integral (63), we use a 2D-trapezoidal rule with $N = N_t \times N_\varphi$ points in the t and φ directions. The outer-to-inner translations we perform using a pointwise multiplication $V(t,\varphi) = T(t,\varphi)F_\infty(t,\varphi)$.

However, the UMLFMA has two characteristic new features compared with the original MLFMA. First of all, the translation function $T(t,\varphi)$ is unknown and must be solved numerically from an ill-posed integral equation. The second new feature is that the wave-vector $\mathbf{k}$ is different for each level, except for $ka \geq \gamma$, and we need to extrapolate the far-field and incoming wave patterns from one level to another.

3.3.1. Integral Equation for the Translation Function

We assume that $\mathbf{D}$ is fixed, i.e. we compute a separate translation function $T(t,\varphi)$ for each of the $7^3 - 3^3 = 316$ different translation vectors $\mathbf{D}$ at each level. Due to (56) the translation function $T(t,\varphi)$ is a solution of the integral equation

$$\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} K(\mathbf{d},t,\varphi)\,T(t,\varphi)\,\mathrm{d}t\,\mathrm{d}\varphi = G(\mathbf{D}+\mathbf{d}) \quad \text{for all } \mathbf{d}\in\tilde{Q}, \tag{64}$$

where

$$K(\mathbf{d},t,\varphi) = e^{i\mathbf{k}(t,\varphi)\cdot\mathbf{d}} \tag{65}$$

is the kernel, G is the Green's function (1) and $\widetilde{Q}$ is a cube with side length $2a$ and center at the origin. Now, the trouble is that (64) is an integral equation of the first kind with a compact operator, and so it is an *ill-posed problem*, see e.g. [14]. Therefore, we cannot expect to be able to solve the translation function $T(t,\varphi)$ with arbitrary accuracy.

To solve the integral Equation (64), we begin by discretizing the integrals using a two-dimensional trapezoidal rule with N sample points (t_n,φ_n) with $n = 1,\ldots,N$, giving

$$\frac{(2\pi)^2}{N}\sum_{n=1}^{N} K(\mathbf{d},t_n,\varphi_n)\,T(t_n,\varphi_n) \approx G(\mathbf{D}+\mathbf{d}). \tag{66}$$

Next, we enforce the equation at $M > N$ points $\mathbf{d}_m \in \widetilde{Q}$, to get an over-determined matrix equation $AT = F$, where

$$A_{mn} = \frac{(2\pi)^2}{N}K(\mathbf{d}_m,t_n,\varphi_n), \tag{67}$$

$$T_n = T(t_n,\varphi_n) \quad \text{and} \tag{68}$$

$$F_m = G(\mathbf{D}+\mathbf{d}_m). \tag{69}$$

Finally we solve the equation in the least squares sense using the singular value decomposition $A = USV^H$ of A and get

$$T \approx VS^{-1}U^H F, \tag{70}$$

where we regularize the solution T by a singular-value truncation. More specifically, we retain the singular values σ_n for which $\sigma_n > \tau\sigma_1$, where σ_1 is the largest singular value of A and τ is the chosen regularization parameter.

The above solution scheme is straightforward and the computational cost is negligible, since the matrix A does not depend on the vector $\mathbf{D}$ and it thus suffices to compute one singular value decomposition per level. However, the question of choosing the appropriate sampling rate N and also the parameters γ and τ remains to be answered.

Studying (61), we see that the parameter γ adjusts the oscillation of the kernel K. By increasing γ, we increase the oscillation of K and thereby need to increase the sampling rate N, and at the same time we get more degrees of freedom to find a more accurate translation function T. On the other hand, we would like to keep the sampling rate as low as possible. We also note that the interpolations and anterpolations between levels become less accurate if γ is large. Finally, we need to choose τ large enough so that $|T|$ is not too large.

Numerical tests suggest that $N = 9 \times 15$, $\gamma = 3$ and $\tau = 10^{-5}$ are suitable for $ka < \gamma$ and two digits accuracy in the outer-to-inner translations, if we do not take into account the possible errors induced by the interpolations and anterpolations in the outer-to-outer and inner-to-inner translations. By increasing N and γ while decreasing τ, the accuracy in (64) will be improved, but eventually the errors in the interpolations and anterpolations become too large.

3.3.2. Interpolation and Anterpolation

For higher levels, i.e. for $ka \geq \gamma$, the parameter α is zero and all translations can be handled in the same way as for the propagating part of the Green's function. Furthermore, the possible interpolations and anterpolations in the φ-direction can be performed efficiently using FFT at any level. Thus, we need to derive an interpolation scheme from level l to level $l+1$ in the t-direction, and we may assume that the number of samples in the φ-direction is the same N_φ for both levels, and further that $\alpha_l > \alpha_{l+1} \geq 0$.

To derive the interpolation matrix E^l from level l to level $l+1$, we proceed as in Section 3.2.2 for the evanescent σ-interpolation. Let F^l and F^{l+1} be the level l and $l+1$ far-field sample matrices representing the same far-field, due to a source in a level l division cube Q with side length a, and with matrix elements

$$F^l_{mn} = F^l_\infty(t_m, \varphi_n), \quad m = 1, \ldots, N^l_t, \quad n = 1, \ldots, N_\varphi. \tag{71}$$

We want to find an $N^{l+1}_t \times N^l_t$ interpolation matrix E^l so that $F^{l+1} = E^l F^l$. Using the same arguments as in Section 3.2.2, we get a

matrix-equation $\widetilde{F}^{l+1} = E^l \widetilde{F}^l$, with

$$\widetilde{F}^j = e^{-iz_n k \cos(t_m - i\alpha_j)} J_0\left(\rho_n k \sin(t_m - i\alpha_j)\right), \quad j = l,\, l+1, \tag{72}$$

where $k\sin(t - i\alpha)$ and $k\cos(t - i\alpha)$ should preferably be evaluated using (59) and (60) if $ka \ll \gamma$. Again, we choose the test points inside the level l division cube Q with side length a, i.e.

$$0 \le \rho_n \le \frac{a}{\sqrt{2}}, \quad -\frac{a}{2} \le z_n \le \frac{a}{2}. \tag{73}$$

Using enough test points, we get an over-determined linear system with full rank, which we solve for E^l in the least squares sense. Numerical tests suggest that this interpolation scheme works with good accuracy if γ is small and ka is small enough compared with γ, say $\gamma \lesssim 1$ and $ka \lesssim \gamma/10$. Increasing γ makes the interpolations less accurate, and we also observe that for ka close to γ the accuracy is worse than for small ka.

To better understand why the interpolations and anterpolations are difficult, it is illuminating also to consider another interpolation scheme. Looking at the original UMLFMA representation (7), we see that the interpolations and anterpolations in t or θ are essentially extrapolations between different horizontal contours in the complex plane. Thus, after some straightforward manipulations, we see that the interpolation of a far-field pattern from level l to level $l+1$ can be expressed as an analytic continuation of the form

$$F_\infty^{l+1}(t, \varphi) = F_\infty^l(t + i\beta, \varphi), \quad \beta = \begin{cases} \ln 2, & 2ka \le \gamma \\ -\ln(ka/\gamma), & ka < \gamma < 2ka. \end{cases} \tag{74}$$

Considering the far-field pattern as a trigonometric polynomial in t, we can easily implement this analytic continuation efficiently using FFT. However, it turns out that this interpolation scheme is significantly less accurate than the scheme described above.

The anterpolations needed in the inner-to-inner translations can be done using the transpose $(E^l)^T$ of the matrix E^l, as for the evanescent part of the spectral representation. More specifically, the anterpolation matrix H^l from level $l+1$ to level l is

$$H^l = \frac{N_t^l}{N_t^{l+1}} \left(E^l\right)^T. \tag{75}$$

Finally, we also note that if $k = 0$, then the translation function T and the extrapolation matrix E do not depend on the level l, as it is easily seen. It that case, the interpolations and anterpolations also avoid the most problematic case $ka \approx \gamma$.

3.4. Broadband Combinations

For implementing a broadband MLFMA, we have three options using the above translation procedures. The first option is using the spectral representation for all levels. The second option is using the spectral representation for lower levels and switching to the FFT-based MLFMA for higher levels, i.e., when the division cube side length is larger than approximately one wavelength. This combination is very straightforward to implement, as outlined in Section 3.1, since the FFT-based MLFMA uses the same representations for the far-field and incoming wave patterns as the propagating part of the spectral representation, and thus no interpolations between different representations are needed. Finally, the third option is using the UMLFMA for lower levels and switching to the FFT-based MLFMA when $ka \geq \gamma$. This transition is also straightforward to implement, as outlined in Section 3.3.

4. NUMERICAL TESTS FOR ERROR CONTROL

To compare the different translation procedures, we consider the simple benchmark problem of calculating the scalar field $F(\mathbf{r})$ for $\mathbf{r} \in \Omega_f$ due to a planar source $q(\mathbf{r}')$, where $\mathbf{r}' \in \Omega_q$, as shown in Fig. 5.

The source region Ω_q is a planar rectangle consisting of the points

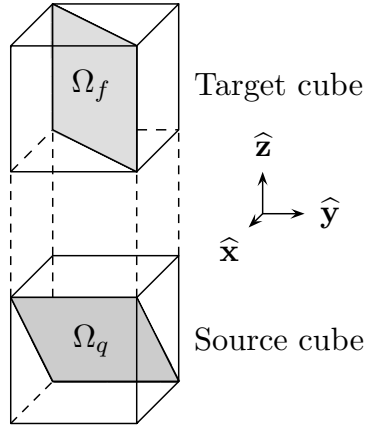

Figure 5. Geometry of the benchmark problem. The source cube is centered at the origin and the side length $a = 2$.

$\mathbf{r}'$ whose Cartesian coordinates are

$$\left|x'\right| \leq 1, \quad \left|y'\right| \leq 1 \quad \text{and} \quad z' = x', \tag{76}$$

while the similar field region Ω_f has the coordinates

$$|x| \leq 1, \quad y = x \quad \text{and} \quad 3 \leq z \leq 5, \tag{77}$$

and the planar source density is constant $q = 1$. The exact field is

$$F(\mathbf{r}) = \int_{\Omega_q} G(\mathbf{r} - \mathbf{r}')q(\mathbf{r}')\,\mathrm{d}S', \quad \text{for } \mathbf{r} \in \Omega_f, \tag{78}$$

which we take to be the discretized integral over Ω_q using a 20×20-point Gauss-Legendre quadrature rule, i.e., we replace the source $q(\mathbf{r}')$ by the point sources arising from the quadrature rule. Next we compute the field $F(\mathbf{r})$ by translation steps as described below, and call the result the *computed field* $\widetilde{F}(\mathbf{r})$. To measure the errors in this computation we estimate the relative L_2 error

$$\frac{\sqrt{\int_{\Omega_f} \left|\widetilde{F}(\mathbf{r}) - F(\mathbf{r})\right|^2 \,\mathrm{d}S}}{\sqrt{\int_{\Omega_f} |F(\mathbf{r})|^2 \,\mathrm{d}S}}, \tag{79}$$

by evaluating the fields F and $\widetilde{F}$ at the points given by a 20×20-point Gauss-Legendre quadrature rule.

The computation is carried out as follows. First, we subdivide the source cube at level l into eight sub-cubes at level $l-1$, where four of the cubes contains 100 source points each and the other ones are empty. Then, we compute the far-field patterns for each of the non-empty sub-cubes from the point sources. Next, we perform the *outer-to-outer translations* from level $l - 1$ to level l by interpolating the far field patterns from level $l - 1$ to level l, and combining them by shifting the origin to the parent cube. Thereafter, we perform the *outer-to-inner translation* from the source cube to the field cube at level l. We also subdivide the field cube into eight sub-cubes, such that the four non-empty sub-cubes contain 100 field points each. Then, we perform the *inner-to-inner translations* from level l to level $l - 1$ by shifting the origin to the level $l - 1$ sub-cubes and then anterpolating the incoming wave patterns to the level $l - 1$ samples. Finally, we evaluate the exact field at the points in the level $l - 1$ sub-cubes and compute the L_2 error-estimate.

For comparison, we also perform the computation as a one level variant, without outer-to-outer and inner-to-inner translations, so that the far-field is computed directly from the sources at level l and the field is computed using the level l incoming wave patterns. Thus, we can demonstrate whether the interpolations and anterpolations introduce any significant additional errors compared to the outer-to-inner translations.

We choose the level numbering such that $ka = 2^l\pi$ at level l, i.e. the division cube side length a is 2^{l-1} wavelengths. Results for the levels -5 to 5 are presented below, corresponding to $a = \lambda/64 \ldots 16\lambda$. However, both the spectral representation and the UMLFMA works down to $k = 0$ essentially as well as at level -5.

The translations are performed using the spectral representation, as described above in Sections 3.1 and 3.2, for all levels. For higher levels, we also perform the translations using the FFT-based MLFMA as briefly outlined in Section 3.1. Finally, we perform the translations using the UMLFMA, as described in Section 3.3, for lower levels.

The number of sample points needed for the spectral representation and the FFT-based MLFMA as well as the degree L of the Rokhlin translation function used in (34) are presented in Table 2 for different levels and two different target accuracies. The degree L is computed using (5) if $L < kD$, and otherwise numerically chosen to get the appropriate accuracy for a worst-case translation. The num-

Table 2. The degree L of the Rokhlin translation function T_L and the number of sample points for the propagating part and the evanescent part, for different levels and two different target precisions.

	Digits $d_0 = 2$			Digits $d_0 = 4$		
level	L	prop.	evan.	L	prop.	evan.
5	191	101×200	2×580(164)	200	109×216	4×598(178)
4	100	55×108	2×294(88)	105	61×120	4×308(100)
3	52	31×60	2×152(50)	65	36×70	5×162(58)
2	30	19×36	3×78(30)	42	22×42	6×88(36)
1	20	12×22	3×42(18)		15×28	7×50(24)
0	14	8×14	4×24(14)		11×20	9×40(20)
-1		6×10	5×20(10)		8×14	10×38(18)
-2		5×8	5×18(10)		6×10	11×38(18)
-3		4×6	6×18(10)		5×8	11×36(18)
-4		3×4	6×18(10)		4×6	11×36(18)
-5		3×4	6×18(10)		4×6	11×36(18)
-6		3×4	6×18(10)		3×4	11×36(18)

ber of sample points for the propagating part, which is the same as for the FFT-based MLFMA, is $N+1$ samples in the θ-direction times $2N$ samples in the φ-direction, with N given by (26). The number of sample points for the evanescent part is given as $M \times 2N(2N')$, with M being the number of sample points in the σ-direction and $2N$ the number of sample points in the φ-direction for the outer-to-inner translations. The far-field and incoming wave patterns are, however, sampled using $M \times 2N'$ points, and also the outer-to-outer and inner-to-inner translations are performed using these lower number of points in the φ-direction, as outlined in Section 3.2.

For the UMLFMA we use the same number of sample points and parameters for all levels: $N_t = 9$ points in the t-direction, $N_\varphi = 15$ points in the φ-direction, and the parameters $\gamma = 3$ and $\tau = 10^{-5}$. These numbers are selected, by numerical experiments, to give two digits accuracy in the outer-to-inner translations.

Table 3 presents the results for all methods aiming at two digits accuracy. The level l refers to the level where the outer-to-inner translation is performed. For the one-level variant this is the only level, and for the two-level variant this is the upper level. We observe that all methods give good results for the one-level version, but in the two-level version we observe significant differences. The accuracy of the spectral representation decreases slightly at higher levels, mainly due to the fact that the σ-interpolations and anterpolations become less accurate. This is probably insignificant, especially since it seems more optimal to switch over to the FFT-based MLFMA at higher levels

Table 3. Relative L_2 errors for the different translation procedures aiming at two digits accuracy.

	Spectral repr.		FFT-MLFMA		UMLFMA	
level	1 level	2 levels	1 level	2 levels	1 level	2 levels
5	4.73e-04	1.14e-03	1.49e-06	2.35e-06		
4	4.12e-04	7.72e-04	1.76e-05	2.01e-05		
3	1.02e-03	1.38e-03	2.13e-05	2.21e-05		
2	5.26e-04	1.86e-03	2.00e-05	2.94e-05		
1	2.16e-03	2.11e-03	9.42e-05	1.43e-04		
0	2.97e-03	2.92e-03	1.72e-04	1.72e-04		
-1	3.17e-03	3.19e-03			1.81e-04	1.24e-02
-2	4.85e-03	4.85e-03			1.26e-04	4.72e-03
-3	6.29e-04	6.34e-04			1.20e-04	3.51e-03
-4	6.31e-04	6.31e-04			1.22e-04	3.43e-03
-5	6.31e-04	6.31e-04			1.23e-04	3.43e-03

anyway.

The more dramatic decrease in accuracy, when comparing the two-level version with the one-level version, happens for the UMLFMA. At lower levels, we loose almost one significant digit and near the transition region $ka \approx \gamma$ the situation is even worse. The accuracy in the interpolations and anterpolations can be improved by choosing a smaller γ, but then the accuracy in the outer-to-inner translations decreases. Furthermore, we then push the problematic transition region $ka \approx \gamma$ to lower levels, making it less straightforward to switch over to the FFT-based MLFMA.

Table 4. Error controllability of the spectral representation and the FFT-based MLFMA. The numbers are the relative L_2-errors for the two-level version at different levels and two different target precisions d_0.

	Spectral repr.		FFT-MLFMA	
level	$d_0 = 2$	$d_0 = 4$	$d_0 = 2$	$d_0 = 4$
5	1.14e-03	6.41e-06	2.35e-06	1.03e-08
4	7.72e-04	4.88e-06	2.01e-05	1.24e-07
3	1.38e-03	6.76e-06	2.21e-05	2.51e-07
2	1.86e-03	7.05e-06	2.94e-05	4.53e-07
1	2.11e-03	4.92e-05	1.43e-04	
0	2.92e-03	3.23e-05	1.72e-04	
-1	3.19e-03	1.65e-05		
-2	4.85e-03	9.59e-06		
-3	6.34e-04	7.28e-06		
-4	6.31e-04	9.35e-06		
-5	6.31e-04	5.46e-06		

Table 4 demonstrates the error controllability of the spectral representation and the FFT-based MLFMA. By increasing the number of sample points as given in Table 2 we of course increase the computational cost, but also the accuracy increases with approximately two digits as wanted. Here, we compare the relative L_2-errors for the two-level version only. Finally, we note that it appears impossible to increase the accuracy in the UMLFMA by two digits.

5. CONCLUSIONS

In this paper, we have considered the translation procedures needed to efficiently implement a broadband MLFMA. To overcome the low-frequency breakdown, we have considered two representations: the

spectral representation, also known as the inhomogeneous plane-wave expansion, and the UMLFMA.

For the spectral representation, we have combined the best parts of the previously reported implementations [6–8] and also proposed some new improvements. For the propagating part, the use of an entirely FFT-based approach, as described in [12] for the traditional MLFMA, seems to be the best available method. The error controllability is excellent, a minimal number of sample points needs to be stored and it is trivial to switch from the spectral representation to the FFT-based traditional MLFMA at higher levels. Also in [8], an FFT-based approach is briefly mentioned as an alternative, but apparently not implemented.

For the evanescent part of the spectral representation, we have used a generalized Gaussian quadrature rule [9] for the σ-direction as is also used in [6, 8]. This appears to be the most efficient quadrature rule available. The most novel feature of our proposed implementation of the evanescent part is, however, the observation that we can efficiently perform the interpolations and anterpolations in the σ-direction using the sample points and interpolation matrices. Compared to [6, 8] this is much more straightforward and perhaps even more efficient. The only drawback is that we need a fairly large number of samples, thus making the computations faster at the cost of an increased memory-consumption.

For comparison we have also described a somewhat modified implementation of the UMLFMA [10], and compared the error controllability of the two different representations. The UMLFMA is promising, since the computational cost is considerably smaller and the method is simpler to implement, but the error controllability is poor. For many applications, the limited accuracy may be sufficient, but it seems impossible to get significantly higher accuracies without changing method. The spectral representation, on the other hand, is error controllable, as we have demonstrated.

REFERENCES

1. Song, J. M. and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *Microwave Opt. Tech. Lett.*, Vol. 10, 14–19, Sept. 1995.
2. Chew, W. C., J. M. Jin, E. Michielssen, and J. M. Song (eds.), *Fast and Efficient Algorithms in Computational Electromagnetics*, Artech House, 2001.
3. Rokhlin, V., "Diagonal form of translation operators for

the Helmholtz equation in three dimensions," *Applied and Computational Harmonic Analysis*, Vol. 1, 82–93, Dec. 1993.

4. Coifman, R., V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: a pedestrian prescription," *IEEE Ant. Prop. Mag.*, Vol. 35, 7–12, June 1993.
5. Michielssen, E. and W. C. Chew, "Fast steepest descent path algorithm for analyzing scattering from two-dimensional objects," *Radio Science*, Vol. 31, 1215–1224, Sep.–Oct. 1996.
6. Greengard, L., J. Huang, V. Rokhlin, and S. Wadzura, "Accelerating fast multipole methods for the Helmholtz equation at low frequencies," *IEEE Comput. Sci. Eng.*, Vol. 5, 32–38, Jul.–Sep. 1998.
7. Jiang, L. J. and W. C. Chew, "Low-frequency fast inhomogeneous plane-wave algorithm (LF-FIPWA)," *Microwave Opt. Tech. Lett.*, Vol. 40, 117–122, Jan. 2004.
8. Darve, E. and P. Havé, "A fast multipole method for Maxwell equations stable at all frequencies," *Phil. Trans. R. Soc. Lond. A*, Vol. 362, 603–628, Mar. 2004.
9. Yarvin, N. and V. Rokhlin, "Generalized Gaussian quadratures and singular value decompositions of integral operators," *SIAM J. Sci. Comput.*, Vol. 20, No. 2, 699–718, 1998.
10. Xuan, L., A. Zhu, R. J. Adams, and S. D. Gedney, "A broadband multilevel fast multipole algorithm," in *IEEE AP-S International Symposium and USNC/URSI National Radio Science Meeting*, Vol. 2, 1195–1198, Monterey, CA, June 2004.
11. Hastriter, M. L., S. Ohnuki, and W. C. Chew, "Error control of the translation operator in 3D MLFMA," *Microwave Opt. Tech. Lett.*, Vol. 37, 184–188, May 2003.
12. Sarvas, J., "Performing interpolation and anterpolation entirely by fast fourier transform in the 3D multilevel fast multipole algorithm," *SIAM J. Numer. Anal.*, Vol. 41, No. 6, 2180–2196, 2003.
13. Bucci, O. M. and G. Franceschetti, "On the spatial bandwidth of scattered fields," *IEEE Trans. Antennas Propagat.*, Vol. AP-35, 1445–1455, Dec. 1987.
14. Kress, R., *Linear Integral Equations*, *Applied Mathematical Sciences*, second ed., Vol. 82, Springer-Verlag, New York, 1999.

Henrik Wallén was born in 1975. He received the M.Sc. (Tech.) degree in Electrical Engineering in 2000 from the Helsinki University of Technology (HUT), Espoo, Finland. He is currently doing his postgraduate studies working at the HUT Electromagnetics Laboratory. His main research interests include electromagnetic theory and fast computational methods.

Jukka Sarvas received the M.Sc. degree in 1968 and the Ph.D. degree in 1972, both in mathematics, in the University of Helsinki. He has been in the University of Helsinki, in the Helsinki University of Technology, and 1982–1984 in Outokumpu Co. In 1988–2002 he was the director of the Rolf Nevanlinna Research Institute of Applied Mathematics and Statistics in the University of Helsinki. He has been a visiting researcher 1974–1975 and 1979–1980 in the University of Michigan, and 1999–2000 in the University of Illinois at Urbana-Champaign. Since 2002 he has been a professor in computational electromagnetics in Electromagnetics Laboratory of the Helsinki University of Technology. His main interests include the field computing with integral equations and the fast methods.