Fast Computation of RCS by Fast Multipole Method in Conjunction with Lifting Wavelet-Like Transform

M. S. Chen[1], W. Sha[2], Q. Wu[3], Z. X. Huang[3] and X. L. Wu[1]

1. Department of Physics and Electronic Engineering, Hefei Teachers College, 327 Jinzhai Road, Hefei 230061, China

Email: cmsh@ahu.edu.cn

2.Department of Electrical and Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong, China

3. School of Electronic Science and Technology, Anhui University, Hefei, 230039, China

***Abstract*—The fast multipole method (FMM) in conjugation with the lifting wavelet-like transform (LWLT) scheme is proposed for the scattering analysis of different shaped three-dimensional (3-D) perfectly electrical conducting (PEC) objects. As an in space matrix compression technique, the method proposed can sparsify the aggregation matrix and disaggregation matrix in time. The computational complexity and choice of proper wavelet are also discussed. Numerical simulation and complexity analysis have shown that the method proposed can speed up agggregation/disaggregation step in FMM with lower memory consumed.**

***Index Terms*—Lifting wavelet like transform, Method of moments, Fast multipole method, Electromagnetic scattering.**

I. INTRODUCTION

Electromagnetic scattering problems of arbitrarily shaped three-dimensional (3-D) objects can be dealt with by the method of moments (MOM) [1], which has been widely used and deeply researched over the past decades. In MOM, a dense matrix equation of order $N \times N$ always generated when the surface current density of the object is approximated by N basis functions. The solution of the matrix equation by iterative methods requires a computational complexity of $O(N^2)$.

Over past few years, a number of new techniques have been proposed to reduce the complexity of matrix-vector multiplication (MVM) in various iterative methods. In these new techniques, there are the fast multipole method (FMM) [2, 3], the adaptive integral method (AIM) [4], and the conjugate gradient-fast Fourier transform (CG-FFT) algorithm [5], etc. The fast multipole method and the multilevel fast multiple algorithm (MLFMA) [6, 7] are well known among them, which can reduce $O(N^2)$ for MVM to $O(N^{1.5})$ and $O(N \log N)$, respectively.

Another interesting technique is the so-called wavelet transform method [8-10]. Generally, wavelets have been applied to the solution of integral equations in two ways. One is to be used directly as basis functions and test functions in MOM [11, 12]. The other is the discrete wavelet transform (DWT), which is applied to the impedance matrix to obtain a sparse matrix equation in wavelet-domain [13, 14]. The applications of wavelet matrix transform have been widely used during the past decades. Unfortunately, they are mainly confined to the analysis of two-dimensional (2-D) problems, or to special structures such as wire in which the current direction is one-dimensional.

Recently, following the wide application of FMM in computational electromagnetic, many approach has been proposed to speed up it. A new matrix compression technique based on singular value decomposition (SVD) is proposed in [15] to compress the aggregation matrix of the FMM. By the matrix compressing technique proposed in [16], Martinsson and Rokhlin successfully accelerated the kernel-independent FMM in one dimension [17].

In this paper, lifting wavelet-like transform (LWLT), as a novel method presented in [18], is applied to the FMM to compress the elements in aggregation matrix and disaggregation matrix in time, which can further speed up the MVM and save much memory when the FMM is used. Numerical results for different shaped three-dimensional objects are considered and computational complexity analysis is also considered. Compared with the FMM technique, the application of the LWLT to the FMM can accelerate the MVM by factor of two and can save memory with proper wavelets selected.

II. THEORY

A. Fast Multipole Method

The electromagnetic scattering analysis of a 3-D arbitrarily shaped conductor can be operated by finding the solution of an integral equation where the unknown function is the induced current distribution. For perfectly electrically conducting (PEC) objects illuminated by an incident field $\mathbf{E}^i(\mathbf{r})$, the electric field integral equation (EFIE) is given by

$$\hat{\mathbf{t}} \cdot \int_S \bar{\mathbf{G}}(\mathbf{r},\mathbf{r}') \cdot \mathbf{J}(\mathbf{r}') dS' = \frac{1}{jk\eta} \hat{\mathbf{t}} \cdot \mathbf{E}^i(\mathbf{r}) \tag{1}$$

for $\mathbf{r}$ on S, where $\mathbf{J}$ is the unknown current distribution, $\hat{\mathbf{t}}$ is any unit tangent vector on surface S, and $\bar{\mathbf{G}}(\mathbf{r},\mathbf{r}')$ is the well-known free-space dyadic Green's function.

When we apply the method of moments to EFIE, it will yield a matrix equation

$$[Z_{mn}][x_n] = [F_m] \tag{2}$$

where x_n is the unknown current coefficients and

$$Z_{mn} = \int_S dS \mathbf{t}_m(\mathbf{r}) \bullet \int_S dS' \bar{\mathbf{G}}(\mathbf{r},\mathbf{r}') \bullet \mathbf{j}_n(\mathbf{r}') \tag{3}$$

$$F_m = \frac{1}{jk\eta} \int_S dS \mathbf{t}_m(\mathbf{r}) \bullet \mathbf{E}^i(r) \tag{4}$$

To speed up the matrix-vector multiplication in solution of equation (2), the fast multipole method is introduced [19], which provides a sparse decomposition of the dense matrix $[Z_{mn}]$ in (2):

$$\mathbf{Z} = \mathbf{Z}^{near} + \mathbf{V}^H T \mathbf{V} \tag{5}$$

where $\mathbf{Z}^{near}$ represents nearby interactions and $\mathbf{V}^H T\mathbf{V}$ the rest.

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} & \cdots & \mathbf{V}_{1N_j} \\ \mathbf{V}_{21} & \mathbf{V}_{22} & \cdots & \mathbf{V}_{2N_j} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{V}_{K1} & \mathbf{V}_{K2} & \cdots & \mathbf{V}_{KN_j} \end{pmatrix} \tag{6}$$

$$T = \begin{pmatrix} T_1 & & & \\ & T_2 & & \\ & & \ddots & \\ & & & T_K \end{pmatrix} \tag{7}$$

$$\mathbf{V}_{pn} = \int_{S'} (\bar{\mathbf{I}} - \hat{\mathbf{k}}_p \hat{\mathbf{k}}_p) \bullet \mathbf{j}_n(\mathbf{r}') e^{j\mathbf{k}_p \bullet (\mathbf{r}' - \mathbf{r}'_o)} dS' \tag{8}$$

$$T_p = \frac{k^2 \eta}{16\pi^2} \omega_p \sum_{l=0}^{L} (-j)^l (2l+1) h_l^{(2)}(kX) P_l(\hat{\mathbf{k}}_p \bullet \hat{\mathbf{X}})$$

$$p = 1, 2, \cdots, 2L^2 \tag{9}$$

Since $\bar{\mathbf{I}} - \hat{\mathbf{k}}_p \hat{\mathbf{k}}_p = \hat{\theta}\hat{\theta} + \hat{\phi}\hat{\phi}$, then $\mathbf{V}_{mp}$ has only θ and φ components. When the number of groups is chosen to be $\sqrt{N}$, the computation complexity and memory consumed is reduced to be $O(N^{\frac{3}{2}})$.

B. FMM Speed up by Lifting Wavelet- like Transform

Following the wide application of FMM, how to further accelerate $\mathbf{V}^H T\mathbf{V}\mathbf{x}$ and reduce memory consumed has been a hot topic. The lifting wavelet-like transform is applied in this paper to sparsify the aggregation and disaggregation matrix.

In traditional wavelet-like transform, wavelet transform matrices $\tilde{\mathbf{W}}$ and $\mathbf{W}$ are always constructed. Using the identity

$$\mathbf{W}\tilde{\mathbf{W}} = \tilde{\mathbf{W}}\mathbf{W} = \mathbf{I} \tag{10}$$

The far field interactions will be represented by

$$\mathbf{V}^H T\mathbf{V}\mathbf{x} = \mathbf{V}^H \mathbf{W}\tilde{\mathbf{W}} T \mathbf{W}\tilde{\mathbf{W}}\mathbf{V}\mathbf{x} \tag{11}$$

Setting $\tilde{\mathbf{V}}^* = \mathbf{V}^H \mathbf{W}$ and $\tilde{\mathbf{V}} = \tilde{\mathbf{W}}\mathbf{V}$, one can obtain

$$\mathbf{V}^H T\mathbf{V}\mathbf{x} = \tilde{\mathbf{V}}^* \tilde{\mathbf{W}} T \mathbf{W} \tilde{\mathbf{V}} \mathbf{x} \tag{12}$$

Then the MVM will be completed by the following steps:

Firstly, $\mathbf{V}^H$, $\mathbf{V}$ and $\mathbf{T}$ are generated and wavelet matrix transform is applied in time by $\tilde{\mathbf{V}}^* = \mathbf{V}^H \mathbf{W}$ and $\tilde{\mathbf{V}} = \tilde{\mathbf{W}}\mathbf{V}$, then $\tilde{\mathbf{V}}^*$ and $\tilde{\mathbf{V}}$ is a sparse matrix by the threshold σ_m.

Secondly, complete aggregation by $\mathbf{x}_1 = \tilde{\mathbf{V}}\mathbf{x}$, and the inverse wavelet transform for $\mathbf{x}_1$ is implement by $\mathbf{x}_2 = \mathbf{W}\mathbf{x}_1$.

Thirdly, complete translation by $\mathbf{x}_3 = \mathbf{T}\mathbf{x}_2$.

Finally, the forward wavelet transform for $\mathbf{x}_3$ is applied by $\mathbf{x}_4 = \tilde{\mathbf{W}}\mathbf{x}_3$, and the disaggregation is completed by $\mathbf{x}_5 = \tilde{\mathbf{V}}^*\mathbf{x}_4$.

To save up CPU time and memory consumed for transform matrix, the lifting wavelet-like transform is introduced to complete the forward transform and inverse transform. In the LWLT scheme, the wavelet transform is directly operated to the object matrix according to factorization of the polyphase matrices

$$\tilde{\mathbf{P}}(z^{-1})^t = \prod_{i=1}^{m} \begin{pmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{pmatrix} \begin{pmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & 1/F \end{pmatrix} \tag{14}$$

$$\mathbf{P}(z) = \prod_{i=1}^{m} \begin{pmatrix} 1 & s_i(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ t_i(z) & 1 \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & 1/F \end{pmatrix} \tag{15}$$

in which $s_i(z)$ and $t_i(z)$ are Laurent polynomials, and F is a nonzero constant.

Without the help of wavelet transform matrices $\tilde{\mathbf{W}}$ and $\mathbf{W}$, the transform can also be described by predict step and update step in lifting scheme, which can be outlined in the following three basic operations.

Split: Divide the original data ($\mathbf{x}[n]$) into odd subsets ($\mathbf{x}_o[n]$) and even subsets ($\mathbf{x}_e[n]$)

$$\begin{cases} \mathbf{x}_o[n] = \mathbf{x}[2n-1] \\ \mathbf{x}_e[n] = \mathbf{x}[2n] \end{cases} \tag{16}$$

Predict: Generate high frequency component $\mathbf{d}[n]$ as the error in predicting odd subsets from even subsets with prediction operator Q .

$$\mathbf{d}[n] = \mathbf{x}_o[n] - Q(\mathbf{x}_e[n]) \tag{17}$$

Update: Generate low frequency component $\mathbf{c}[n]$ as a coarse similarity to original signal by applying an update operator U to the high frequency component and being added to even subsets

$$\mathbf{c}[n] = \mathbf{x}_e[n] + U(\mathbf{d}[n]) \tag{16}$$

The operators Q and U above can be deduced from the polyphase matrices described in (14) and (15). The forward transform is implemented according to $\tilde{\mathbf{P}}(z^{-1})^t$ and the inverse transform is operated by $\mathbf{P}(z)$, and specific examples can be found in [18]. Take the lifting scheme of Haar wavelet for example to give a brief interpretation for how LWLT works.

For Haar wavelet, the polyphase matrix $\tilde{\mathbf{P}}(z^{-1})^t$ is given as

$$\tilde{\mathbf{P}}(z^{-1})^t = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{1}{2} & \sqrt{2} \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{17}$$

As described above, when the lifting scheme of Haar wavelet applied, the signal (the row or column in the matrix) will undergo three basic operations as below:

$$\mathbf{x}_o(l) = \mathbf{x}(2l+1) \tag{18}$$

$$\mathbf{x}_e(l) = \mathbf{x}(2l) \tag{19}$$

$$\mathbf{x}_o^{(1)}(l) = \mathbf{x}_o(l) - \mathbf{x}_e(l) \tag{20}$$

$$\tilde{\mathbf{x}}_e(l) = \sqrt{2}\mathbf{x}_e(l) + 0.5 \cdot \mathbf{x}_o^{(1)}(l) \tag{21}$$

$$\tilde{\mathbf{x}}_o(l) = \frac{\sqrt{2}}{2}\mathbf{x}_o^{(1)} \tag{22}$$

Just after one level transform, the fluctuation term $\tilde{\mathbf{x}}_o$, which can be predicted by the difference of the odd and even components, will contain the minor information of the aggregation matrix $\tilde{\mathbf{V}}$ (or disaggregation matrix $\tilde{\mathbf{V}}^*$). While the leading term $\tilde{\mathbf{x}}_e$, which can be updated by the fluctuation term and the even component, will include the major information and properties of $\tilde{\mathbf{V}}$ (or $\tilde{\mathbf{V}}^*$). The fluctuation information of the matrices will leads to the fluctuation of currents. And the noise-like fluctuation in currents is expected to be very small and random, which will play little contribution for far field scattering analysis and can be neglected, hence many small elements in $\tilde{\mathbf{x}}_o$ can be set to be zero. In the second level, the similar operations will be used to $\tilde{\mathbf{x}}_e$, and the more sparse matrix will be obtained after multilevel transform.

As can be seen from the above description, the sparsified aggregation matrix $\tilde{\mathbf{V}}$ and disaggregation matrix $\tilde{\mathbf{V}}^*$ are obtained while two auxiliary lifting wavelet-like transform is added to the MVM, and the complexity of it will be analyzed in the following sections.

C. *Choice of Proper Wavelet*

Before the discussion about choice of proper wavelet, we want to give a brief introduce for the lifting operations for Daubechies wavelets with different varnishing moments. Similarly to the description about Haar wavelet in previous section, the forward transform $\tilde{\mathbf{x}} = \tilde{\mathbf{W}}\mathbf{x}$ for Daubechies wavelets with two varnishing moments (db2) is implemented according to factorization of the polyphase matrix $\tilde{\mathbf{P}}(z^{-1})^t$

$$\tilde{\mathbf{P}}(z^{-1})^t = \begin{bmatrix} \frac{\sqrt{3}+1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3}-2}{4} z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sqrt{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{23}$$

and can be operated by following steps:

$$\mathbf{x}_e(l) = \mathbf{x}(2l) \;\; ; \;\; \mathbf{x}_o(l) = \mathbf{x}(2l+1) \tag{24}$$

$$\mathbf{x}_e^{(1)}(l) = \mathbf{x}_e(l) \; ; \; \mathbf{x}_o^{(1)}(l) = \mathbf{x}_o(l) - \sqrt{3}\mathbf{x}_e(l) \tag{25}$$

$$\mathbf{x}_e^{(2)}(l) = \mathbf{x}_e^{(1)}(l) + \frac{\sqrt{3}}{4}\mathbf{x}_o^{(1)}(l) + \frac{\sqrt{3}-2}{4}\mathbf{x}_o^{(1)}(l+1) \; ; \;\; \mathbf{x}_o^{(2)}(l) = \mathbf{x}_o^{(1)}(l) \tag{26}$$

$$\mathbf{x}_e^{(3)}(l) = \mathbf{x}_e^{(2)}(l) \; ; \; \mathbf{x}_o^{(3)}(l) = \mathbf{x}_o^{(2)}(l) + \mathbf{x}_e^{(2)}(l-1) \tag{27}$$

$$\mathbf{x}_e^{(4)}(l)=\frac{\sqrt{3}+1}{\sqrt{2}}\mathbf{x}_e^{(3)}(l)\ ;\ \mathbf{x}_o^{(4)}(l)=\frac{\sqrt{3}-1}{\sqrt{2}}\mathbf{x}_o^{(3)}(l) \tag{28}$$

$$\tilde{\mathbf{x}}_e=\mathbf{x}_e^{(4)}(l)\ ;\ \tilde{\mathbf{x}}_o=\mathbf{x}_o^{(4)}(l) \tag{29}$$

While the inverse transform is corresponding to $\mathbf{P}(z)$ and is omitted here for the sake of brevity.

Similarly, the implemention of forward transform for Daubechies wavelets with four varnishing moments (db4) is given as below:

$$\mathbf{x}_e(l)=\mathbf{x}(2l)\ ;\ \ \mathbf{x}_o(l)=\mathbf{x}(2l+1) \tag{30}$$

$$\mathbf{x}_e^{(1)}(l)=\mathbf{x}_e(l)\ ;\ \mathbf{x}_o^{(1)}(l)=\mathbf{x}_o(l)+d1(1)\mathbf{x}_e(l+1) \tag{31}$$

$$\mathbf{x}_e^{(2)}(l)=\mathbf{x}_e^{(1)}(l)+p1(1)\mathbf{x}_o^{(1)}(l)+p1(2)\mathbf{x}_o^{(1)}(l-1)\ ;\ \mathbf{x}_o^{(2)}(l)=\mathbf{x}_o^{(1)}(l) \tag{32}$$

$$\mathbf{x}_e^{(3)}(l)=\mathbf{x}_e^{(2)}(l)\ ;\ \mathbf{x}_o^{(3)}(l)=\mathbf{x}_o^{(2)}(l)+d2(2)\mathbf{x}_e^{(2)}(l+1)+d2(1)\mathbf{x}_e^{(2)}(l+2) \tag{33}$$

$$\mathbf{x}_e^{(4)}(l)=\mathbf{x}_e^{(3)}(l)+p2(1)\mathbf{x}_o^{(3)}(l)+p2(2)\mathbf{x}_o^{(3)}(l-1)\ ;\ \mathbf{x}_o^{(4)}(l)=\mathbf{x}_o^{(3)}(l) \tag{34}$$

$$\mathbf{x}_e^{(5)}(l)=\mathbf{x}_e^{(4)}(l)\ ;\ \mathbf{x}_o^{(5)}(l)=\mathbf{x}_o^{(4)}(l)+d3(3)x_e^{(4)}(l-2)+d3(2)x_e^{(4)}(l-1)+d3(1)x_e^{(4)}(l) \tag{35}$$

$$\tilde{\mathbf{x}}_e=p3(1)\mathbf{x}_e^{(4)}(l)\ ;\ \tilde{\mathbf{x}}_o=d4(1)\mathbf{x}_o^{(4)}(l) \tag{36}$$

The coefficients mentiond above can be get from Matlab 7.0 by command function "liftwave( )", and are given in Table I.

TABLE I The coefficients in lifting scheme for db4 wavelet

Predict Coefficients	Update Coefficients
d1 -0.3223	p1 -1.1171 -0.3001
d2 -0.0188 0.1176	p2 2.1318 0.6364
d3 -0.4691 0.1400 -0.0248	p3 0.7341
d4 1.3622	

Finally, one can accomplish the forward transform for Daubechies wavelets with eight varnishing moments (db8) by following steps:

$$\mathbf{x}_e(l)=\mathbf{x}(2l)\ ;\ \ \mathbf{x}_o(l)=\mathbf{x}(2l+1) \tag{37}$$

$$\mathbf{x}_e^{(1)}(l)=\mathbf{x}_e(l)\ ;\ \mathbf{x}_o^{(1)}(l)=\mathbf{x}_o(l)+d^{(1)}(1)\mathbf{x}_e(l) \tag{38}$$

$$\mathbf{x}_e^{(2)}(l) = \mathbf{x}_e^{(1)}(l) + p^{(1)}(1)\mathbf{x}_o^{(1)}(l+1) + p^{(1)}(2)\mathbf{x}_o^{(1)}(l)\,;\ \mathbf{x}_o^{(2)}(l) = \mathbf{x}_o^{(1)}(l) \tag{39}$$

$$\mathbf{x}_e^{(3)}(l) = \mathbf{x}_e^{(2)}(l)\,;\ \mathbf{x}_o^{(3)}(l) = \mathbf{x}_o^{(2)}(l) + d^{(2)}(2)\mathbf{x}_e^{(2)}(l-2) + d^{(2)}(1)\mathbf{x}_e^{(2)}(l-1) \tag{40}$$

$$\mathbf{x}_e^{(4)}(l) = \mathbf{x}_e^{(3)}(l) + p^{(2)}(1)\mathbf{x}_o^{(3)}(l+3) + p^{(2)}(2)\mathbf{x}_o^{(3)}(l+2)\,;\ \mathbf{x}_o^{(4)}(l) = \mathbf{x}_o^{(3)}(l) \tag{41}$$

$$\mathbf{x}_e^{(5)}(l) = \mathbf{x}_e^{(4)}(l)\,;\ \mathbf{x}_o^{(5)}(l) = \mathbf{x}_o^{(4)}(l) + d^{(3)}(2)x_e^{(4)}(l-4) + d^{(3)}(1)x_e^{(4)}(l-3) \tag{42}$$

$$\mathbf{x}_e^{(6)}(l) = \mathbf{x}_e^{(5)}(l) + p^{(3)}(1)\mathbf{x}_o^{(5)}(l+5) + p^{(3)}(2)\mathbf{x}_o^{(5)}(l+4)\,;\ \mathbf{x}_o^{(6)}(l) = \mathbf{x}_o^{(5)}(l) \tag{43}$$

$$\mathbf{x}_e^{(7)}(l) = \mathbf{x}_e^{(6)}(l)\,;\ \mathbf{x}_o^{(7)}(l) = \mathbf{x}_o^{(6)}(l) + d^{(4)}(2)x_e^{(6)}(l-4) + d^{(4)}(1)x_e^{(6)}(l-3) \tag{44}$$

$$\mathbf{x}_e^{(8)}(l) = \mathbf{x}_e^{(7)}(l) + p^{(4)}(1)\mathbf{x}_o^{(7)}(l+5) + p^{(4)}(2)\mathbf{x}_o^{(7)}(l+4)\,;\ \mathbf{x}_o^{(8)}(l) = \mathbf{x}_o^{(7)}(l) \tag{45}$$

$$\mathbf{x}_e^{(9)}(l) = \mathbf{x}_e^{(8)}(l)\,;\ \mathbf{x}_o^{(9)}(l) = \mathbf{x}_o^{(8)}(l) + d^{(5)}(3)x_e^{(8)}(l-7) + d^{(5)}(2)x_e^{(8)}(l-6) + d^{(5)}(1)x_e^{(8)}(l-5) \tag{46}$$

$$\tilde{\mathbf{x}}_e(l) = p^{(5)}(1)\mathbf{x}_e^{(9)}(l)\,;\ \tilde{\mathbf{x}}_o(l) = d^{(6)}(1)\mathbf{x}_o^{(9)}(l) \tag{47}$$

The lifting coefficients for db8 are given in Table II.

TABLE II The coefficients in lifting scheme for db8 wavelet

Predict Coefficients	Update Coefficients
d1 -5.7496	p1 -0.0523 0.1688
d2 14.5428 -7.4021	p2 -0.0324 0.0609
d3 5.8187 -2.7557	p3 0.9453 0.2420
d4 0.0002 -0.0018	p4 -0.9526 -0.2241
d5 1.0497 -0.2470 0.0272	p5 3.5494
d6 0.2817	

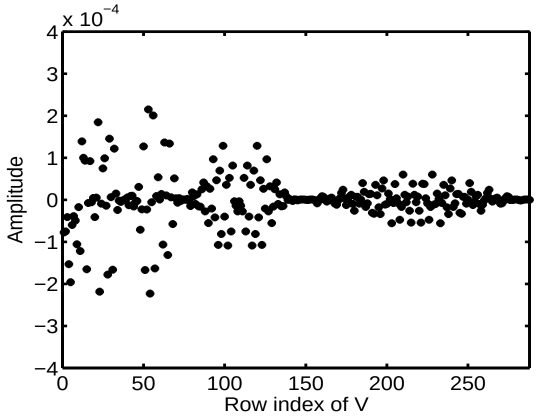
The number of multiplication operations within the lifting scheme is set to be q which can be counted from the polyphase matrices. If the length of the signal is K, the total computational complexity for implementation lifting scheme can be computed by

$$q \times \frac{K}{2} \times (1 + \frac{1}{2} + \frac{1}{4} + ...) = qK \tag{48}$$

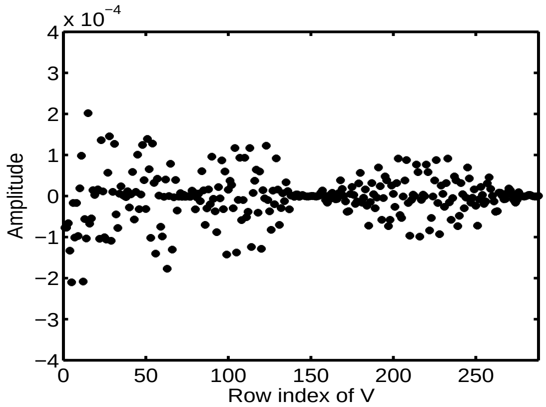
and for Haar, db2, db4, db8, the value of q is 3, 5, 12, 20, respectively.

To select proper wavelets tailored to FMM, take a PEC sphere with its diameter of 5λ ($L = 12$)

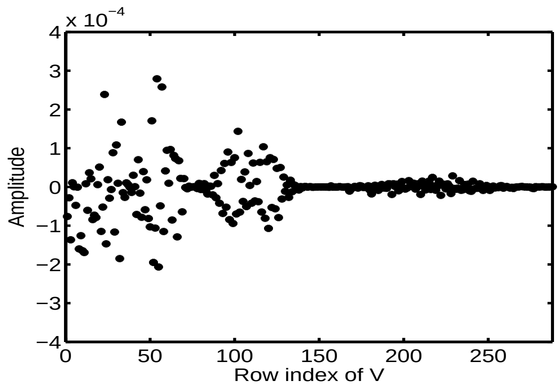
for example, Haar and Daubechies wavelets with different varnishing moments are applied and compared here. After the lifting wavelet-like transform by Haar, db2, db4, and db8, a column of $\tilde{\mathbf{V}}$ is presented in Figure 1.



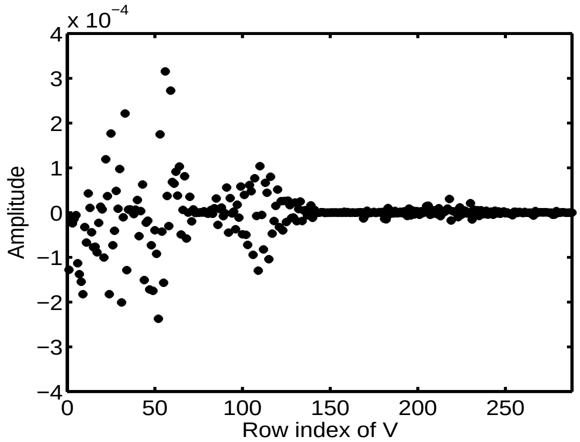
(a) The elements distribution after Haar wavelet transform.



(b) The elements distribution after two layer db2 transform.



(c) The elements distribution after two layer db4 transform.



(d) The elements distribution after two layer db8 transform.

Fig.1 The elements distribution in disaggregation matrix after the application of LWLT

As one can see from the above figure, and considering the transform complexity and the sparsity, db4 is a better choice.

For a field group with M_i RWG functions and a source group with N_j RWG functions, there are $2L^2$ elements in each column of $\mathbf{V}$, and in the presented scheme, the LWLT is actually applied to $\mathbf{V}$ column by column. Take the matrix $\mathbf{V}$ for example, once a certain column is generated and the transform for it is implemented, then the clipping operation is used with the threshold and only the left elements in the column are stored. So the proposed method is an in-space and in-time compressing technique. The threshold for the m column is defined by

$$\sigma_m = \tau \cdot \frac{1}{K} \sum_{p=1}^{K} \left(\left| \tilde{\mathbf{V}}_\theta(p,m) \right|^2 + \left| \tilde{\mathbf{V}}_\varphi(p,m) \right|^2 \right)^{\frac{1}{2}} \tag{49}$$

and numerical simulations show that $\tau \in [0.8, 1.2]$, and the accuracy is controlled by τ.

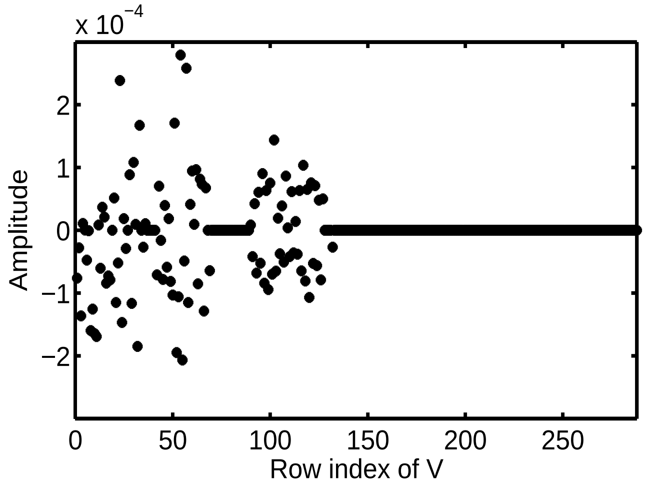


Fig.2 The elements distribution in aggregation matrix after the application of clipping technique

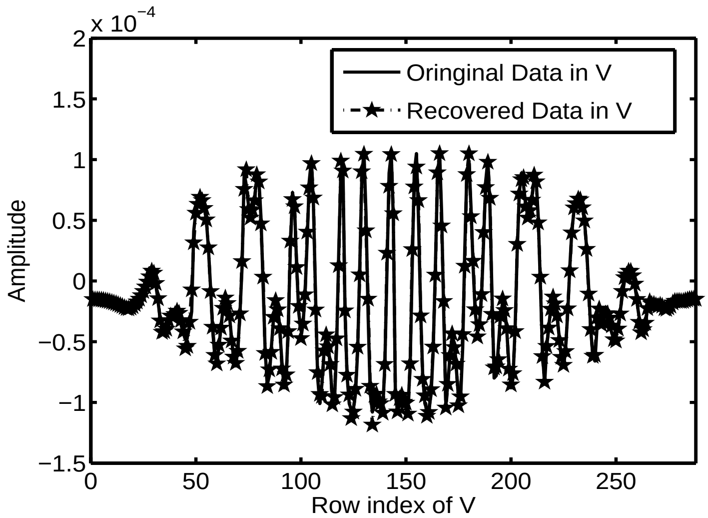


Fig.3 The elements distribution in aggregation matrix obtained by inverse lifting wavelet transform from the column presented in Fig. 2, which is compared with the original column in $\mathbf{v}$.

As can be seen from (c) in Fig.1, most of the elements are far smaller than the others. The results after the clipping operation ($\tau = 0.9$) is shown in Fig.2 and only about 30% of the total elements are left, then the inverse LWLT is implemented and the new column after transform is given in Fig.3 which agrees well with the original column in $\mathbf{V}$.

D. Discussions about the computational complexity in MVM for far-field interactions

As we have mentioned above, when we obtained two sparse matrix $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{V}}^*$, two additional lifting wavelet-like transform is also added. To analysis the total complexity, take conjugate-gradient (CG) method for example, where exists two MVM $[Z_{mn}][\alpha_n]$ and $[Z_{mn}]^T[\alpha_n]$ for each iterative operation. For a problem with N unknowns, when the total group number is set to be $\sqrt{N}$, and take into the operations for translation, the effectiveness can be evaluated by

$$\xi = \frac{4qNK + 8nq\sqrt{N}K + 8\rho nNK + 4nNK}{12nNK} = \frac{q}{3n} + \frac{2q}{3\sqrt{N}} + \frac{2\rho + 1}{3} \tag{50}$$

where ρ is the sparsity of $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{V}}^*$, n is the total number of iterative operations and $K = 2L^2$. $4pNK$ is consumed by the transformation of $\mathbf{V}_\theta$, $\mathbf{V}_\varphi$, $\mathbf{V}_\theta^H$ and $\mathbf{V}_\varphi^H$, $8np\sqrt{N}K$ is for $\mathbf{x}_2 = \mathbf{W}\mathbf{x}_1$ and $\mathbf{x}_4 = \tilde{\mathbf{W}}\mathbf{x}_3$, $8\rho nNK$ is consumed by $\mathbf{x}_1 = \tilde{\mathbf{V}}\mathbf{x}$ and $\mathbf{x}_5 = \tilde{\mathbf{V}}^*\mathbf{x}_4$, $4nNK$ is for $\mathbf{x}_3 = \mathbf{T}\mathbf{x}_2$, $12nNK$ is for $\mathbf{V}^H T\mathbf{V}\mathbf{x}$ in the traditional FMM.

For large scale problems, (50) will be approximated by

$$\xi \approx \frac{2\rho + 1}{3} \tag{51}$$

Similarly, the memory saved can be evaluated by

$$\zeta = \frac{(4\rho + 1)NK}{5nNK} = \frac{4\rho + 1}{5} \tag{52}$$

The translation matrix $\mathbf{T}$ is highly sparse and can be further sparsified through the use of a windowed translation operator [20], then the weight for $\mathbf{T}$ is reduced, and the value of ξ and ζ will approach to ρ .

III. NUMERICAL RESULTS

To validate the analysis presented in the previous sections, numerical simulation for different shaped objects is considered. By conjugate-gradient solver with the same residual error (1e-5), we will test the threshold defined in equation (49) and the sparsity (defined as the percentage content of nonzero elements) for different shaped objects. Meanwhile, the accuracy of the proposed method will be verified by comparing with the analytical solution or the result of traditional FMM method. Finally, to show the efficiency of the proposed method, the CPU time and memory consumed for far field computation are given in Table III.

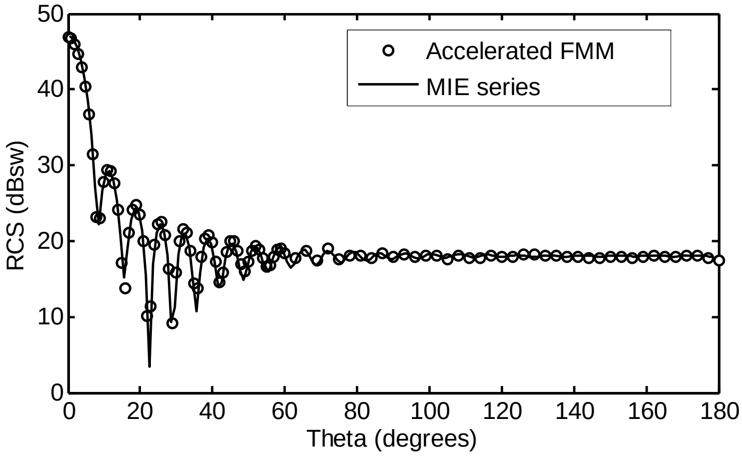


Fig. 4 The E-plane bistatic RCS of a PEC sphere with a diameter of 5λ

As the first example, a PEC sphere with a diameter of 9λ, which is illuminated by a plane wave propagating in the *z* direction and E-polarized in the *x* direction, is considered. The total number of unknowns is 34680 and the unknowns are divided into 194 groups. When the sparsity of disaggregation matrix and aggregation matrix is 33.17% ($\tau = 0.9$, $K = 288$), the bistatic radar cross section (RCS) of the sphere computed by the FMM accelerated by lifting scheme is compared with that of the analytical solution by MIE series and shown in Fig.4, from which we can conclude that the proposed method can obtain an accurate solution with sparse disaggregation matrix and aggregation matrix.

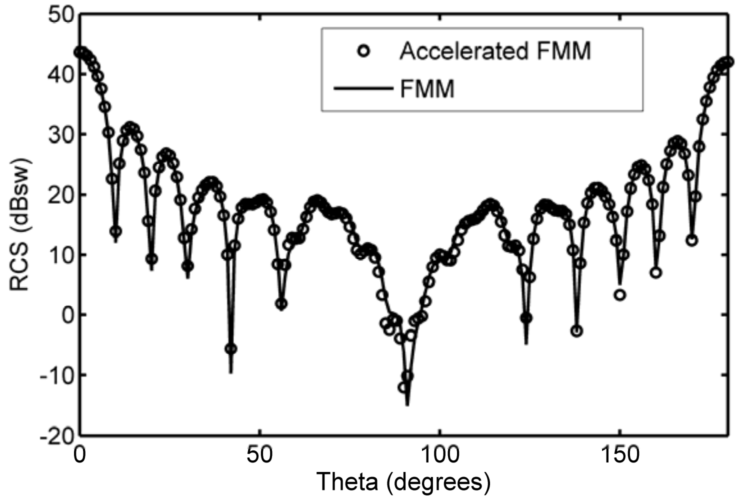
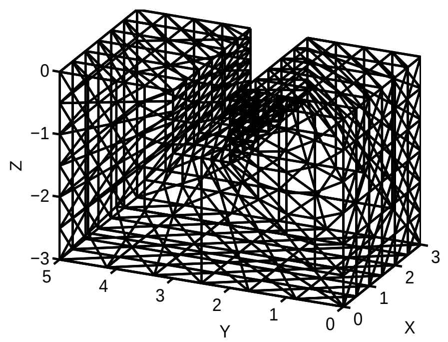
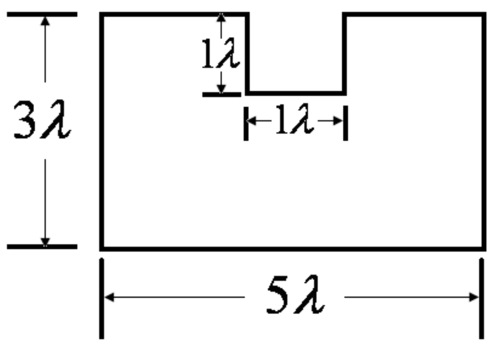


Fig. 5 The E-plane bistatic RCS of a PEC cube with a side length of 6λ

A PEC cube with a side length of 6λ is considered as the second example. The object is illuminated by a plane wave propagating in the *z* direction and E-polarized in the *x* direction, and the surface of it is discretized into 24300 triangular elements which results into 36450 unknowns. The sparsity of disaggregation matrix and aggregation matrix obtained is 30.97% ($\tau = 1.1$, $K = 288$), the accelerated FMM method obtained the accurate result more quickly as compared with the traditional FMM as shown in Fig.5.

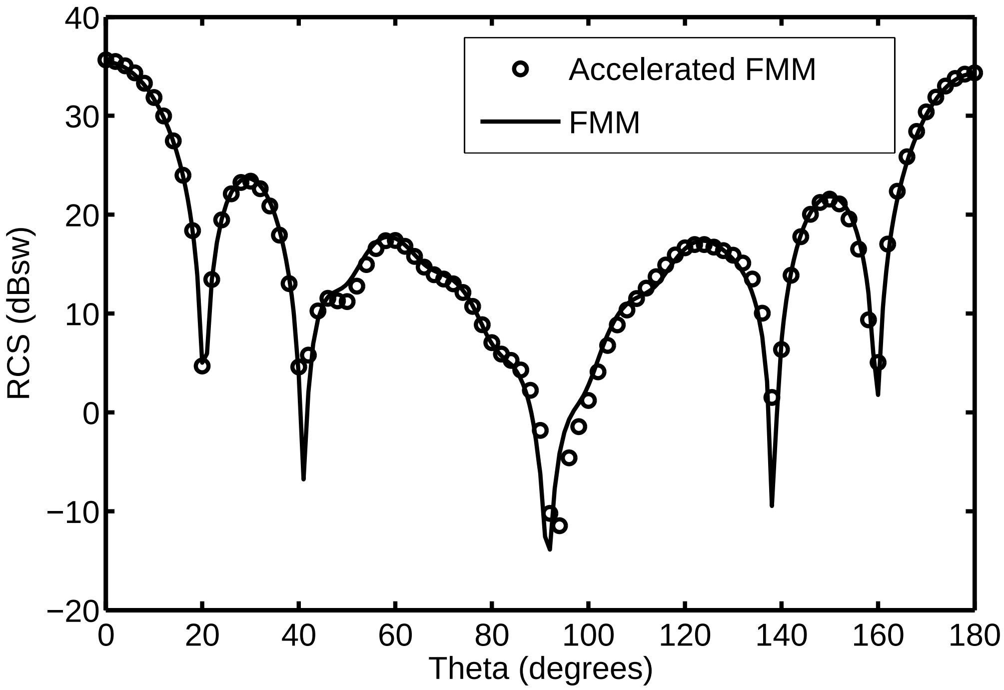




(a) Triangular elements discretization of the object **(b)** Cross section size description of the object

Fig.6 The geometrical description of a cuboid with a slot

The PEC cuboid with a slot shown in Fig.6, which is illuminated by a plane wave propagating in the *z* direction and E-polarized in the *x* direction, is considered with total unknowns of 19800. With the lifting wavelet-like transform is applied to FMM, the sparsity of disaggregation matrix and aggregation matrix obtained is 33.8% ($\tau = 0.9$, $K = 200$). As shown in Fig.7, under such sparsity, the RCS of the object computed by the accelerated FMM scheme agrees well with that of the traditional FMM.



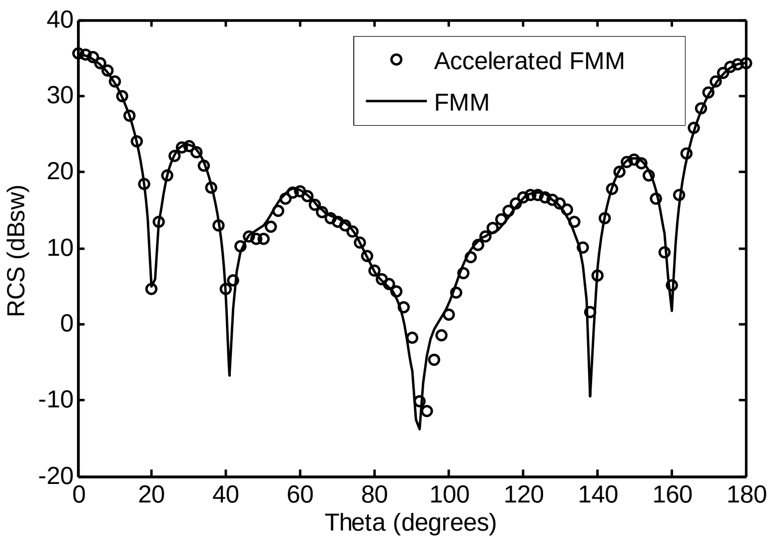


Fig. 7 The E-plane bistatic RCS of the PEC cuboid with a slot

The total CPU time and memory consumed for the MVM in far-field computation is shown in Table III. We can conclude that the proposed method can speed up the far-field computation by a factor of two with half memory consumed.

TABLE III The CPU time and memory required for the MVM operations of far-field interactions

Example	CPU time for far field computation		Memory required for Far-field computation	
	FMM	Accelerated FMM	FMM	Accelerated FMM
Fig. 4	507 seconds	283 seconds	536MB	249MB
Fig. 5	528 seconds	278 seconds	601MB	271MB
Fig. 7	334 seconds	179 seconds	158 MB	70 MB

IV. CONCLUSION

The Lifting wavelet-like transform is applied to the fast multipole method to calculate the radar cross section of different shaped three-dimensional PEC objects. The implementation of the lifting wavelet-like transform is described in detail, and a threshold technique is proposed to form a in-time compressing method. By numerical simulation and complexity analysis, we can conclude that with the help of proper LWLT scheme, the CPU time and memory consumed by the MVM in FMM are reduced by a factor of two.

ACKNOWLEDGMENT

This work is supported by Anhui Provincial Natural Science Foundation under grant number 090412047 and the Natural Science Foundation of the Anhui Higher Education Institution of China under grant number KJ2008A036& KJ2009A62, and partially by the National Natural Science Foundation of China (No.60671051).

REFERENCES

[1] Miller, E.K., Medgysi-Mitschang, L., and Newman, E.H.: 'Computational Electromagnetics: Frequency Domain Method of Moments', (IEEE Press, New York, 1992)

[2] Coifman, R., and Rokhlin, V.: 'The fast multipole method for the wave equation: a pedestrian prescription', *IEEE Antennas Propag. Mag.*, 1993, **35**, pp.7-12

[3] Wang, S., Su, Y., and Guan, X.: 'Applying fast multipole method to higher-order hierarchical Legendre basis functions in electromagnetic scattering', *IET Microwaves Antennas Propag.*, 2008, **2**, pp. 6-9

[4] Bleszynski, E., Bleszynski, M., and Jaroszewicz, T.: 'AIM: adaptive integral method for solving large-scale electromagnetic scattering and radiation problems', *Radio Sci.*, 1996, **31**, pp.1225-1251

[5] Sarkar, T. K., Arvas, E., and Rao, S. M.: 'Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies', *IEEE Trans. Antennas Propag.*, 1986, **34**, pp.635-640

[6] Song, J. M., and Chew, W. C.: 'Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering', *Microwave Opt. Technol. Lett.*, 1995, **10**, pp.14-19

[7] Garciá, E., Delgado, C., Diego, I. G., and Cátedra, M. F.: 'An iterative solution for electrically large problems combining the characteristic basis function method and the multilevel fast multipole algorithm', *IEEE Trans. Antennas Propag.*, 2008, **56**, pp. 2363-2371

[8] Steinberg, Z. B., and Leviatan, Y.: 'On the use of wavelet expansions in the method of moments'. *IEEE Trans. Antennas Propag.*, 1993，**41**, pp.610–619

[9] Wagner, R.L., and Chew, W.C.: 'A study of wavelets for the solution of electromagnetic

integral-equations’, *IEEE Trans. Antennas Propag.*, 1995, **43**, pp.802–810

[10] Ravnik, J., Škerget, L., and Zunic, Z.: ‘Comparison between wavelet and fast multipole data sparse approximations for Poisson and kinematics boundary-domain integral equations’, *Comput. Methods Appl. Mech. Eng.*, 2009, **198**, pp.1473-1485

[11] Wang, G.: ‘On the utilization of periodic wavelet expansions in the moment methods’, I*EEE Trans. Microwave Theory Tech.*, 1995, **43**, pp.2495–2498

[12] Barmada, S., and Raugi, M.: ‘Analysis of scattering problems by MOM with intervallic wavelets and operators’, *Appl. Comput. Electromagn. Soc. J.*, 2003, **18**, pp.62–67

[13] Sarkar, T. K., Wicks, M. C., and Salazar-Palma M.: ‘Wavelet Applications in Engineering Electromagnetics’, (Artech House, London, 2002)

[14] Huybrechs, D., and Vandewalle, S.: ‘A two-dimensional wavelet-packet transform for matrix compression of integral equations with highly oscillatory kernel’, *J. Comput. Appl. Math.*, 2006, **197**, pp.218-232

[15] Rodriguez, J. L., Taboada, J. M., Araujo, M. G. etc.: ‘On the use of the singular value decomposition in the fast multipole method’, *IEEE Trans. Antennas Propag*., 2008, 56, pp. 2325-2334

[16] Cheng, H., Gimbutas, Z., Martinsson, P.G., etc.: ‘On the compression of low rank matrices’, *SIAM J. Sci. Comput.*, 2005, **26**, pp. 1389-1404

[17] Martinsson, P. G., and Rokhlin, V.: ‘An accelerated kernel-independent fast multipole method in one dimension’, *SIAM J. Sci. Comput.*, 2007, 29, pp.1160-1178

[18] Daubechies, I., and Sweldens, W.: ‘Factoring wavelet transforms into lifting steps’, *J. Fourier Anal. Appl.*, 1998, **4**, pp.247-269

[19] Chew, W. C., Jin, J. M., Michielssen, E., and Song, J.M.: ‘Fast and Efficient Algorithms in Computational Electromagnetics’, (Artech House, Boston, 2001)

[20] Wagner, R. L., and Chew, W. C.: ‘A ray-propagation fast multipole algorithm’, *Microwave*

Opt. Technol. Lett., 1994, **7**, pp.435-438