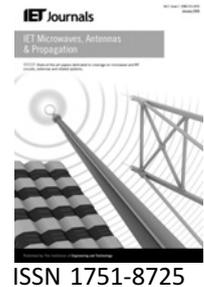


Published in IET Microwaves, Antennas & Propagation  
 Received on 11th August 2009  
 Revised on 30th March 2010  
 doi: 10.1049/iet-map.2009.0417



# Fast computation of radar cross-section by fast multipole method in conjunction with lifting wavelet-like transform

M.S. Chen<sup>1</sup> W.E.I. Sha<sup>2</sup> Q. Wu<sup>3</sup> Z.X. Huang<sup>3</sup> X.L. Wu<sup>1</sup>

<sup>1</sup>Department of Physics and Electronic Engineering, Hefei Normal University, 1688 Lianhua Road, Hefei 230601, People's Republic of China

<sup>2</sup>Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, People's Republic of China

<sup>3</sup>School of Electronic Science and Technology, Anhui University, Hefei 230039, People's Republic of China  
 E-mail: chenms@ustc.edu.cn

**Abstract:** The fast multipole method (FMM) in conjunction with the lifting wavelet-like transform scheme is proposed for the scattering analysis of differently shaped three-dimensional perfectly electrical conducting objects. As a flexible and efficient matrix compression technique, the proposed method can sparsify the aggregation matrix and disaggregation matrix in real time with compression ratio about 30%. The computational complexity and choice of proper wavelet are also discussed. Numerical simulation and complexity analysis have shown that the proposed method can speed up the aggregation and disaggregation steps of the FMM with lower memory requirements.

## 1 Introduction

Electromagnetic scattering problems of arbitrarily shaped three-dimensional (3-D) objects can be dealt with by the method of moments (MOM), which has been widely used and extensively studied over the past decades [1]. For the MOM, an  $N \times N$  dense impedance matrix equation is to be generated when the surface or volume equivalent currents of the object is approximated by  $N$  basis functions. In particular, the solution of this matrix equation by iterative methods requires the computational complexity of  $O(N^2)$ .

A variety of algorithms have been proposed to reduce the complexity of matrix-vector multiplication (MVM) in iterative methods. These algorithms include the fast multipole method (FMM) [2, 3], the adaptive integral method (AIM) [4] and the conjugate gradient (CG) fast Fourier transform algorithm [5]. The FMM and its multilevel version [6, 7] reduce the complexity of MVM to  $O(N^{1.5})$  and  $O(N \log N)$ , respectively.

Except for the above algorithms, the wavelet transform method [8–10] is also applied to the solution of integral equations as an efficient tool. Generally, wavelets have been applied to MOM in two ways. One is to be used directly as basis functions and test functions [11, 12] and the other is the discrete wavelet transform, which is applied to the impedance matrix to obtain a sparse matrix equation in the wavelet-domain [13, 14]. The applications of wavelet matrix transform have been used widely during the past decades. Unfortunately, they are mainly confined to the analysis of two-dimensional (2-D) problems or to special structures such as wires in which the current direction is 1-D.

Although aware of the wide application of FMM in computational electromagnetics, recently many researchers have proposed new techniques to further reduce its computational complexity. Based on singular value decomposition, a new matrix compression technique [15] sparsifies the aggregation matrix of the FMM. By the matrix compression technique proposed in [16],

Martinsson and Rokhlin [17] successfully accelerated the kernel-independent FMM in one dimension.

In this paper, we apply the lifting wavelet-like transform (LWLT) [18] to FMM for sparsifying the aggregation and disaggregation matrices in real time. Numerical results for differently shaped three-dimensional objects are considered and the relevant computational complexity analysis is also presented. Compared with traditional FMM, the introduction of the LWLT can further accelerate the MVM for far-field computation by factor of two and saves considerable memory when proper wavelets are selected.

## 2 Theory

### 2.1 Fast multipole method

For 3-D arbitrarily shaped PEC objects illuminated by an incident field  $E^i(r)$ , the electric field integral equation is given by

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

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

Using MOM, it yields a matrix equation of the form

$$[Z_{mn}][x_n] = [F_m] \quad (2)$$

where  $x_n$  is the unknown current coefficients and

$$Z_{mn} = \int_S dSt_m(r) \times \int_S dS' \bar{\mathbf{G}}(r, r') j_n(r') \quad (3)$$

$$F_m = \frac{1}{jk\eta} \int_S dSt_m(r) E^i(r) \quad (4)$$

To speed up the solution of (2), FMM is employed [19], which decomposes the dense matrix  $[Z_{mn}]$  of (2) as

$$\mathbf{Z} = \mathbf{Z}^{\text{near}} + \mathbf{V}^H \mathbf{TV} \quad (5)$$

where  $\mathbf{Z}^{\text{near}}$  represents nearby interactions and  $\mathbf{V}^H \mathbf{TV}$  represents the far-field interactions with the mathematical expressions as

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

$$\mathbf{T} = \begin{pmatrix} T_1 & & & \\ & T_2 & & \\ & & \ddots & \\ & & & T_K \end{pmatrix} \quad (7)$$

$$\mathbf{V}_{pn} = \int_{S'} (\bar{\mathbf{I}} - \hat{\mathbf{k}}_p \hat{\mathbf{k}}_p) j_n(r') e^{jk_p(r'-r'_o)} dS' \quad (8)$$

$$T_p = \frac{k^2 \eta}{16\pi^2} \omega_p \sum_{l=0}^L (-j)^l (2l+1) b_l^{(2)}(kX) P_l(\hat{\mathbf{k}}_p \cdot \hat{\mathbf{X}}), \quad (9)$$

$$p = 1, 2, \dots, 2L^2$$

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

### 2.2 Accelerate FMM by LWLT

In this section, we will introduce the LWLT to sparsify the aggregation and disaggregation matrices  $\mathbf{V}^H$  and  $\mathbf{V}$ .

Considering the orthogonal properties of wavelet transform matrices

$$\mathbf{W} \tilde{\mathbf{W}} = \tilde{\mathbf{W}} \mathbf{W} = \mathbf{I} \quad (10)$$

the far-field interactions can be rewritten as

$$\mathbf{V}^H \mathbf{TV} \mathbf{x} = \mathbf{V}^H \mathbf{W} \tilde{\mathbf{W}} \mathbf{TW} \tilde{\mathbf{W}} \mathbf{V} \mathbf{x} \quad (11)$$

Letting  $\tilde{\mathbf{V}}^* = \mathbf{V}^H \mathbf{W}$  and  $\tilde{\mathbf{V}} = \tilde{\mathbf{W}} \mathbf{V}$ , one gets

$$\mathbf{V}^H \mathbf{TV} \mathbf{x} = \tilde{\mathbf{V}}^* \tilde{\mathbf{W}} \mathbf{TW} \tilde{\mathbf{V}} \mathbf{x} \quad (12)$$

Hence the MVM can be implemented by the following steps:

- First, the wavelet matrix transform is applied to  $\mathbf{V}^H$  and  $\mathbf{V}$  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 sparsified by the threshold  $\sigma_m$ .
- Second, the aggregation step is implemented by  $\mathbf{x}_1 = \tilde{\mathbf{V}} \mathbf{x}$  followed by the inverse wavelet transform  $\mathbf{x}_2 = \mathbf{W} \mathbf{x}_1$ .
- Third, the translation step is  $\mathbf{x}_3 = \mathbf{T} \mathbf{x}_2$ .
- Finally, the forward wavelet transform is implemented by  $\mathbf{x}_4 = \tilde{\mathbf{W}} \mathbf{x}_3$  followed by the disaggregation step  $\mathbf{x}_5 = \tilde{\mathbf{V}}^* \mathbf{x}_4$ .

To reduce the CPU time and memory for the above wavelet transform, the lifting scheme is introduced. In LWLT, the wavelet transform is implemented according to

factorisation of the polyphase matrices

$$\tilde{\mathbf{P}}(z^{-1})^t = \prod_{i=1}^m \begin{pmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{pmatrix} \begin{pmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & 1/F \end{pmatrix} \quad (13)$$

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

where  $s_i(z)$  and  $t_i(z)$  are the Laurent polynomials and  $F$  is a non-zero constant.

Without the transform matrices  $\tilde{\mathbf{W}}$  and  $\mathbf{W}$ , the wavelet transform can be operated by the split, predict and update steps in the lifting scheme.

*Split:* the original data  $(x[n])$  is split into the odd subsets  $(x_o[n])$  and the even subsets  $(x_e[n])$

$$\begin{cases} x_o[n] = x[2n - 1] \\ x_e[n] = x[2n] \end{cases} \quad (15)$$

*Predict:* high-frequency (fluctuation) component  $d[n]$  is obtained by predicting odd subsets from even subsets with prediction operator  $Q$

$$d[n] = x_o[n] - Q(x_e[n]) \quad (16)$$

*Update:* low-frequency (smooth) component  $c[n]$  is obtained by applying an update operator  $U$  to the high-frequency component and being added to even subsets

$$c[n] = x_e[n] + U(d[n]) \quad (17)$$

The operators  $Q$  and  $U$  above can be deduced from the polyphase matrices described in (13) and (14). The forward and inverse wavelet transforms are operated by  $\tilde{\mathbf{P}}(z^{-1})^t$  and  $\mathbf{P}(z)$ , respectively. Specific examples can be found in [18].

We take the Haar wavelet as an example to give a brief interpretation of the numerical implementations of lifting scheme. For the Haar wavelet, the polyphase matrix  $\tilde{\mathbf{P}}(z^{-1})^t$  is given by

$$\tilde{\mathbf{P}}(z^{-1})^t = \begin{pmatrix} \sqrt{2} & 0 \\ 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{2\sqrt{2}} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (18)$$

As described above, when the lifting scheme is applied, the signal (the row or column in the matrix) will undergo three

basic operations as given below

$$x_o(l) = x(2l + 1) \quad (19)$$

$$x_e(l) = x(2l) \quad (20)$$

$$x_o^{(1)}(l) = x_o(l) - x_e(l) \quad (21)$$

$$x_e^{(1)}(l) = x_e(l) + \frac{1}{2\sqrt{2}} x_o^{(1)}(l) \quad (22)$$

$$\tilde{x}_e(l) = \sqrt{2} x_e^{(1)}; \tilde{x}_o(l) = \frac{\sqrt{2}}{2} x_o^{(1)} \quad (23)$$

where (23) is used for guaranteeing the orthonormal property of the LWLT. After one level implementation above, the fluctuation term  $\tilde{x}_o$ , 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 smooth term  $\tilde{x}_e$ , which can be updated by the fluctuation term and the even component, will contain the major information of  $\tilde{\mathbf{V}}$  (or  $\tilde{\mathbf{V}}^*$ ). The fluctuation information of the matrices will lead to fluctuation of the currents. In particular, the noise-like fluctuation of currents is expected to be very small and random, which will make little contribution for far-field scattering analysis and therefore can be neglected. Mathematically, most small elements in  $\tilde{x}_o$  can be set to be zero. For the following levels, the same operations will be applied to  $\tilde{x}_e$  and a sparser matrix will be obtained.

As can be seen from the above description, two auxiliary LWLTs for sparsifying the aggregation matrix  $\tilde{\mathbf{V}}$  and the disaggregation matrix  $\tilde{\mathbf{V}}^*$  are added to the MVM. The computational complexity of the LWLT will be analysed in the following sections.

### 2.3 Choice of proper wavelets

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

$$\begin{aligned} \tilde{\mathbf{P}}(z^{-1})^t &= \begin{bmatrix} \frac{\sqrt{3}+1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z & 1 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3}-2}{4} z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sqrt{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (24) \end{aligned}$$

The above can be operated by the following steps

$$x_e(l) = x(2l); \quad x_o(l) = x(2l + 1) \quad (25)$$

$$x_e^{(1)}(l) = x_e(l); \quad x_o^{(1)}(l) = x_o(l) - \sqrt{3}x_e(l) \quad (26)$$

$$x_e^{(2)}(l) = x_e^{(1)}(l) + \frac{\sqrt{3}}{4}x_o^{(1)}(l) + \frac{\sqrt{3}-2}{4}x_o^{(1)}(l+1); \quad (27)$$

$$x_o^{(2)}(l) = x_o^{(1)}(l)$$

$$x_e^{(3)}(l) = x_e^{(2)}(l); \quad x_o^{(3)}(l) = x_o^{(2)}(l) + x_e^{(2)}(l-1) \quad (28)$$

$$x_e^{(4)}(l) = \frac{\sqrt{3}+1}{\sqrt{2}}x_e^{(3)}(l); \quad x_o^{(4)}(l) = \frac{\sqrt{3}-1}{\sqrt{2}}x_o^{(3)}(l) \quad (29)$$

$$\tilde{x}_e = x_e^{(4)}(l); \quad \tilde{x}_o = x_o^{(4)}(l) \quad (30)$$

The inverse LWLT corresponding to  $P(z)$  is omitted here for the sake of brevity. Similarly, the implementation of the forward LWLT for the Daubechies wavelet with fourth-order vanishing moments (db4) is given by

$$x_e(l) = x(2l); \quad x_o(l) = x(2l + 1) \quad (31)$$

$$x_e^{(1)}(l) = x_e(l); \quad x_o^{(1)}(l) = x_o(l) + \alpha^{(1)}(1)x_e(l+1) \quad (32)$$

$$x_e^{(2)}(l) = x_e^{(1)}(l) + \beta^{(1)}(1)x_o^{(1)}(l) + \beta^{(1)}(2)x_o^{(1)}(l-1); \quad (33)$$

$$x_o^{(2)}(l) = x_o^{(1)}(l)$$

$$x_e^{(3)}(l) = x_e^{(2)}(l);$$

$$x_o^{(3)}(l) = x_o^{(2)}(l) + \alpha^{(2)}(2)x_e^{(2)}(l+1) + \alpha^{(2)}(1)x_e^{(2)}(l+2) \quad (34)$$

$$x_e^{(4)}(l) = x_e^{(3)}(l) + \beta^{(2)}(1)x_o^{(3)}(l) + \beta^{(2)}(2)x_o^{(3)}(l-1); \quad (35)$$

$$x_o^{(4)}(l) = x_o^{(3)}(l)$$

$$x_e^{(5)}(l) = x_e^{(4)}(l);$$

$$x_o^{(5)}(l) = x_o^{(4)}(l) + \alpha^{(3)}(3)x_e^{(4)}(l-2) \quad (36)$$

$$+ \alpha^{(3)}(2)x_e^{(4)}(l-1) + \alpha^{(3)}(1)x_e^{(4)}(l)$$

$$\tilde{x}_e = \beta^{(3)}(1)x_e^{(5)}(l); \quad \tilde{x}_o = \alpha^{(4)}(1)x_o^{(5)}(l) \quad (37)$$

The coefficients above can be obtained from Matlab 7.0 by the command function 'liftwave()', and are listed in Table 1.

Finally, one can implement the forward LWLT for the Daubechies wavelets with eighth-order vanishing moments (db8) by the following steps

$$x_e(l) = x(2l); \quad x_o(l) = x(2l + 1) \quad (38)$$

$$x_e^{(1)}(l) = x_e(l); \quad x_o^{(1)}(l) = x_o(l) + \alpha^{(1)}(1)x_e(l) \quad (39)$$

$$x_e^{(2)}(l) = x_e^{(1)}(l) + \beta^{(1)}(1)x_o^{(1)}(l+1) + \beta^{(1)}(2)x_o^{(1)}(l); \quad (40)$$

$$x_o^{(2)}(l) = x_o^{(1)}(l)$$

$$x_e^{(3)}(l) = x_e^{(2)}(l);$$

$$x_o^{(3)}(l) = x_o^{(2)}(l) + \alpha^{(2)}(2)x_e^{(2)}(l-2) + \alpha^{(2)}(1)x_e^{(2)}(l-1) \quad (41)$$

$$x_e^{(4)}(l) = x_e^{(3)}(l) + \beta^{(2)}(1)x_o^{(3)}(l+3) + \beta^{(2)}(2)x_o^{(3)}(l+2);$$

$$x_o^{(4)}(l) = x_o^{(3)}(l) \quad (42)$$

$$x_e^{(5)}(l) = x_e^{(4)}(l);$$

$$x_o^{(5)}(l) = x_o^{(4)}(l) + \alpha^{(3)}(2)x_e^{(4)}(l-4) + \alpha^{(3)}(1)x_e^{(4)}(l-3) \quad (43)$$

$$x_e^{(6)}(l) = x_e^{(5)}(l) + \beta^{(3)}(1)x_o^{(5)}(l+5) + \beta^{(3)}(2)x_o^{(5)}(l+4);$$

$$x_o^{(6)}(l) = x_o^{(5)}(l) \quad (44)$$

$$x_e^{(7)}(l) = x_e^{(6)}(l);$$

$$x_o^{(7)}(l) = x_o^{(6)}(l) + \alpha^{(4)}(2)x_e^{(6)}(l-4) + \alpha^{(4)}(1)x_e^{(6)}(l-3) \quad (45)$$

$$x_e^{(8)}(l) = x_e^{(7)}(l) + \beta^{(4)}(1)x_o^{(7)}(l+5) + \beta^{(4)}(2)x_o^{(7)}(l+4);$$

$$x_o^{(8)}(l) = x_o^{(7)}(l) \quad (46)$$

$$x_e^{(9)}(l) = x_e^{(8)}(l); \quad (47)$$

$$x_o^{(9)}(l) = x_o^{(8)}(l) + \alpha^{(5)}(3)x_e^{(8)}(l-7) + \alpha^{(5)}(2)x_e^{(8)}(l-6) + \alpha^{(5)}(1)x_e^{(8)}(l-5)$$

$$\tilde{x}_e(l) = \beta^{(5)}(1)x_e^{(9)}(l); \quad \tilde{x}_o(l) = \alpha^{(6)}(1)x_o^{(9)}(l) \quad (48)$$

The lifting coefficients for db8 are listed in Table 2.

The number of multiplication operations for the lifting scheme is set to be  $g$ , which can be counted from the

**Table 1** Coefficients in lifting scheme for db4 wavelet

Predict coefficients				Update coefficients		
$\alpha^{(1)}$	-0.3223			$\beta^{(1)}$	-1.1171	-0.3001
$\alpha^{(2)}$	-0.0188	0.1176		$\beta^{(2)}$	2.1318	0.6364
$\alpha^{(3)}$	-0.4691	0.1400	-0.0248	$\beta^{(3)}$	0.7341	
$\alpha^{(4)}$	-1.3622					

**Table 2** Coefficients in lifting scheme for db8 wavelet

Predict coefficients				Update coefficients		
$\alpha^{(1)}$	-5.7496			$\beta^{(1)}$	-0.0523	0.1688
$\alpha^{(2)}$	14.5428	-7.4021		$\beta^{(2)}$	-0.0324	0.0609
$\alpha^{(3)}$	5.8187	-2.7557		$\beta^{(3)}$	0.9453	0.2420
$\alpha^{(4)}$	0.0002	-0.0018		$\beta^{(4)}$	-0.9526	-0.2241
$\alpha^{(5)}$	1.0497	-0.2470	0.0272	$\beta^{(5)}$	3.5494	
$\alpha^{(6)}$	0.2817					

polyphase matrices or from Tables 1 and 2. If the length of the signal is  $K$ , the total computational complexity for implementing the lifting scheme is

$$q \times \frac{K}{2} \times \left(1 + \frac{1}{2} + \frac{1}{4} + \dots\right) = qK \quad (49)$$

For the Haar, db2, db4 and db8 wavelets, the values of  $q$  are 3, 5, 12 and 20, respectively.

To select proper wavelets tailored to FMM, take a PEC sphere with the diameter of  $5\lambda$  ( $L = 12$ ) as an example. The Haar and Daubechies wavelets with different vanishing moments are employed and compared. After the LWLT of the Haar, db2, db4 and db8, a column of  $\tilde{V}$  is presented in Fig. 1.

As seen in Fig. 1, and considering the transform complexity and sparsity, db4 is a better choice.

For a field group with  $M_i$  Rao–Wilton–Glisson (RWG) functions and a source group with  $N_j$  RWG functions, there are  $2L^2$  elements in each column of  $V$  and the LWLT is applied to  $V$  column-by-column. Take the matrix  $V$  for example, the LWLT is implemented for a column of  $V$ , then the clipping operation is used with the threshold and only the large elements in the column are stored. So, the proposed method is an in-space and real-time compression technique. The threshold for the  $m$ th column is defined by

$$\sigma_m = \tau \frac{1}{K} \sum_{p=1}^K (|\tilde{V}_\theta(p, m)|^2 + |\tilde{V}_\varphi(p, m)|^2)^{1/2} \quad (50)$$

with  $\tau \in [0.8, 1.2]$  by the numerical simulation.

As can be seen from Fig. 1c, most of the elements are far smaller than the others. The result after the clipping operation ( $\tau = 0.9$ ) is shown in Fig. 2 and only about 30% of the total elements are non-zeros. Next, the inverse LWLT is implemented and the recovered column is given in Fig. 3 which agrees well with the original column in  $V$ .

## 2.4 Computational complexity of MVM for far-field interactions

As mentioned above, after obtaining two sparse matrices  $\tilde{V}$  and  $\tilde{V}^*$ , two additional LWLTs are executed. To analyse the total complexity, we take the CG method for example, which needs two MVMs  $[Z_{mn}][\alpha_n]$  and  $[Z_{mn}]^H[\alpha_n]$  for each iteration. For a problem with  $N$  unknowns and  $\sqrt{N}$  total groups, taking into the operations for the translation step, the effectiveness can be evaluated by

$$\xi = \frac{4qNK + 8nq\sqrt{N}K + 8\rho nNK + 4nNK}{12nNK} = \frac{q}{3n} + \frac{2q}{3\sqrt{N}} + \frac{2\rho + 1}{3} \quad (51)$$

where  $\rho$  is the sparsity of  $\tilde{V}$  and  $\tilde{V}^*$ ,  $n$  is the iteration number and  $K = 2L^2$ . For the traditional FMM,  $12nNK$  operation is required. For the proposed scheme,  $4qNK$  operation is required for  $V_\theta, V_\varphi, V_\theta^H$  and  $V_\varphi^H$ ,  $8nq\sqrt{N}K$  is for  $x_2 = \tilde{W}x_1$  and  $x_4 = \tilde{W}x_3$ ,  $8\rho nNK$  is for  $x_1 = \tilde{V}x$  and  $x_5 = \tilde{V}^*x_4$  and  $4nNK$  is for  $x_3 = Tx_2$ .

For the large-scale problems, (51) can be approximated by

$$\xi \simeq \frac{2\rho + 1}{3} \quad (52)$$

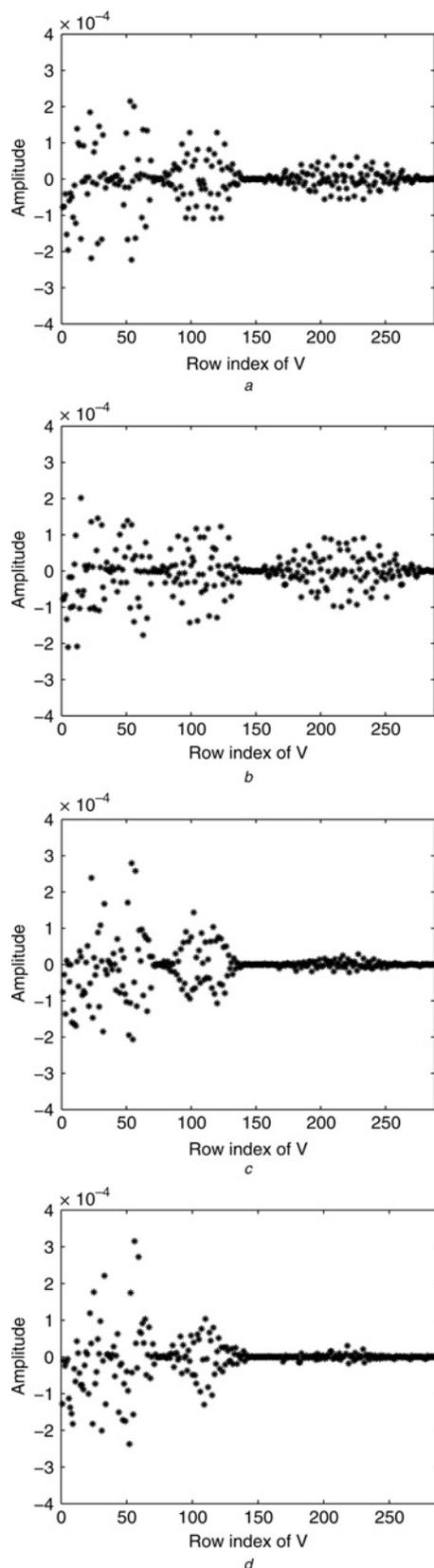
Similarly, the ratio of memory saved can be evaluated by

$$\zeta = \frac{(4\rho + 1)NK}{5nNK} = \frac{4\rho + 1}{5} \quad (53)$$

The sparse translation matrix  $T$  can be further sparsified through the utilisation of a windowed translation operator [20]. As a result, the weight for  $T$  will be reduced, and the values of  $\xi$  and  $\zeta$  will approach  $\rho$ .

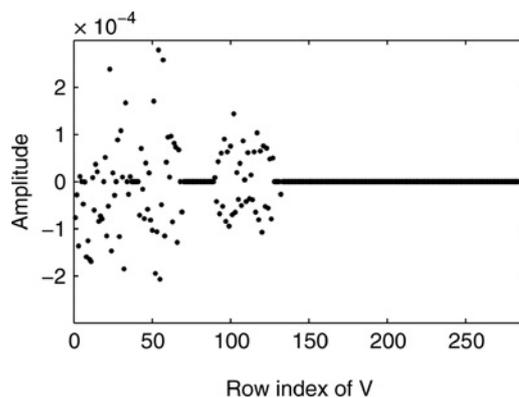
## 3 Numerical results

To validate the analysis presented in the previous sections, numerical simulation for different shaped objects is considered. Using the CG solver with the same residual error ( $1 \times 10^{-5}$ ), we test the threshold defined in (50) and

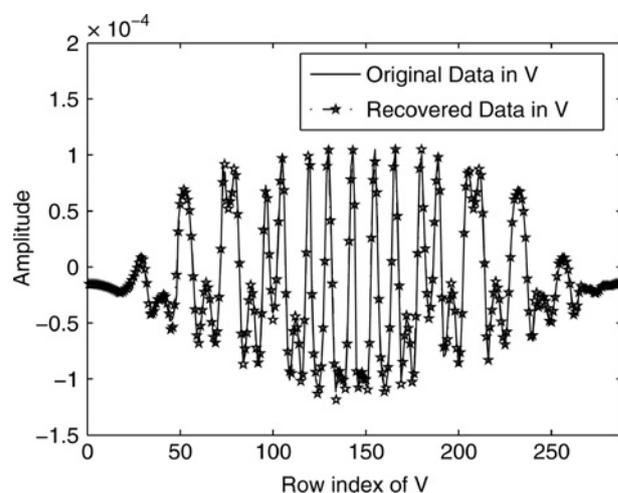


**Figure 1** Elements distributions in the disaggregation matrix after the forward LWLT

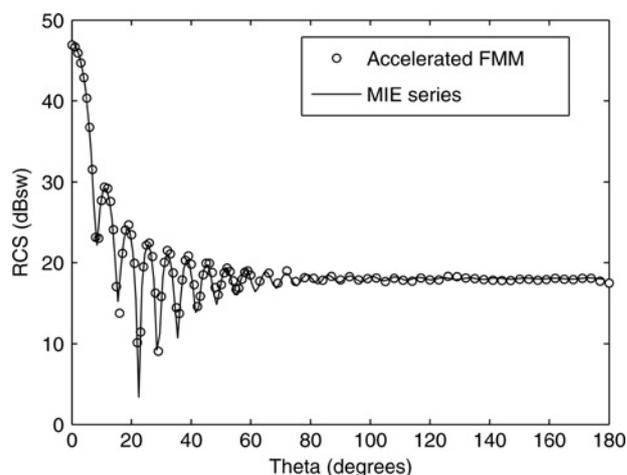
- a Elements distribution after two-level Haar LWLT
- b Elements distribution after two-level db2 LWLT
- c Elements distribution after two-level db4 LWLT
- d Elements distribution after two-level db8 LWLT



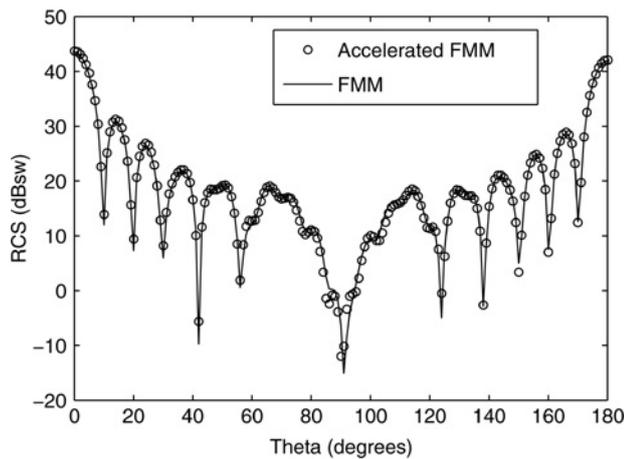
**Figure 2** Elements distribution in the aggregation matrix after clipping implementation



**Figure 3** Recovered elements distribution from the column presented in Fig. 2 by the inverse LWLT, which is compared with the original column in V



**Figure 4** E-plane bistatic RCS of a PEC sphere with diameter of  $9\lambda$



**Figure 5** *E*-plane bistatic RCS of a PEC cube with side length of  $6\lambda$

the sparsity (defined as the percentage content of non-zero elements) for differently shaped objects. Meanwhile, the accuracy of the proposed method will be verified by comparing to the analytical solution or the results of traditional FMM. Finally, to show the efficiency of the proposed method, the CPU time and memory consumed for far-field computation will be listed in a table.

As a first example, a PEC sphere with a diameter of  $9\lambda$  is considered, which is illuminated by a plane wave propagating

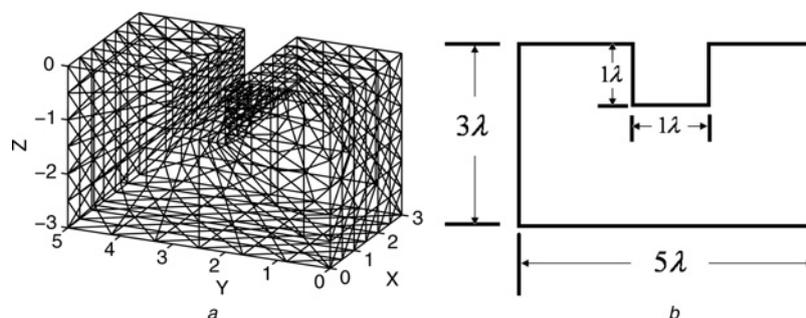
in the  $z$ -direction and  $E$ -polarised in the  $x$ -direction. The total number of unknowns is 34 680 and the unknowns are divided into 194 groups. When the sparsity of disaggregation matrix and aggregation matrix are 33.17% ( $\tau = 0.9$ ,  $K = 288$ ), the bistatic radar cross-section (RCS) of the sphere calculated by the FMM in conjugation with lifting scheme is compared to that of the analytical solution by Mie series. As shown in Fig. 4, we can see that the proposed method can obtain an accurate solution with the sparse disaggregation and aggregation matrices.

A PEC cube with side length of  $6\lambda$  is considered as the second example. The object is illuminated by a plane wave propagating in the  $z$ -direction and  $E$ -polarised in the  $x$ -direction, and its surface is discretised into 24 300 triangular elements and therefore 36 450 unknowns are generated. The sparsity of the disaggregation matrix and aggregation matrix obtained is 30.97% ( $\tau = 1.1$ ,  $K = 288$ ). As shown in Fig. 5 and Table 3, the accelerated FMM method can achieve accurate results with less CPU time compared to the traditional FMM.

As shown in Fig. 6, the PEC cuboid with a slot is illuminated by a plane wave propagating in the  $z$ -direction and  $E$ -polarised in the  $x$ -direction. The number of total unknowns is 19 800. By applying the LWLT to FMM, the sparsity of the disaggregation and aggregation matrices is 33.8% ( $\tau = 0.9$ ,  $K = 200$ ). As shown in Fig. 7, under such sparsity condition, the RCS of the object computed by the

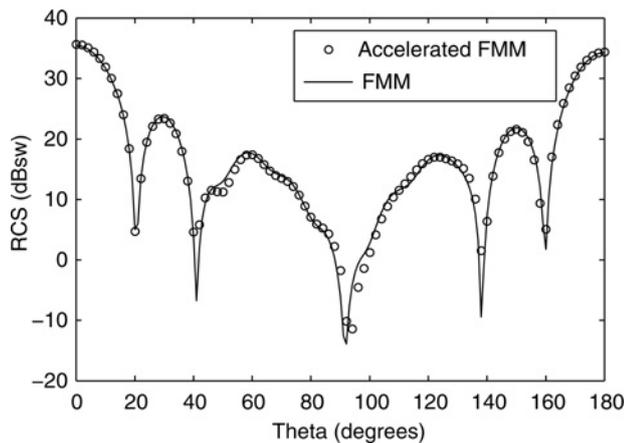
**Table 3** CPU time and memory consumed by the MVM for the far-field calculation

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



**Figure 6** Geometrical description of a cuboid with a slot

*a* Triangular elements discretisation of the object  
*b* Cross section of the object



**Figure 7** E-plane bistatic RCS of the PEC cuboid with a slot

accelerated FMM scheme agrees well with that of the traditional FMM.

The total CPU time and memory consumed by the MVM for the far-field calculation is listed in Table 3, from which we can conclude that the proposed method speeds up the far-field calculation by a factor of two with half of memory consumed.

## 4 Conclusion

The LWLT is applied to speed up the FMM for calculating the RCS of different shaped 3-D PEC objects. The theory, physical interpretation and numerical implementations of the LWLT are specified in this paper. According to the complexity–sparsity trade-off, the Daubechies wavelet with fourth-order vanishing moments is the proper choice to accelerate the FMM. By numerical simulation and complexity analysis, we can conclude that with the help of the lifting scheme, both the CPU time and memory of the MVM in the far-field calculation can be reduced by a factor of two.

## 5 Acknowledgments

The authors thank the anonymous reviewers for their useful comments and constructive suggestions. This work is supported by Science and Technological Fund of Anhui Province for Outstanding Youth under grant no. 10040606Y08, Anhui Provincial Natural Science Foundation under grant no. 090412047 and Science and Technology Planning Project of Anhui Province under grant no. 09020204029, and partially by the Key Program of National Natural Science Foundation of China under grant no. 60931002.

## 6 References

[1] MILLER E.K., MEDGYSI-MITSCHANG L., NEWMAN E.H.: 'Computational electromagnetics: frequency domain method of moments' (IEEE Press, New York, 1992)

[2] COIFMAN R., 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., GUAN X.: 'Applying fast multipole method to higher-order hierarchical Legendre basis functions in electromagnetic scattering', *IET Microw. Antennas Propag.*, 2008, **2**, pp. 6–9

[4] BLESZYNSKI E., BLESZYNSKI M., 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., 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., CHEW W.C.: 'Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering', *Microw. Opt. Technol. Lett.*, 1995, **10**, pp. 14–19

[7] GARCÍA E., DELGADO C., DIEGO I.G., 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., 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., 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., 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', *IEEE Trans. Microw. Theory Tech.*, 1995, **43**, pp. 2495–2498

[12] BARMADA S., 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., SALAZAR-PALMA M.: 'Wavelet applications in engineering electromagnetics' (Artech House, London, 2002)

- [14] HUYBRECHS D., 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., ET AL.: '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., ET AL.: 'On the compression of low rank matrices', *SIAM J. Sci. Comput.*, 2005, **26**, pp. 1389–1404
- [17] MARTINSSON P.G., ROKHLIN V.: 'An accelerated kernel-independent fast multipole method in one dimension', *SIAM J. Sci. Comput.*, 2007, **29**, pp. 1160–1178
- [18] DAUBECHIES I., 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., SONG J.M.: 'Fast and efficient algorithms in computational electromagnetics' (Artech House, Boston, 2001)
- [20] WAGNER R.L., CHEW W.C.: 'A ray-propagation fast multipole algorithm', *Microw. Opt. Technol. Lett.*, 1994, **7**, pp. 435–438