Published in IET Microwaves, Antennas & Propagation
Received on 29th April 2009
Revised on 10th December 2009
doi: 10.1049/iet-map.2009.0229

In Special Issue on Asia-Pacific Microwave Conference

ISSN 1751-8725

$\mathcal{H}$- and $\mathcal{H}^2$-matrix-based fast integral-equation solvers for large-scale electromagnetic analysis

W. Chai D. Jiao

School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA
E-mail: djiao@purdue.edu

Abstract: Integral-equation (IE)-based methods generally lead to dense systems of linear equations. The resulting matrices, although dense, can be thought of as 'data sparse', that is, they can be specified by few parameters. Based on this observation, two fast IE-based methods were developed for large-scale electromagnetic analysis. One is a centre-point $\mathcal{H}$-matrix-based method of linear complexity for large-scale analysis of static problems or problems having small electric sizes. The other is an $\mathcal{H}^2$-matrix-based method of controlled accuracy and linear complexity for large-scale analysis of electrodynamic problems across a wide range of electric sizes. Numerical simulations from a small number of unknowns to over 1 million unknowns, from small electric sizes to over 60 wavelengths have demonstrated the accuracy and efficiency of the proposed methods. The methods are kernel independent, and hence suitable for any IE-based formulation. In addition, they are applicable to arbitrarily shaped structures.

1 Introduction

Computational electromagnetic methods can be categorised into two classes: integral-equation (IE)-based methods and partial differential equation (PDE)-based ones. Compared to PDE-based methods, IE-based methods generally lead to dense systems of linear equations. When a direct method is used, the operation count is proportional to $O(N^3)$ and the memory requirement is proportional to $O(N^2)$, with N being the matrix size. When an iterative solver is used, the memory requirement remains the same, and the computing time is proportional to $O(N_{it}N^2)$, where N_{it} denotes the total number of iterations required to reach convergence. In recent years, fast multipole-based methods [1], fast fourier transform (FFT)-based methods [2, 3] and fast low-rank compression methods [4–6] have been developed that dramatically reduce the memory requirement of the iterative dense matrix solvers to $O(N \log N)$, and the CPU time to $O(N \log N)$ for electrodynamic problems. This represents an impressive improvement as compared with conventional $O(N^3)$ or $O(N^2)$ techniques.

However, the analysis and design of advanced engineering systems across the entire electromagnetic spectrum results in numerical problems that are of very large scale, requiring billions of parameters to describe them accurately. For example, accurate electromagnetics-guided full-chip optical proximity correction for the faithful reproduction of design onto the wafer at 45 nm technology node and beyond. As another example, the design of high-speed mixed-signal integrated circuits crosscutting electronics, photonics and micro-electromechanical systems (MEMS) technologies. Driven by the large problem size encountered in the analysis and design of advanced engineering systems, there is a continued demand of reducing the complexity of computational electromagnetic methods.

IE-based methods generally lead to dense systems of linear equations. The resulting matrices, although dense, can be thought of as 'data sparse', that is, they can be specified by few parameters. Based on this observation, in this work, we develop $\mathcal{H}$- and $\mathcal{H}^2$-matrix-based methods of linear complexity for large-scale electromagnetic analysis. The '$\mathcal{H}$ (hierarchical) matrix' is a general mathematical framework [7–9], which enables a highly compact representation and efficient numerical computation of dense matrices. To be specific, if matrix $\boldsymbol{C}$ is an $m \times n$ off-diagonal block in an $\mathcal{H}$

matrix which describes interactions on upper levels in the hierarchy, it can be written as $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}^{\mathrm{T}}$, where $\boldsymbol{A}$ is of dimension $m \times r$, $\boldsymbol{B}$ is of dimension $n \times r$ and r denotes the rank of $\boldsymbol{C}$ with $r < m$ and $r < n$. Storage requirements and matrix–vector multiplications using $\mathcal{H}$-matrices have been shown to be of complexity $O(N \log N)$, with N being the matrix size. From a mathematical point of view, existing low-rank compression-based IE methods [4–6] developed for electromagnetic analysis, although their technical details could be very different and many of them predated the literature of $\mathcal{H}$ matrices, can be viewed in the framework of $\mathcal{H}$ matrices. Recent advances in the $\mathcal{H}$-matrix mathematical framework suggested that a matrix–vector product can be performed in $O(N)$ operations by introducing $\mathcal{H}^2$ matrices [10–12]. The nested structure is the key difference between general $\mathcal{H}$ matrices and $\mathcal{H}^2$ matrices, since it permits an efficient reuse of information across the cluster tree. The accuracy and complexity analysis given in the literature of $\mathcal{H}$- and $\mathcal{H}^2$ matrices was generally conducted based on kernel functions that do not change with frequency. In addition, no linear complexity has been obtained for electrodynamic problems.

In [13], we demonstrated the feasibility of an $\mathcal{H}$-matrix-based representation for IE-based analysis of electrodynamic problems. In this work, we give a detailed error analysis of the $\mathcal{H}$-matrix-based representation of electrodynamic problems. We show that the $\mathcal{H}$-matrix-based representation is error bounded. In addition, we show that by developing a centre-point $\mathcal{H}$-matrix-based method, the time and memory complexity of solving the dense system matrix can be reduced to $O(N)$. This method can be viewed as an $\mathcal{H}^2$-matrix-based representation of rank one. Since rank one is generally not sufficient for accurate simulation of electrically large problems, we limit the use of the proposed centre-point $\mathcal{H}$-matrix-based method to the analysis of static problems or problems having small electric sizes. We apply the proposed method to the analysis of large-scale on-chip 3-D interconnects embedded in inhomogeneous materials. The method successfully solves dense matrices involving more than 1 million unknowns with fast CPU time and modest memory consumption.

In [14], we developed an $\mathcal{H}^2$-matrix-based linear-time IE solver for electromagnetic analysis. However, the error of $\mathcal{H}^2$-matrix-based representation of an electrodynamic problem and its impact on computational complexity was not addressed clearly. In [15], we gave a detailed error analysis of the $\mathcal{H}^2$-matrix-based representation of electrodynamic problems. In addition, we showed that a direct application of $\mathcal{H}^2$-matrix-based techniques to electrodynamic problems would result in a complexity greater than $O(N)$, and hence we developed a rank function to maintain the same order of accuracy in a range of electric sizes without compromising $O(N)$ computational complexity. In this work, we give a detailed performance analysis of the accuracy control scheme. We also report simulation results of large-scale scattering problems involving 60 wavelengths and more than 1 million unknowns.

The remainder of this paper is organised as follows. In Section 2, an IE formulation is briefed. In Section 3, the data-sparse representation of the dense system of equations is detailed, which includes an $\mathcal{H}$-matrix representation and its error bound, an $\mathcal{H}^2$-matrix representation and its error bound, and $\mathcal{H}$- and $\mathcal{H}^2$-matrix block partition. In Section 4, we show a centre-point $\mathcal{H}$-matrix-based method of linear complexity, which can be used for accurate analysis of large-scale problems having small electric sizes. In Section 5, we show a linear-complexity $\mathcal{H}^2$-matrix-based method of controlled accuracy, which can be used for large-scale analysis of electrodynamic problems across a wide range of electric sizes. In Section 6, we give numerical results to demonstrate the accuracy and efficiency of the proposed methods. Section 7 relates to our conclusions.

2 IE formulations

The proposed methods are kernel independent, and hence suitable for any IE-based formulation. Here, without loss of generality, we give two examples: an IE formulation for electrostatic analysis and an electric-field IE for electrodynamic analysis.

2.1 IE formulation for electrostatic analysis

The electric potential in a multi-conductor structure embedded in an inhomogeneous material satisfies [16]

$$\Phi(\boldsymbol{r}) = \int_{S_{\mathrm{c}}} \sigma_{\mathrm{c}}(\boldsymbol{r}')g(\boldsymbol{r}, \boldsymbol{r}')\,\mathrm{d}r' + \int_{S_{\mathrm{d}}} \sigma_{\mathrm{d}}(\boldsymbol{r}')g(\boldsymbol{r}, \boldsymbol{r}')\,\mathrm{d}r'$$

where σ_{c} is the surface charge density on conducting surface S_{c}, σ_{d} is the equivalent polarisation charge density on dielectric interfaces S_{d}, Φ is the electric scalar potential, $\boldsymbol{r}$ is the observation point, $\boldsymbol{r}'$ is the source point and g is the static Green's function. The normal continuity of electric flux density is then imposed at the dielectric interface when solving the problem. A method-of-moments-based solution of the IE results in the following linear system

$$\boldsymbol{G}q = v$$

where $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{P}_{\mathrm{cc}} & \boldsymbol{P}_{\mathrm{cd}} \\ \boldsymbol{E}_{\mathrm{dc}} & \boldsymbol{E}_{\mathrm{dd}} \end{bmatrix}$, $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{c}} \\ \boldsymbol{q}_{\mathrm{d}} \end{bmatrix}$ and $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_{\mathrm{c}} \\ 0 \end{bmatrix}$, in which $\boldsymbol{q}_{\mathrm{c}}$ and $\boldsymbol{q}_{\mathrm{d}}$ are the charge vectors of the conductor panels and dielectric–dielectric interface panels, respectively, and $\boldsymbol{v}_{\mathrm{c}}$ is the potential vector associated with the conductor panels. The entries of $\boldsymbol{P}$ and $\boldsymbol{E}$ are

$$\boldsymbol{P}_{ij} = \frac{1}{a_i}\frac{1}{a_j}\int_{S_i}\int_{S_j} g(\boldsymbol{r}, \boldsymbol{r}')\,\mathrm{d}r\,\mathrm{d}r'$$

$$\boldsymbol{E}_{ij} = (\varepsilon_a - \varepsilon_b)\frac{\partial}{\partial n_a}\frac{1}{a_i}\frac{1}{a_j}\int_{S_i}\int_{S_j} g(\boldsymbol{r}, \boldsymbol{r}')\,\mathrm{d}r\,\mathrm{d}r'$$

doi: 10.1049/iet-map.2009.0229

where a_i and a_j are the areas of panels S_i and S_j, respectively, $\hat{\boldsymbol{n}}$ is a unit vector normal to the dielectric interface, ε_a and ε_b are the permittivity in the two dielectric regions separated by the interface and g is the Green's function $g = (\boldsymbol{r}, \boldsymbol{r}') = 1/(4\pi\varepsilon|\boldsymbol{r} - \boldsymbol{r}'|)$. The diagonal entries of $\boldsymbol{E}_{\mathrm{dd}}$ are $e_{ij} = (\varepsilon_a + \varepsilon_b)/(2a_i\varepsilon_0)$.

2.2 Electric-field IE for electrodynamic analysis

Consider a three-dimensional arbitrarily shaped conducting object immersed in a medium characterised by permittivity ε and permeability μ. The object is illuminated by an incident wave $\boldsymbol{E}_{\mathrm{i}}$ that induces current $\boldsymbol{J}_{\mathrm{s}}$ on the conducting surface. The current satisfies the following electric-field IE [1, 17]

$$\boldsymbol{E}_{\mathrm{i}}|_{\tan} = \iint_S \left[\mathrm{j}\omega\mu \boldsymbol{J}_{\mathrm{s}}(\boldsymbol{r}')g(\boldsymbol{r}, \boldsymbol{r}') - \frac{\mathrm{j}}{\omega\varepsilon}(\nabla' \cdot \boldsymbol{J}_{\mathrm{s}}(\boldsymbol{r}'))\nabla' g(\boldsymbol{r}, \boldsymbol{r}') \right]_{\tan} \mathrm{d}s' \tag{1}$$

in which the Green's function $g(\boldsymbol{r}, \boldsymbol{r}') = \mathrm{e}^{-\mathrm{j}\kappa|\boldsymbol{r}-\boldsymbol{r}'|}/(4\pi|\boldsymbol{r} - \boldsymbol{r}'|)$, ω is the angular frequency and κ is the wave number which is $\omega\sqrt{\mu\varepsilon}$. The subscript 'tan' denotes the component that is tangential to the conducting surface S. By expanding the unknown surface current density $\boldsymbol{J}_{\mathrm{s}}$ using Rao-Wilton-Glisson (RWG) basis functions [17], and applying Galerkin's method to (1), we obtain

$$\iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) \cdot \boldsymbol{E}_{\mathrm{i}}(\boldsymbol{r})\,\mathrm{d}s = \sum_{n=1}^{N} I_n \iint_{S_m} \mathrm{d}s \iint_{S_n} \mathrm{d}s' \left[\mathrm{j}\omega\mu \boldsymbol{J}_m(\boldsymbol{r}) \cdot \boldsymbol{J}_n(\boldsymbol{r}') - \frac{\mathrm{j}}{\omega\varepsilon}(\nabla \cdot \boldsymbol{J}_m(\boldsymbol{r}))(\nabla' \cdot \boldsymbol{J}_n(\boldsymbol{r}')) \right] g(\boldsymbol{r}, \boldsymbol{r}') \tag{2}$$

in which $\boldsymbol{J}_m(\boldsymbol{J}_n)$ are basis functions, and N is the total number of basis functions. Equation (2) can be written as

$$\boldsymbol{G}I = V \tag{3}$$

where

$$\boldsymbol{G}_{mn} = \iint_{S_m} \mathrm{d}s \iint_{S_n} \mathrm{d}s' \left[\mathrm{j}\omega\mu \boldsymbol{J}_m(\boldsymbol{r}) \cdot \boldsymbol{J}_n(\boldsymbol{r}') - \frac{\mathrm{j}}{\omega\varepsilon}(\nabla \cdot \boldsymbol{J}_m(\boldsymbol{r}))(\nabla' \cdot \boldsymbol{J}_n(\boldsymbol{r}')) \right] g(\boldsymbol{r}, \boldsymbol{r}') \tag{4}$$

and

$$V_m = \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) \cdot \boldsymbol{E}_{\mathrm{i}}(\boldsymbol{r})\,\mathrm{d}s \tag{5}$$

A straightforward approach to solving (3) can be very expensive, since matrix $\boldsymbol{G}$ is dense in the sense that all entries are non-zero. Our approach is to approximate $\boldsymbol{G}$ by a matrix (with error well controlled), which can be stored in a data-sparse format, that is, $\boldsymbol{G}$ can be specified by few parameters, from which a significant reduction in complexity can be achieved.

Since the proposed methods are kernel independent, in the following sections, we use the electric-field IE as an example to illustrate the basic procedure.

3 Data-sparse representation, error bound, block partition

3.1 $\mathcal{H}$-matrix representation and its error bound

Denoting the index set of the basis functions used in the discretisation of (1) by $\mathcal{I} = \{1, 2, \ldots, N\}$. We fix two subsets t and s of the $\mathcal{I}$ and define the corresponding domains Ω_t and Ω_s as the union of the supports of the basis functions

$$\Omega_t = \bigcup_{i \in t} \mathrm{supp}(\boldsymbol{J}_i), \quad \Omega_s = \bigcup_{i \in s} \mathrm{supp}(\boldsymbol{J}_i) \tag{6}$$

in which 'supp' denotes the space domain that is occupied by the basis function. If t and s are far away from each other (the criterion will be established soon) and $\mathrm{diam}(\Omega_t) \leq \mathrm{diam}(\Omega_s)$, where $\mathrm{diam}(\cdot)$ is the Euclidean diameter of a set, the original kernel function $g(\boldsymbol{r}, \boldsymbol{r}')$ in (1) can be replaced by a degenerate approximation

$$\tilde{g}^{t,s}(\boldsymbol{r}, \boldsymbol{r}') = \sum_{v \in K} g(\xi_v^t, \boldsymbol{r}') L_v^t(\boldsymbol{r}) \tag{7}$$

where

$$K = \{v \in N^d : v_i \leq p \quad \text{for all } i \in \{1, \ldots, d\}\} = \{1, \ldots, p\}^d \tag{8}$$

in which $d = 1$, 2, 3 for one-, two-, and three-dimensional (3-D) problems, respectively, and p is the number of interpolation points along each dimension. In (7), $(\xi_v^t)_{v \in K}$ is a family of interpolation points in t, and $(L_v^t)_{v \in K}$ are the corresponding Lagrange polynomials satisfying $L_v(\xi_\tau) = \delta_{v,\tau}$ for all $v, \tau \in K$. The interpolation in (7) is performed on an axis-parallel bounding box $\boldsymbol{Q}_t \supseteq \Omega_t$, which is the tensor product of intervals. In a d-dimensional case, $\boldsymbol{Q}_t = [a_1, b_1] \times \cdots \times [a_d, b_d]$. The $(\xi_v^t)_{v \in K}$ and $(L_v^t)_{v \in K}$ can be written as

$$\xi_v^t := (\xi_{v_1}^{[a_1,b_1]}, \ldots, \xi_{v_d}^{[a_d,b_d]}), \quad L_v^t := (L_{v_1}^{[a_1,b_1]}, \ldots, L_{v_d}^{[a_d,b_d]}) \tag{9}$$

The advantage of the degenerate approximation is two-fold. First, the double integral in (4) is separated into two single

integrals

$$\tilde{\boldsymbol{G}}_{mn}^{t,s} = \sum_{v\in K} \mathrm{j}\omega\mu \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) L_v^t(\boldsymbol{r})\,\mathrm{d}s \cdot \iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}') g(\xi_v^t, \boldsymbol{r}')\,\mathrm{d}s' - \sum_{v\in K} \frac{\mathrm{j}}{\omega\varepsilon} \iint_{S_m} (\nabla\cdot \boldsymbol{J}_m(\boldsymbol{r})) L_v^t(\boldsymbol{r})\,\mathrm{d}s \times \iint_{S_n} (\nabla'\cdot \boldsymbol{J}_n(\boldsymbol{r}')) g(\xi_v^t, \boldsymbol{r}')\,\mathrm{d}s' \tag{10}$$

for $m \in t$, $n \in s$, $v \in K$. Second, the submatrix $\tilde{\boldsymbol{G}}^{t,s}$ can be represented in a factorised form

$$\tilde{\boldsymbol{G}}^{(t,s)} = \boldsymbol{A}^{(t,s)} (\boldsymbol{B}^{(t,s)})^{\mathrm{T}} \tag{11}$$

where the matrices $\boldsymbol{A}^{(t,s)} \in \mathbb{C}^{t\times K}$ and $\boldsymbol{B}^{(t,s)} \in \mathbb{C}^{s\times K}$, which can be computed as below

$$\boldsymbol{A}^{(t,s)} \boldsymbol{B}^{(t,s)^{\mathrm{T}}} = \boldsymbol{A}_1^{(t,s)} \boldsymbol{B}_1^{(t,s)^{\mathrm{T}}} - \boldsymbol{A}_2^{(t,s)} \boldsymbol{B}_2^{(t,s)^{\mathrm{T}}} \tag{12}$$

in which

$$\begin{aligned} \boldsymbol{A}_{1mv}^{(t,s)} &= \mathrm{j}\omega\mu \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) L_v^t(\boldsymbol{r})\,\mathrm{d}s, \\ \boldsymbol{B}_{1nv}^{(t,s)} &= \iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}') g(\xi_v^t, \boldsymbol{r}')\,\mathrm{d}s' \\ \boldsymbol{A}_{2mv}^{(t,s)} &= \frac{\mathrm{j}}{\omega\varepsilon} \iint_{S_m} (\nabla\cdot \boldsymbol{J}_m(\boldsymbol{r})) L_v^t(\boldsymbol{r})\,\mathrm{d}s, \\ \boldsymbol{B}_{2nv}^{(t,s)} &= \iint_{S_n} (\nabla\cdot \boldsymbol{J}_n(\boldsymbol{r}')) g(\xi_v^t, \boldsymbol{r}')\,\mathrm{d}s' \end{aligned} \tag{13}$$

for all $m \in t$, $n \in s$, $v \in K$.

Clearly, the rank of the matrix $\tilde{\boldsymbol{G}}^{t,s}$ is at most $2\#K$ regardless of the cardinality of t (assuming $\#K < \#t$), where # denotes the cardinality of a set. For example, if $d = 2$ and $p = 2$, that is, the problem is two-dimensional and two interpolation points are used along each dimension, the cardinality of K is 4. And hence, the rank of $\tilde{\boldsymbol{G}}^{t,s}$ is 8 irrespective of the cardinality of t. It is worth mentioning that although $\boldsymbol{A}_1^{(t,s)} \boldsymbol{B}_1^{(t,s)^{\mathrm{T}}}$ in (12) involves a vector operation, which apparently leads to a rank greater than $\#K$, in our implementation, we obtain rank $\#K$ by the following procedure. Let

$$\boldsymbol{A}_1 = (\boldsymbol{A}_x, \boldsymbol{A}_y, \boldsymbol{A}_z) \quad \text{and} \quad \boldsymbol{B}_1 = (\boldsymbol{B}_x, \boldsymbol{B}_y, \boldsymbol{B}_z)$$

$\boldsymbol{A}_1 \boldsymbol{B}_1^{\mathrm{T}}$ becomes

$$\begin{aligned} \boldsymbol{A}_1 \boldsymbol{B}_1^{\mathrm{T}} &= (\boldsymbol{A}_x, \boldsymbol{A}_y, \boldsymbol{A}_z) \cdot (\boldsymbol{B}_x, \boldsymbol{B}_y, \boldsymbol{B}_z)^{\mathrm{T}} \\ &= \boldsymbol{A}_x \boldsymbol{B}_x^{\mathrm{T}} + \boldsymbol{A}_y \boldsymbol{B}_y^{\mathrm{T}} + \boldsymbol{A}_z \boldsymbol{B}_z^{\mathrm{T}} \\ &\simeq \boldsymbol{A}_x \boldsymbol{B}_x^{\mathrm{T}} \oplus_k \boldsymbol{A}_y \boldsymbol{B}_y^{\mathrm{T}} \oplus_k \boldsymbol{A}_z \boldsymbol{B}_z^{\mathrm{T}} \end{aligned}$$

where $\oplus_k$ is used to compute the matrix addition with a targeted rank k, where $k = \#K$. We use $\boldsymbol{G}_{1xy} = \boldsymbol{A}_x \boldsymbol{B}_x^{\mathrm{T}} \oplus_k \boldsymbol{A}_y \boldsymbol{B}_y^{\mathrm{T}}$ as an example to show how $\oplus_k$ can be performed. Since both $\boldsymbol{A}_x \boldsymbol{B}_x^{\mathrm{T}}$ and $\boldsymbol{A}_y \boldsymbol{B}_y^{\mathrm{T}}$ are of rank k, the addition $\boldsymbol{G}_{1xy} = \boldsymbol{A}_x \boldsymbol{B}_x^{\mathrm{T}} + \boldsymbol{A}_y \boldsymbol{B}_y^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{A}_x & \boldsymbol{A}_y \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_x & \boldsymbol{B}_y \end{bmatrix}^{\mathrm{T}}$ should have a rank $2k$. To obtain $\boldsymbol{G}_{1xy}$ of rank k, we compute $\boldsymbol{G}_{1xy}$ by performing a singular value decomposition of $\begin{bmatrix} \boldsymbol{A}_x & \boldsymbol{A}_y \end{bmatrix} \begin{bmatrix} \boldsymbol{B}_x & \boldsymbol{B}_y \end{bmatrix}^{\mathrm{T}}$ [18, p. 123], and then ordering the singular values $s_1 \geq s_2 \geq \ldots s_{2k}$ with respect to their magnitude, and then discard all s_i with $i > k$. The truncated result is shown to be an optimal approximation $\boldsymbol{G}_{1xy}$ with rank k, in the sense that $\|\boldsymbol{G}_{1xy} - \boldsymbol{G}_{1xy}^k\|$ is minimal in the Frobenius or spectral norm [18, p. 109]. After obtaining $\boldsymbol{G}_{1xy}^k$, we can compute $\boldsymbol{A}_1 \boldsymbol{B}_1^{\mathrm{T}} = \boldsymbol{G}_{1xy}^k \oplus_k \boldsymbol{A}_z \boldsymbol{B}_z^{\mathrm{T}}$ in a similar way.

If $\mathrm{diam}(\Omega_t) \geq \mathrm{diam}(\Omega_s)$, $g(\boldsymbol{r}, \boldsymbol{r}')$ in (1) is approximated by

$$\tilde{g}^{t,s}(\boldsymbol{r}, \boldsymbol{r}') = \sum_{v\in K} g(\boldsymbol{r}, \xi_v^s) L_v^s(\boldsymbol{r}') \tag{14}$$

The matrices $\boldsymbol{A}^{(t,s)}$ and $\boldsymbol{B}^{(t,s)}$ are computed using (12) but with

$$\begin{aligned} \boldsymbol{A}_{1mv}^{(t,s)} &= \mathrm{j}\omega\mu \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) g(\boldsymbol{r}, \xi_v^s)\,\mathrm{d}s, \boldsymbol{B}_{1nv}^{(t,s)} = \iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}') L_v^s(\boldsymbol{r}')\,\mathrm{d}s' \\ \boldsymbol{A}_{2mv}^{(t,s)} &= \frac{\mathrm{j}}{\omega\varepsilon} \iint_{S_m} (\nabla\cdot \boldsymbol{J}_m(\boldsymbol{r})) g(\boldsymbol{r}, \xi_v^s)\,\mathrm{d}s, \\ \boldsymbol{B}_{2nv}^{(t,s)} &= \iint_{S_n} (\nabla\cdot \boldsymbol{J}_n(\boldsymbol{r}')) L_v^s(\boldsymbol{r}')\,\mathrm{d}s' \end{aligned} \tag{15}$$

for all $m \in t$, $n \in s$, $v \in K$.

To estimate the error bound of (11), we introduce an admissibility condition [18]

$$(t,\ s) \text{ are admissible} := \begin{cases} \text{True} & \text{if } \min\{\mathrm{diam}(\boldsymbol{Q}_t), \mathrm{diam}(\boldsymbol{Q}_s)\} \leq \eta\, \mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s) \\ \text{False} & \text{otherwise} \end{cases} \tag{16}$$

in which $\mathrm{dist}(\cdot, \cdot)$ is the Euclidean distance of two sets, and η is a parameter that controls the admissibility condition. This condition ensures that we are dealing with a region where the Green's function is expected to be smooth or at least separable.

From [19, p. 328], let $g \in C^\infty(\boldsymbol{Q}_t)$ such that there are positive real constants C_g and γ_g satisfying

$$\|\partial_j^n g\|_{\infty, \boldsymbol{Q}_t} \leq C_g \gamma_g^n n! \tag{17}$$

doi: 10.1049/iet-map.2009.0229

for all $j = \{1,2, \ldots, d\}$ and $n \in \mathbb{N}_0$, we have

$$\|g(\boldsymbol{r}, \boldsymbol{r}') - \tilde{g}^{(t,s)}(\boldsymbol{r}, \boldsymbol{r}')\|_{\infty, \boldsymbol{Q}_t \times \boldsymbol{Q}_s} \leq 8ed(\Lambda_p)^d p C_g (1 + \gamma_g \min\{\mathrm{diam}(Q_t), \mathrm{diam}(Q_s)\}) \times \left(1 + \frac{2}{\gamma_g \min\{\mathrm{diam}(Q_t), \mathrm{diam}(Q_s)\}}\right)^{-p} \quad (18)$$

where Λ_p is a constant related to p and the interpolation scheme. The parameters C_g and γ_g are dependent on the kernel function g. They are derived as follows.

From $g(\boldsymbol{r}, \boldsymbol{r}') = \mathrm{e}^{-\mathrm{j}\kappa R}/4\pi R$, where

$$R = |\boldsymbol{r} - \boldsymbol{r}'| = \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2}$$

we have

$$|\partial R_\zeta| = \left|\frac{x - x'}{R}\right| \leq 1, \quad \zeta = x, y, z \quad (19)$$

Therefore

$$\|\partial_j^n g\|_{\infty, \boldsymbol{Q}_t} \leq \frac{\kappa^n}{R} + \frac{n\kappa^{n-1}}{R^2} + \frac{n(n-1)\kappa^{n-2}}{R^3} + \cdots + \frac{n(n-1)\cdots 2\kappa}{R^n} + \frac{n!}{R^{n+1}} \quad (20)$$

where κ is the wave number.

From (20), we can derive

$$C_g = \frac{1}{4\pi\,\mathrm{dist}(Q_t, Q_s)}, \quad \gamma_g = \kappa + \frac{1}{\mathrm{dist}(Q_t, Q_s)} \quad (21)$$

Hence, from (18), we have

$$\|g(\boldsymbol{r}, \boldsymbol{r}') - \tilde{g}^{(t,s)}(\boldsymbol{r}, \boldsymbol{r}')\|_{\infty, \boldsymbol{Q}_t \times \boldsymbol{Q}_s} \leq 8ed(\Lambda_p)^d p \frac{1}{\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s)} \times \left[1 + \left(\kappa + \frac{1}{\mathrm{dist}(Q_t, Q_s)}\right) \min(\mathrm{diam}(Q_t), \mathrm{diam}(Q_s))\right] \times \left[1 + \frac{2}{(\kappa + (1/\mathrm{dist}(Q_t, Q_s)))\min(\mathrm{diam}(Q_t), \mathrm{diam}(Q_s))}\right]^{-p} \quad (22)$$

If the admissibility condition given in (16) is satisfied, that is,

$$\min\{\mathrm{diam}(\boldsymbol{Q}_t), \mathrm{diam}(\boldsymbol{Q}_s)\} \leq \eta\,\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s)$$

from (22), we obtain

$$\|g(\boldsymbol{r}, \boldsymbol{r}') - \tilde{g}^{(t,s)}(\boldsymbol{r}, \boldsymbol{r}')\|_{\infty, \boldsymbol{Q}_t \times \boldsymbol{Q}_s} \leq 8ed(\Lambda_p)^d p \frac{1}{\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s)} [1 + \kappa\eta\,\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s) + \eta] \times \left[1 + \frac{2}{\kappa\eta\,\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s) + \eta}\right]^{-p} \quad (23)$$

Clearly, exponential convergence with respect to p can be obtained irrespective of the electric size $\kappa\,\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s)$ and the choice of η. In addition, given a required level of accuracy, when the electric size increases, the error of the $\mathcal{H}$-matrix approximation can be controlled to the same level either by decreasing η to maintain a constant $\kappa\eta\,\mathrm{dist}(\boldsymbol{Q}_t, \boldsymbol{Q}_s)$, or by increasing the number of interpolation points p, or by the combination of both.

Matrices that exhibit a property shown in (11) belong to $\mathcal{H}$ matrices. In an $\mathcal{H}$ matrix, the blocks that satisfy the admissibility condition are represented by low rank matrices of the form $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}^{\mathrm{T}}$, where $\boldsymbol{A}$ is of dimension $m \times r$, $\boldsymbol{B}$ is of dimension $n \times r$ and r denotes the rank of $\boldsymbol{C}$ with $r < m$ and $r < n$. The low-rank matrices are called as rkmatrices. The blocks that do not satisfy the admissibility condition are represented by full matrices. As can be seen from (23), the error of the $\mathcal{H}$-matrix representation is well controlled. It was shown that an $\mathcal{H}$ matrix of dimension N can be compactly stored in $O(N \log N)$ parameters [7–9].

3.2 $\mathcal{H}^2$-matrix representation and its error bound

If subsets t and s of the $\mathcal{I}$ satisfy a strong admissibility condition [18] as shown below

$$(t, s)\,\text{are admissible} := \begin{cases} \text{True} & \text{if } \max\{\mathrm{diam}(\Omega_t), \mathrm{diam}(\Omega_s)\} \leq \eta\,\mathrm{dist}(\Omega_t, \Omega_s) \\ \text{False} & \text{otherwise} \end{cases} \quad (24)$$

the original kernel function $g(\boldsymbol{r}, \boldsymbol{r}')$ in (1) can be replaced by

$$\tilde{g}^{t,s}(\boldsymbol{r}, \boldsymbol{r}') = \sum_{v \in K^t} \sum_{\mu \in K^s} g(\xi_v^t, \xi_\mu^s) L_v^t(\boldsymbol{r}) L_\mu^s(\boldsymbol{r}') \quad (25)$$

where $(\xi_v^t)_{v \in K^t}$ is a family of interpolation points in t; $(\xi_\mu^s)_{\mu \in K^s}$ is a family of interpolation points in s; and $(L_v^t)_{v \in K^t}$ and $(L_\mu^s)_{\mu \in K^s}$ are the corresponding Lagrange polynomials satisfying $L_v(\xi_\tau) = \delta_{v,\tau}$ for all $v, \tau \in K^t$, and $L_\mu(\xi_\tau) = \delta_{\mu,\tau}$ for all $\mu, \tau \in K^s$.

The double integral in (4) can then be separated into two single integrals

$$\tilde{\boldsymbol{G}}^{t,s}_{mn} := \sum_{v\in K^t}\sum_{\mu\in K^s} \mathrm{j}\omega\mu g(\xi^t_v, \xi^s_\mu) \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) L^t_v(\boldsymbol{r})\,\mathrm{d}s \cdot \iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}') L^s_\mu(\boldsymbol{r}')\,\mathrm{d}s' - \sum_{v\in K^t}\sum_{\mu\in K^s} \frac{\mathrm{j}}{\omega\varepsilon} g(\xi^t_v, \xi^s_\mu) \times \iint_{S_m} (\nabla\cdot\boldsymbol{J}_m(\boldsymbol{r})) L^t_v(\boldsymbol{r})\,\mathrm{d}s \iint_{S_n} (\nabla'\cdot\boldsymbol{J}_n(\boldsymbol{r}')) L^s_\mu(\boldsymbol{r}')\,\mathrm{d}s'$$

for $m \in t$, $n \in s$, $v \in K^t$, $\mu \in K^s$. As a result, the submatrix $\tilde{\boldsymbol{G}}^{t,s}$ can be represented in a factorised form

$$\tilde{\boldsymbol{G}}^{t,s} := \boldsymbol{V}^t \boldsymbol{S}^{t,s} \boldsymbol{V}^{s\mathrm{T}}, \quad \boldsymbol{V}^t \in \mathbb{C}^{t\times 2K^t}, \ \boldsymbol{S}^{t,s} \in \mathbb{C}^{2K^t\times 2K^s}, \ \boldsymbol{V}^s \in \mathbb{C}^{s\times 2K^s} \tag{26}$$

where

$$\boldsymbol{V}^t = [\boldsymbol{V}^t_1 \quad \boldsymbol{V}^t_2], \quad \boldsymbol{V}^s = [\boldsymbol{V}^s_1 \quad \boldsymbol{V}^s_2], \quad \boldsymbol{S}^{t,s} = \begin{bmatrix} \boldsymbol{S}^{t,s}_1 & 0 \\ 0 & \boldsymbol{S}^{t,s}_2 \end{bmatrix}$$
$$\boldsymbol{V}^t_1, \ \boldsymbol{V}^t_2 \in \mathbb{C}^{t\times K^t}, \quad \boldsymbol{V}^s_1, \boldsymbol{V}^s_2 \in \mathbb{C}^{s\times K^s}, \quad \boldsymbol{S}^{t,s}_1, \boldsymbol{S}^{t,s}_2 \in \mathbb{C}^{K^t\times K^s} \tag{27}$$

and

$$\boldsymbol{V}^t_{1mv} = \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r}) L^t_v(\boldsymbol{r})\,\mathrm{d}s, \boldsymbol{V}^t_{2mv} = \iint_{S_m} (\nabla\cdot\boldsymbol{J}_m(\boldsymbol{r})) L^t_v(\boldsymbol{r})\,\mathrm{d}s$$
$$\boldsymbol{V}^s_{1n\mu} = \iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}') L^s_\mu(\boldsymbol{r}')\,\mathrm{d}s', \boldsymbol{V}^s_{2n\mu} = \iint_{S_n} (\nabla'\cdot\boldsymbol{J}_n(\boldsymbol{r}')) L^s_\mu(\boldsymbol{r}')\,\mathrm{d}s'$$
$$\boldsymbol{S}^{t,s}_{1v\mu} = \mathrm{j}\omega\mu g(\xi^t_v, \xi^s_\mu), \quad \boldsymbol{S}^{t,s}_{2v\mu} = \frac{-\mathrm{j}}{\omega\varepsilon} g(\xi^t_v, \xi^s_\mu), \quad m \in t, \ n \in s, \ v \in K^t, \ \mu \in K^s \tag{28}$$

Clearly, the rank of matrix $\tilde{\boldsymbol{G}}^{t,s}$ is at most $2\#K^t$ or $2\#K^s$ regardless of the cardinality of t and s.

Representation (26) forms an $\mathcal{H}^2$-matrix-based representation of $\boldsymbol{G}$ if the same space of polynomials are used across t and s. It enables an efficient computation of matrix–vector multiplication. $\mathcal{H}^2$ matrices are a specialised subclass of $\mathcal{H}$ matrices. The nested structure is the key difference between general $\mathcal{H}$ matrices and $\mathcal{H}^2$ matrices. It was shown that an $\mathcal{H}^2$ matrix of dimension N can be compactly stored in $O(N)$ parameters [10–12]. In addition, like $\mathcal{H}$-matrix-based techniques, the $\mathcal{H}^2$-matrix-based techniques are kernel independent.

When the strong admissibility condition (24) is satisfied, following the same error analysis derived for $\mathcal{H}$-based representation, we obtain the error bound for $\mathcal{H}^2$-based representation

$$\|g(r,r') - \tilde{g}^{(t,s)}(r,r')\|_{\infty,Q_t\times Q_s} \le \frac{4ed}{\pi}(\Lambda_p)^{2d} p \frac{1}{\mathrm{dist}(Q_t,Q_s)}[1+\sqrt{2}\kappa\eta\,\mathrm{dist}(Q_t,Q_s)+\sqrt{2}\eta] \times \left[1+\frac{2}{\sqrt{2}\kappa\eta\,\mathrm{dist}(Q_t,Q_s)+\sqrt{2}\eta}\right]^{-p} \tag{29}$$

As can be seen from (29), exponential convergence with respect to p can be obtained irrespective of the electric size $\kappa\eta$ dist(Q_t,Q_s) and the choice of η. In [15], we showed an approach that can be used to systematically control the accuracy of $\mathcal{H}^2$-matrix-based solutions to the same order in a wide range of electric sizes without compromising computational complexity. In Section 5, we give a detailed performance analysis of this accuracy control approach.

3.3 $\mathcal{H}$- and $\mathcal{H}^2$-matrix block partition

In this section, we explain how to partition a product index set $\mathcal{I}\times\mathcal{J}$ into admissible and inadmissible blocks. A trivial partition would be $\mathcal{P} - \{(i,j)|i\in\mathcal{I}, j\in\mathcal{J}\}$ where only 1×1 blocks of rank 1 appear. In this case matrix $\tilde{\boldsymbol{G}}$ is identical to $\boldsymbol{G}$. However, this partition is not efficient. In the following, we present general strategies by using a cluster tree and a block cluster tree [20, 21] for the construction of suitable partitions, which enables an efficient computation of $\boldsymbol{G}$.

For the index set of the basis functions $\mathcal{I} = \{1, 2, \ldots, N\}$, we construct a cluster tree $T_\mathcal{I}$, which is a tree with vertex set V and edge set E. Each vertex in the tree is called as a cluster. The label of cluster t is denoted by $\hat{t}$. The set of sons for a cluster $t \in T_\mathcal{I}$ is denoted by sons(t). The root of the tree is the index set $\mathcal{I} = \{1, 2, \ldots, N\}$. To construct a cluster tree, we start from the full index set of basis functions $\mathcal{I}$. We split the computational domain into two subdomains. We continue to split until the number of unknowns in each subdomain is less than or equal to the leafsize which is a parameter to control the tree depth. As a result, we generate a cluster tree $T_\mathcal{I}$ as shown in Fig. 1. Clusters with indices no more than leafsize are leaves. The set of leaves of $T_\mathcal{I}$ is denoted by $\mathcal{L}_\mathcal{I}$.

Constructing an admissible block cluster tree from the cluster trees $T_\mathcal{I}$ and $T_\mathcal{I}$ and a given admissibility condition can be done recursively. We test blocks level by level starting

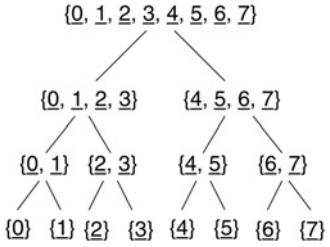

Figure 1 *Cluster tree*

doi: 10.1049/iet-map.2009.0229

with Root($T_{\mathcal{I}}$) and Root($T_{\mathcal{I}}$), and descending in the tree. Given two clusters $t \in T_{\mathcal{I}}$ and $s \in T_{\mathcal{I}}$, we check the admissibility. If the two clusters are admissible, we are done. If they are not admissible, we repeat the procedure for all combinations of sons of t and sons of s. The aforementioned procedure of constructing a block cluster tree results in a matrix structure as shown in Fig. 2*b*. Each matrix block corresponds to a link drawn between $T_{\mathcal{I}}$ and $T_{\mathcal{I}}$ as shown in Fig. 2*a*. Links drawn on the upper level of the tree correspond to admissible blocks denoted by $\mathcal{L}^{+}_{\mathcal{I}\times\mathcal{I}}$, whereas those drawn at the bottommost level represent inadmissible ones denoted by $\mathcal{L}^{-}_{\mathcal{I}\times\mathcal{I}}$. In addition, based on our approach of constructing the cluster tree, it is clear that if there exists a link at the upper level between two clusters' fathers, there cannot exist a link between the two clusters. In Fig. 2*b*, admissible blocks are represented by shaded blocks.

In Fig. 3, we plot the $\mathcal{H}^2$-matrix block structure of a square plate for $N = 1160$, and 3605, respectively, with leafsize chosen as 38, and η set as 2. The unshaded blocks are admissible blocks that are represented by low rank matrices, whereas the shaded ones are inadmissible blocks that are represented by full matrices.

In our numerical computation, the matrix shown in Fig. 2*b* is never formed. Instead, we use the block cluster tree shown in Fig. 2*a* to carry out efficient computation.

4 Centre-point $\mathcal{H}$-matrix-based method of $O(N)$ complexity

It was shown that storage requirements and matrix–vector multiplications of $\mathcal{H}$ matrices are of complexity $O(N \log N)$. In the following, we show that the complexity can be reduced to $O(N)$ by developing a centre-point $\mathcal{H}$-matrix-based method. The method can be viewed as an $\mathcal{H}$-matrix-based method of rank one. It can also be viewed

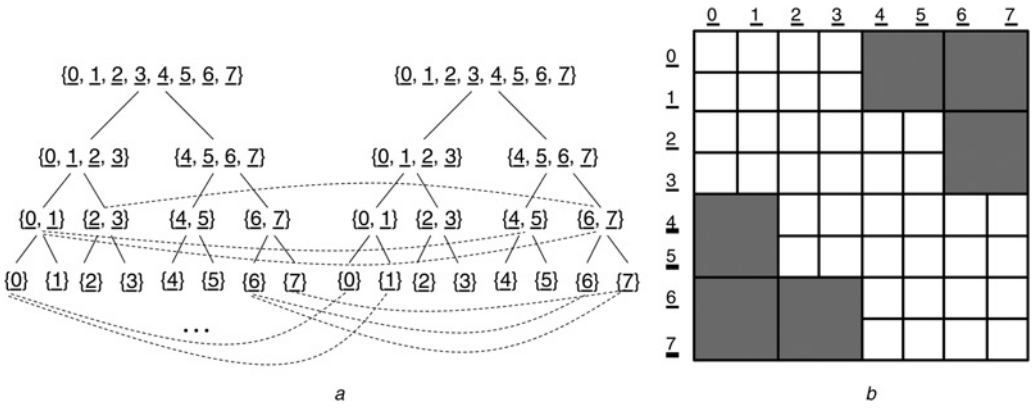

Figure 2 *Block cluster tree and its $\mathcal{H}^2$ matrix structure*

a An admissible block cluster tree
b An $\mathcal{H}$ ($\mathcal{H}^2$)-matrix structure

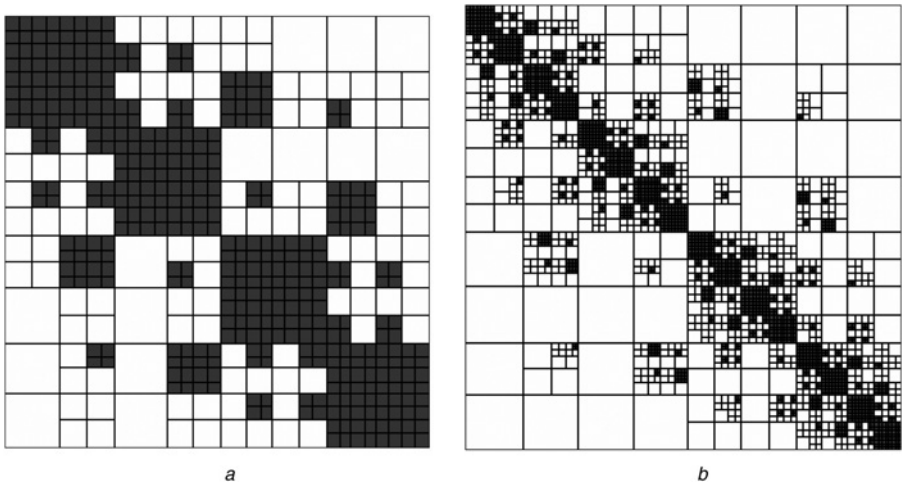

Figure 3 *$\mathcal{H}^2$-matrix block structure of a square plate*

a $N = 1160$
b $N = 3605$

as an $\mathcal{H}^2$-matrix-based method of rank one. The method is suitable for the analysis of static problems or problems having small electric sizes in which rank one is enough for achieving a good accuracy.

Consider an admissible pair (t, s), if we make the admissibility condition used in the $\mathcal{H}$-matrix-based representation more stringent by reducing η or by changing the condition to a strong admissibility condition like (24), further approximation of the kernel function $g(\boldsymbol{r}, \boldsymbol{r}')$ can be made without degrading the accuracy. We can use $\boldsymbol{r}_c$ and $\boldsymbol{r}_c'$ to replace $\boldsymbol{r}$ and $\boldsymbol{r}'$ in $g(\boldsymbol{r}, \boldsymbol{r}')$, where $\boldsymbol{r}_c$ and $\boldsymbol{r}_c'$ are the centre points of supports Ω_t, and Ω_s, respectively. $\tilde{\boldsymbol{G}}^{t,s}$ in (10) hence becomes

$$\begin{aligned}\tilde{\boldsymbol{G}}_{mn}^{t,s} = {} & \mathrm{j}\omega\mu g^{t,s}(\boldsymbol{r}_c, \boldsymbol{r}_c')\left(\iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r})\,\mathrm{d}s\right)\left(\iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}')\,\mathrm{d}s'\right) \\ & - \frac{\mathrm{j}}{\omega\varepsilon} g^{t,s}(\boldsymbol{r}_c, \boldsymbol{r}_c')\left(\iint_{S_m} (\nabla\cdot\boldsymbol{J}_m(\boldsymbol{r}))\,\mathrm{d}s\right) \\ & \times \left(\iint_{S_n} (\nabla\cdot\boldsymbol{J}_n(\boldsymbol{r}'))\,\mathrm{d}s'\right) \end{aligned} \tag{30}$$

which can be written in a factorised form as

$$\tilde{\boldsymbol{G}}_{mn}^{t,s} = (\boldsymbol{A}^{t,s}\boldsymbol{D}^{t,s}\boldsymbol{B}^{t,s^{\mathrm{T}}})_{mn} \tag{31}$$

where

$$\begin{aligned} \boldsymbol{A}^{t,s} &= \begin{bmatrix}\boldsymbol{A}_1^{t,s} & \boldsymbol{A}_2^{t,s}\end{bmatrix}, \quad \boldsymbol{B}^{t,s} = \begin{bmatrix}\boldsymbol{B}_1^{t,s} & \boldsymbol{B}_2^{t,s}\end{bmatrix} \\ \boldsymbol{D}^{t,s} &= \begin{bmatrix} \mathrm{j}\omega\mu g^{t,s}(\boldsymbol{r}_c, \boldsymbol{r}_c') & 0 \\ 0 & -\dfrac{\mathrm{j}}{\omega\varepsilon} g^{t,s}(\boldsymbol{r}_c, \boldsymbol{r}_c') \end{bmatrix} \end{aligned} \tag{32}$$

and $\boldsymbol{A}_1^{t,s}, \boldsymbol{A}_2^{t,s} \in \mathbb{C}^{t\times 1}$, $\boldsymbol{B}_1^{t,s}, \boldsymbol{B}_2^{t,s} \in \mathbb{C}^{s\times 1}$. They are

$$\begin{aligned} \boldsymbol{A}_{1\,m}^{t,s} &= \iint_{S_m} \boldsymbol{J}_m(\boldsymbol{r})\,\mathrm{d}s \quad \boldsymbol{A}_{2m}^{t,s} = \iint_{S_m} (\nabla\cdot\boldsymbol{J}_m(\boldsymbol{r}))\,\mathrm{d}s \\ \boldsymbol{B}_{1n}^{t,s} &= \iint_{S_n} \boldsymbol{J}_n(\boldsymbol{r}')\,\mathrm{d}s' \quad \boldsymbol{B}_{2n}^{t,s} = \iint_{S_n} (\nabla'\cdot\boldsymbol{J}_n(\boldsymbol{r}'))\,\mathrm{d}s' \end{aligned} \tag{33}$$

for $m \in t$, $n \in s$.

From (33) it can be seen that $\boldsymbol{A}^{t,s}$ is independent of cluster s, whereas $\boldsymbol{B}^{t,s}$ is independent of cluster t. Therefore they can be directly denoted by $\boldsymbol{A}^t$ and $\boldsymbol{B}^s$. Since $\boldsymbol{A}^t$ and $\boldsymbol{B}^s$ represent the same integration, $\boldsymbol{B}^t = \boldsymbol{A}^t$. Therefore $\tilde{\boldsymbol{G}}^{t,s}$ is solely determined by $\boldsymbol{A}^t$, $t \in T_{\mathcal{I}}$, and $\boldsymbol{D}^{t,s}$. In fact, this representation of $\tilde{\boldsymbol{G}}^{t,s}$ is a special form of (10) in the case of rank one.

From (33) it can also been seen that a cluster t with $\mathrm{sons}(t) = \{t_1, t_2\}$ $(t_1 \neq t_2)$ satisfies

$$\boldsymbol{A}^t = \begin{bmatrix}\boldsymbol{A}^{t1} \\ \boldsymbol{A}^{t2}\end{bmatrix} \tag{34}$$

This means that we do not need to store $\boldsymbol{A}^t$ for each cluster $t \in T_{\mathcal{I}}$, we only need to store $\boldsymbol{A}^t$ for each leaf cluster. The $\boldsymbol{A}^t$ for all non-leaf clusters can be represented by the combination of $\boldsymbol{A}^t$ from leaf clusters. This enables an efficient matrix–vector multiplication of $O(N)$ complexity.

4.1 Storing $\boldsymbol{G}$ in $O(N)$ complexity

As can be seen from (31) and (33), all the admissible blocks are formed by $(\boldsymbol{A}^t)_{t\in T_{\mathcal{I}}}$ and $(\boldsymbol{D}^{t,s})_{t\in T_{\mathcal{I}}, s\in T_{\mathcal{I}}}$

For each leaf cluster $t \in T_{\mathcal{I}}$, we store the matrix $\boldsymbol{A}^t$ that requires $O(1)\#\hat{t}$ units of storage. For each non-leaf cluster $t \in T_{\mathcal{I}}$, there is no additional storage requirement since it can be represented by its two sons as given in (34).

To store $\boldsymbol{D}^{t,s}$, for each admissible block $b = (t, s)$, we store $g^{t,s}(\boldsymbol{r}_c, \boldsymbol{r}_c')$, $\mathrm{j}\omega\mu$, and $\mathrm{j}/\omega\varepsilon$ based on (32), which requires only $O(1)$ units of storage; whereas for the inadmissible block b, the matrix $\boldsymbol{G}^{t,s}$ needs $O(1)\#\hat{t}\#\hat{s}$ units of storage. The total storage can be obtained by summing over the aforementioned components as

$$\begin{aligned} \text{Total storage} = {} & \mathrm{St}(\text{all leaf clusters}) + \mathrm{St}(\text{all non-leaf clusters}) \\ & + \mathrm{St}(\text{all admissible blocks}) \\ & + \mathrm{St}(\text{all inadmissible blocks}) \\ = {} & \sum_{t\in\mathcal{L}_{\mathcal{I}}} O(1)\#\,\hat{t} + 0 + \sum_{b=(t,s)\in\mathcal{L}^+_{\mathcal{I}\times\mathcal{I}}} O(1) \\ & + \sum_{b=(t,s)\in\mathcal{L}^-_{\mathcal{I}\times\mathcal{I}}} O(1)\#\,\hat{t}\#\,\hat{s} \le O(1)N \\ & + \sum_{t\in\mathcal{T}_{\mathcal{I}}}\sum_{s\in col(t)} O(1) + \sum_{t\in\mathcal{T}_{\mathcal{I}}}\sum_{s\in col(t)} O(1)n_{\min}^2 \\ \le {} & O(1)N + C_{\mathrm{sp}}\sum_{t\in\mathcal{T}_{\mathcal{I}}} O(1) + C_{\mathrm{sp}}\sum_{t\in\mathcal{T}_{\mathcal{I}}} O(1)n_{\min}^2 \\ \le {} & O(1)N + 2O(1)C_{\mathrm{sp}}N + 2O(1)C_{\mathrm{sp}}n_{\min}^2 N \\ \to {} & (O(1) + O(C_{\mathrm{sp}}) + O(C_{\mathrm{sp}}n_{\min}^2))N \end{aligned} \tag{35}$$

where $\mathrm{col}(t) = \{s \in T_{\mathcal{I}} : (t, s) \in T_{\mathcal{I}\times\mathcal{I}}\}$, $n_{\min}$ is leafsize, and C_{sp} is called as a sparsity constant. The constant C_{sp} is the upper bound of the cardinality of the set $\mathrm{col}(t)$. It denotes the maximum number of admissible blocks that can be formed for each cluster $t \in T_{\mathcal{I}}$, which is shown to be a constant [7–9]. As can be seen from (35), the memory complexity of the centre-point-based scheme is linear.

doi: 10.1049/iet-map.2009.0229

4.2 Matrix–vector multiplication of $O(N)$ complexity

Computing $y = \boldsymbol{G}x$ can be decomposed into

$$y_i = \left(\sum_{(t,s)\in\mathcal{L}_{\mathcal{I}\times\mathcal{I}}} \tilde{\boldsymbol{G}}^{t,s} x|_s \right)_i = \left(\sum_{(t,s)\in\mathcal{L}^+_{\mathcal{I}\times\mathcal{I}}} \boldsymbol{A}^t \boldsymbol{D}^{t,s} \boldsymbol{A}^{s\mathrm{T}} x|_s \right)_i + \left(\sum_{(t,s)\in\mathcal{L}^-_{\mathcal{I}\times\mathcal{I}}} \boldsymbol{G}^{t,s} x|_s \right)_i \tag{36}$$

that is multiplying the admissible blocks with the vector, and multiplying the inadmissible blocks with the vector.

For admissible blocks, the matrix–vector multiplication can be performed in three steps:

1. We compute $x^s = \boldsymbol{A}^{s\mathrm{T}} x|_{\hat{s}}$ for all clusters $s \in T_{\mathcal{I}}$. If s is a leaf cluster for tree $T_{\mathcal{I}}$, then we can compute $\boldsymbol{A}^{s\mathrm{T}} x$ directly, which needs $O(1)\#\hat{s}$ operations. If s is not a leaf cluster, then we can first compute x^{s1}, x^{s2} for its two sons s_1, $s_2 \in \mathrm{sons}(s)$. From (34), it can be seen that $\boldsymbol{A}^{s\mathrm{T}} x|_{\hat{s}} = \begin{bmatrix} \boldsymbol{A}^{s1\mathrm{T}} x|_{\hat{s}1} \\ \boldsymbol{A}^{s2\mathrm{T}} x|_{\hat{s}2} \end{bmatrix}$. Clearly, if $x^s := \boldsymbol{A}^{s\mathrm{T}} x|_{\hat{s}}$ for all leaf clusters have been computed, for any other non-leaf cluster, we can directly use the contribution from its two sons to obtain the result of matrix–vector multiplication without any additional operations. Therefore

$$\mathrm{Comp}\left(\sum_{s\in T_{\mathcal{I}}} \boldsymbol{A}^{s\mathrm{T}} x|_{\hat{s}} \right) = \sum_{s\in L_{\mathcal{I}}} O(1)\#\hat{s} = O(1)N \tag{37}$$

2. We compute $y^t = \sum_{s\in S^t} \boldsymbol{D}^{t,s} x^s$ for all clusters $t \in T_{\mathcal{I}}$, where $S^t = \{s \in T_{\mathcal{I}} : (t, s) \in L^+(T_{\mathcal{I}\times\mathcal{I}})\}$, that is, S^t contains all clusters s such that (t, s) is an admissible leaf of the block cluster tree. The multiplication of $\boldsymbol{D}^{t,s} x^s$ requires $O(1)$ operations, and has to be performed for each $s \in S^t$, hence

$$\begin{aligned} &\mathrm{Comp}\left(\sum_{b=(t,s)\in\mathcal{L}^+_{\mathcal{I}\times\mathcal{I}}} \boldsymbol{D}^{t,s} x^s \right) \\ &= \sum_{b=(t,s)\in\mathcal{L}^+_{\mathcal{I}\times\mathcal{I}}} O(1) \le \sum_{b=(t,s)\in\mathcal{L}_{\mathcal{I}\times\mathcal{I}}} O(1) \le \sum_{t\in T_{\mathcal{I}}} \sum_{s\in\mathrm{col}(t)} O(1) \\ &\le C_{\mathrm{sp}} \sum_{t\in T_{\mathcal{I}}} O(1) \le O(1) C_{\mathrm{sp}} \#T_{\mathcal{I}} \le 2O(1) C_{\mathrm{sp}} N \end{aligned} \tag{38}$$

3. We compute $\boldsymbol{A}^t x^s$ for each admissible leaf block (t,s). If t is leaf cluster of $T_{\mathcal{I}}$, we compute $\boldsymbol{A}^t x^s$ directly, the cost of which is $O(1)\#\hat{t}$ operations. If t is a non-leaf cluster, we add its contribution back to its two sons. For example, consider cluster t' which is a son of t. We have $(\boldsymbol{A}^{t'} x^s)_i + (\boldsymbol{A}^t x^s)_i = (\boldsymbol{A}^{t'}(x^s + x^s))_i$. This means that the contribution of $\boldsymbol{A}^t$ to y_i can be efficiently taken into consideration by adding x^s back to its son, the cost of which is $O(1)$ only. Hence, the total complexity is

$$\begin{aligned} &\mathrm{Comp}\left(\sum_{(t,s)\in L^+_{\mathcal{I}\times\mathcal{I}}} \boldsymbol{A}^t x^s \right) \\ &\le \sum_{t\in L_{\mathcal{I}}} \sum_{s\in\mathrm{col}(t)} O(1)\#\hat{t} + \sum_{t\in T_{\mathcal{I}}\setminus L_{\mathcal{I}}} \sum_{s\in\mathrm{col}(t)} O(1) \\ &\le C_{\mathrm{sp}} O(1) N + 2C_{\mathrm{sp}} O(1) N \to O(C_{\mathrm{sp}}) N \end{aligned} \tag{39}$$

For inadmissible blocks, these blocks are treated the same as those in the original matrix $\boldsymbol{G}$, that is, a full-matrix–vector multiplication is performed. The complexity of multiplying each non-admissible block by a vector is $O(n^2_{\min})$. Summing over all the non-admissible blocks, we obtain

$$\begin{aligned} &\mathrm{Comp}\left(\sum_{b=(t,s)\in L^-_{\mathcal{I}\times\mathcal{I}}} \boldsymbol{G}^{t,s} x|_s \right) \\ &\le \sum_{b=(t,s)\in L_{\mathcal{I}\times\mathcal{I}}} O(n^2_{\min}) \\ &\le \sum_{t\in T_{\mathcal{I}}} \sum_{s\in\mathrm{col}(t)} O(n^2_{\min}) \\ &\le 2C_{\mathrm{sp}} O(n^2_{\min}) N \to O(C_{\mathrm{sp}} n^2_{\min}) N \end{aligned} \tag{40}$$

Adding the aforementioned steps, the matrix–vector multiplication requires only $O(N)$ operations.

5 $\mathcal{H}^2$-matrix-based methods of $O(N)$ complexity and controlled accuracy

From (23) and (29), it can be seen that for static problems or problems having small electric sizes, a constant p, and hence a constant rank can keep the accuracy of the $\mathcal{H}$- or $\mathcal{H}^2$-based approximations the same across all the levels of a cluster tree. However, for electrodynamic problems, the use of a constant rank cannot keep the accuracy to the same order when the number of unknowns increases or the electric size increases. This has also been observed in interpolation-based fast IE solvers [22]. It was shown that to ensure the interpolation accuracy, one should use different numbers of interpolation points in the cubes of different sizes at different tree levels. In the following, for the proposed methods, we give a theoretical analysis of this phenomenon by using the error bound derived for $\mathcal{H}$- or $\mathcal{H}^2$-based approximations.

From the properties of a cluster tree, it is known that the lower the tree level is, the larger the cluster size is. Therefore when the tree level is lower, the admissible blocks formed in that level are larger. Hence $1 + 2/[(\kappa + (1/\mathrm{dist}(Q_t, Q_s)))\min(\mathrm{diam}(Q_t), \mathrm{diam}(Q_s))]$ in

(22) becomes smaller, which leads to a slower convergence rate with respect to p. Since increasing frequency is equivalent to increasing the tree depth, in order to keep the same order of accuracy across all tree levels, polynomial order p (the number of interpolation points along one dimension) should be increased when ascending an inverted tree.

Based on the above analysis, we define a polynomial order function that decreases with tree level as

$$p(b) = a + b(L - l(b)) \tag{41}$$

where

$$L = L_{\min} = \min\{\text{level}(\tau):\tau \in \mathcal{L}_{\mathcal{I}}\}, l(b) = \text{level}(t) = \text{level}(s))$$
$$p(b) = a \quad \text{if } L \le l(b) \tag{42}$$

and a, b are two constants. Given a cluster t in an $\mathcal{H}^2$ tree, its rank $k_{\text{var}}(t)$ can be determined from (41) as

$$k_{\text{var}}(t) = p^d(t) = [a + b(L - l(t))]^d \tag{43}$$

where $d = 1, 2, 3$, for 1-, 2-, and 3-D problems, respectively.

To examine the effectiveness of (43) for accuracy control, we did a number of numerical experiments. First, we considered a square plate of four wavelengths, and plotted the error of the $\mathcal{H}^2$-matrix-based $\boldsymbol{G}$ with respect to constant coefficient a with b fixed to be 1. As can be seen from Fig. 4, the larger a, the better the accuracy. In Fig. 5*a*, we plotted the error of the $\mathcal{H}^2$-matrix-based $\boldsymbol{G}$ with respect to electric size when a was chosen as 5, and b chosen as 0. Since b was 0, the rank was in fact a constant across all the tree levels, that is, the rank function was in fact disabled. Hence, we expect that the error of the $\mathcal{H}^2$-matrix-based representation would increase with electric size, which is verified by Fig. 5*a*. In Fig. 5*b*, we plotted the error of the $\mathcal{H}^2$-matrix-based $\boldsymbol{G}$ with respect to electric size when a was chosen as 5, and b was chosen as 2, that is, the rank function was enabled (by setting a non-zero b). A constant order of accuracy can be observed in the entire range. This demonstrates the effectiveness of the proposed rank function for accuracy control. In simulations performed for plotting Fig. 5, the leafsize was chosen as 20, and η was set as 1.

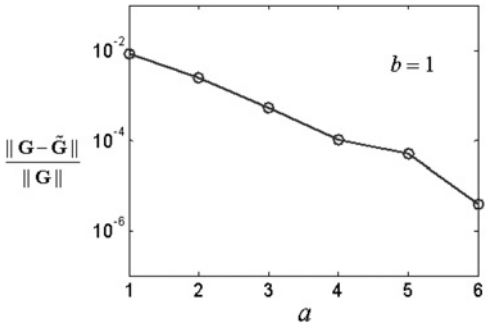

Figure 4 *Effect of a on $\mathcal{H}^2$-matrix-based approximation*

Since the coefficients of the rank function (43) are all constants, when N increases with frequency or electric size, these coefficients do not change. This can render the accuracy the same in a wide range of electric sizes while still keeping the complexity linear for storing $\boldsymbol{G}$ and performing $\mathcal{H}^2$-based matrix–vector multiplication. In [15], we proved that the cost of storage and matrix–vector multiplication is linear based on (43).

6 Numerical results

First, we tested the performance of the proposed centre-point $\mathcal{H}$-matrix-based method for large-scale capacitance extraction. A realistic 3-D on-chip interconnect [16] as shown in Fig. 6 was considered. The relative permittivity is 3.9 in M1 layer, 2.5 from M2 to M6, and 7.0 from M7 to M8. The discretisation of this structure results in 25 556 unknowns. Layers M2–M5 have ten conductors each, whereas layers M7 and M8 have four conductors each, resulting in 48 conductors in total. To test the capability of the proposed method in simulating large-scale examples, the 48-conductor structure shown in Fig. 6 is duplicated horizontally, resulting in 48, 72, 96, 120, 144, 192, 240,

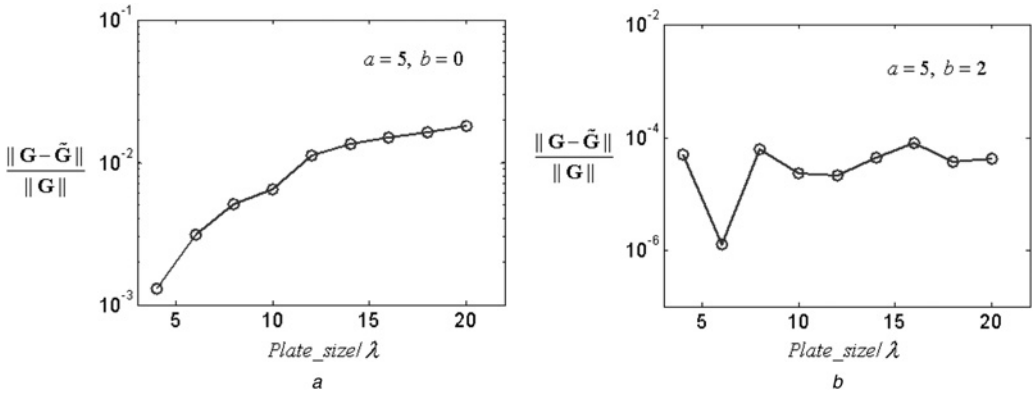

Figure 5 *Effect of b on $\mathcal{H}^2$-based representation*
a $a = 5$, $b = 0$
b $a = 5$, $b = 2$

IET Microw. Antennas Propag., 2010, Vol. 4, Iss. 10, pp. 1583–1596
doi: 10.1049/iet-map.2009.0229

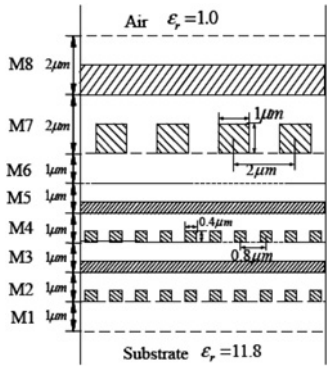

Figure 6 *Illustration of a 3-D on-chip interconnect*

288 and 336 conductors, which lead to more than 1 million unknowns.

As mentioned earlier, the proposed methods are kernel independent, and hence suitable for any IE-based formulation. The IE formulation used for simulating this example is shown in Section 2.1. The simulation parameters are chosen as leafsize = 10 and $\eta = 1$. The error of the proposed centre-point $\mathcal{H}$-matrix-based representation of system matrix $\boldsymbol{G}$ is shown in Fig. 7*a*, with the number of unknowns varying from 25 556 to 1 047 236. A constant order of accuracy can be observed in the entire range. In Fig. 7*a*, the maximal admissible block error is plotted instead of the entire matrix error $\|\boldsymbol{G} - \tilde{\boldsymbol{G}}\|/\|\boldsymbol{G}\|$ because the storage of the original matrix $\boldsymbol{G}$ exceeds what our computer could offer. In addition, the entire matrix error is bounded from above by the maximal admissible block error, and hence the latter can be used as an effective measurement of the error. In Fig. 7*b*, we plot matrix–vector multiplication time of the proposed solver with respect to N. A linear complexity can be clearly observed. In Fig. 7*c*, we plot the memory complexity of the proposed solver, which again demonstrates a linear complexity.

To compare the performance of the proposed solver with an FMM-based solver, we simulate a $k \times k$ bus structure shown in Fig. 8 embedded in free space. The k in this structure varies from 4 to 16. The dimension of each bus is scaled to $1\text{ m} \times 1\text{ m} \times (2k+1)$ m, and the distance between the two bus layers is 1 m. Two methods are compared: FastCap2.0 [23] and the proposed iterative IE solver. The convergence tolerances of both solvers are set to 1%. The simulation parameters of the proposed solver are chosen as leafsize = 8, $\eta = 1.2$ and $p = 2$. Fig. 9*a* shows the accuracy of both FastCap2.0 and the proposed solver with respect to the number of unknowns. The capacitance

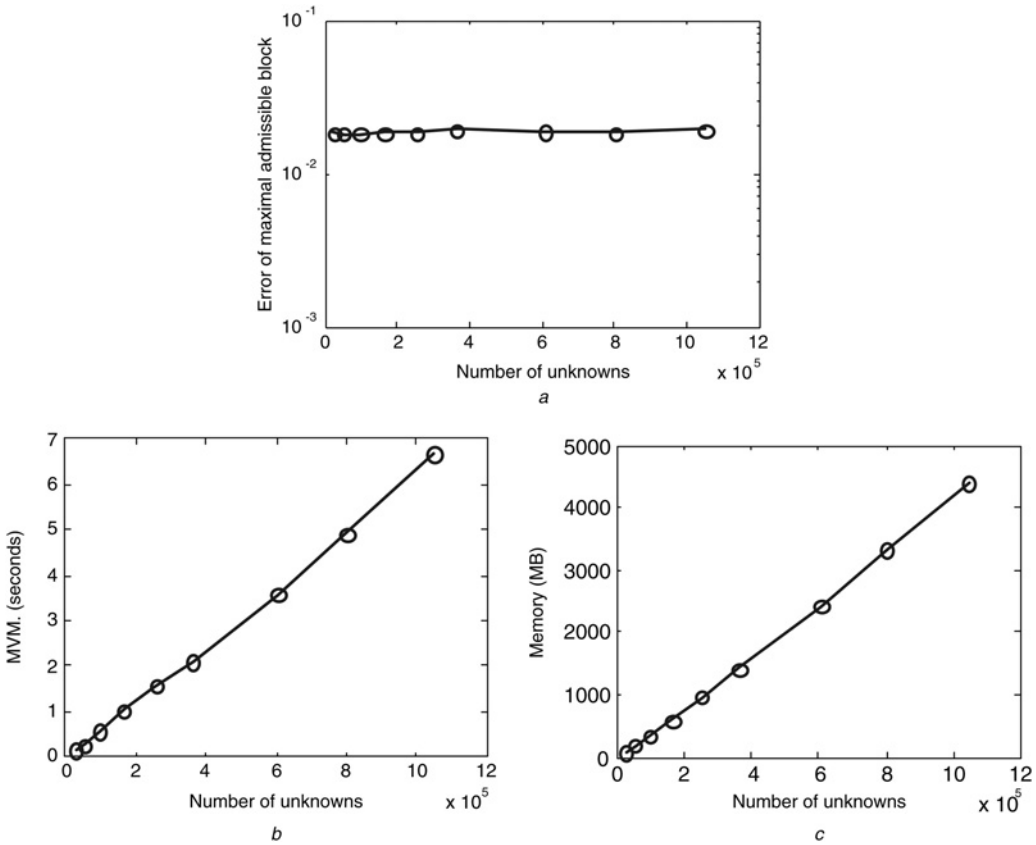

Figure 7 *Simulation of a large-scale on-chip interconnect*

a Maximal admissible block error with respect to the number of unknowns
b Time complexity
c Memory complexity

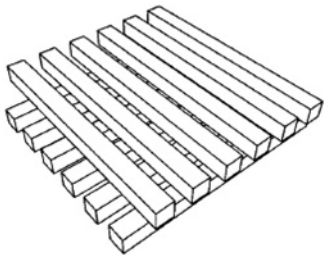

Figure 8 *A k × k bus structure*

error is measured by $\|\boldsymbol{C}-\boldsymbol{C}'\|_{\mathrm{F}}/\|\boldsymbol{C}\|_{\mathrm{F}}$, where $\boldsymbol{C}$ is the capacitance matrix obtained from a full-matrix-based direct solver, and $\boldsymbol{C}'$ is that generated by the proposed solver and FastCap2.0. Excellent accuracy can be observed for both solvers as can be seen from Fig. 9*a*, and the proposed solver is shown to have a better accuracy. In Fig. 9*b*, we plot the total solution time. As can be seen clearly, the time complexity of the proposed solver is linear. In addition, the total solution time of the proposed solver is much less than that of the FMM-based FastCap2.0.

Next, we tested the accuracy and efficiency of the linear-time $\mathcal{H}^2$-matrix-based IE solver for solving large-scale electrodynamic problems. A half sphere from small electric size to 60 wavelengths was simulated. There are only four simulation parameters to choose in the $\mathcal{H}^2$-matrix-based IE solver: η, leafsize, a and b. They were chosen as $\eta = 0.8$, leafsize $= 63$, $a = 4$ and $b = 1$. In Table 1, the error of the $\mathcal{H}^2$-matrix-based $\tilde{\boldsymbol{G}}$ was listed with respect to the electric size of the half sphere from three wavelengths to 17 wavelengths. A constant order of accuracy can be observed. In Table 1, we also listed the maximal admissible block error, which is defined as $\|\boldsymbol{G}^{(t,s)}-\tilde{\boldsymbol{G}}^{(t,s)}\|/(\max(\|\boldsymbol{G}^{(t,t)}\|, \|\boldsymbol{G}^{(s,s)}\|)$. Clearly, the entire matrix error is bounded from above by the maximal admissible block error as shown in Table 1. Therefore the maximal admissible block error can be used as an effective measurement of the error when the number of unknowns is so large that it is not feasible to store the original matrix $\boldsymbol{G}$ for accuracy assessment. In Fig. 10, the memory and CPU time cost were plotted. Again, linear scaling is observed.

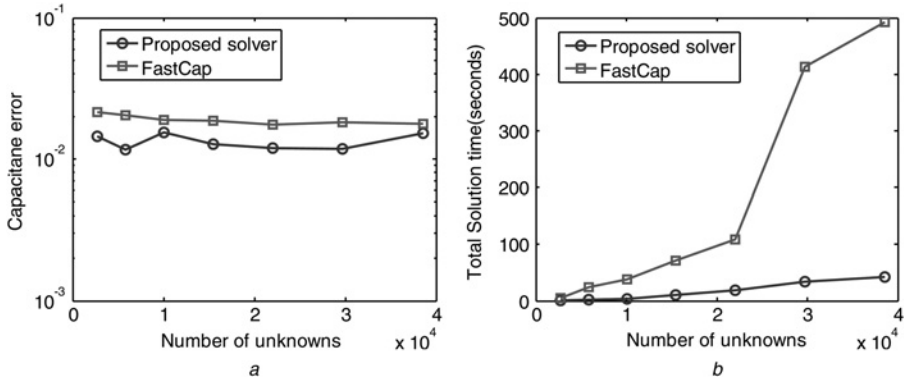

Figure 9 *Simulation of a k × k bus structure by the proposed solver and FMM-based FastCap2.0*

a Accuracy
b Total solution time

Table 1 Accuracy of the $\mathcal{H}^2$-matrix-based representation $\tilde{\boldsymbol{G}}$ with respect to electric size

half_sphere_size/λ	N	$\frac{\|\boldsymbol{G}-\tilde{\boldsymbol{G}}\|}{\|\boldsymbol{G}\|}$	$\frac{\|\boldsymbol{G}^{(t,s)}-\tilde{\boldsymbol{G}}^{(t,s)}\|}{\max(\|\boldsymbol{G}^{(t,t)}\|,\ \|\boldsymbol{G}^{(s,s)}\|)}$
3	2340	4.247538e − 04	3.170883e − 04
5	5310	2.965317e − 04	5.230145e − 04
7	8320	4.336523e − 04	4.799418e − 04
9	14 850	6.425261e − 04	1.096829e − 03
11	21 420	2.175265e − 04	6.108898e − 04
13	29 190	5.464219e − 04	9.608508e − 04
15	35 520	6.394084e − 04	1.034488e − 03
17	43 010	7.004305e − 04	8.451372e − 04

IET Microw. Antennas Propag., 2010, Vol. 4, Iss. 10, pp. 1583–1596

doi: 10.1049/iet-map.2009.0229

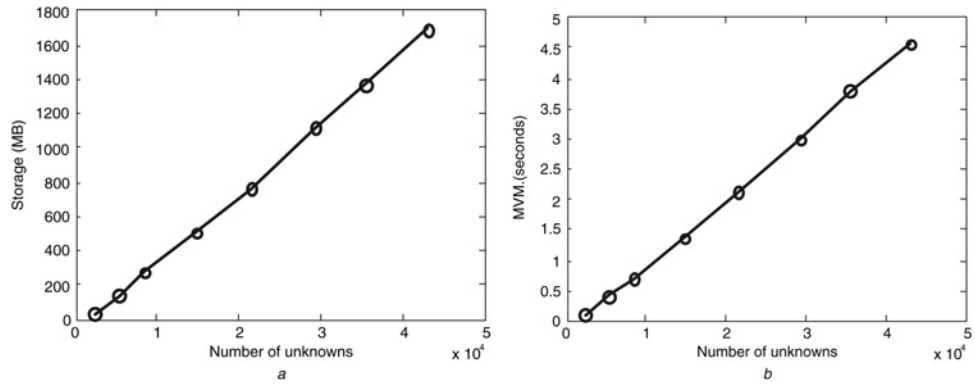

Figure 10 *Simulation of a half sphere*

a Memory consumption as a function of unknowns
b Time complexity

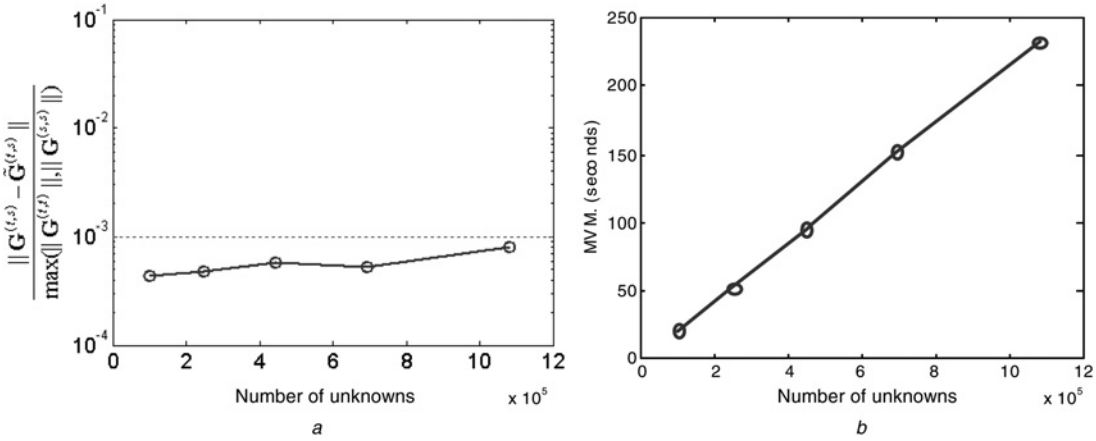

Figure 11 *Simulation of a half sphere to 60 wavelengths*

a $\mathcal{H}^2$-approximation error with respect to *N*
b Time complexity

Next, we enlarged the electric size of the half sphere to 60 wavelengths, resulting in 1 082 050 unknowns. In Fig. 11*a*, we plotted maximal admissible block error with respect to the electric size from 20 wavelengths to 60 wavelengths. Good accuracy is observed in the entire range. In Fig. 11*b*, we plotted the matrix-vector multiplication time versus the number of unknowns. A clear linear scaling is observed.

7 Conclusions

In this work, we demonstrated two linear-complexity IE-based methods. One is a centre-point $\mathcal{H}$-matrix-based method of linear complexity for large-scale analysis of static problems or problems having small electric sizes. The other is an $\mathcal{H}^2$-matrix-based method of controlled accuracy and linear complexity for large-scale analysis of electrodynamic problems across a wide range of electric sizes. Numerical simulations from a small number of unknowns to over 1 million unknowns, from small electric sizes to over 60 wavelengths have demonstrated the accuracy and efficiency of the proposed methods.

8 Acknowledgment

This work was supported by NSF under award No. 0747578 and No. 0702567.

9 References

[1] CHEW W.C., JIN J.M., MICHIELSSEN E., SONG J.M. (EDS.): 'Fast and efficient algorithms in computational electromagnetics' (Artech House, Norwood, MA, 2001)

[2] BOJARSKI N.N.: '*k*-space formulation of the electromagnetic scattering problem'. Technical Report AFAL-TR-71–75, Air Force Avionics Lab., March 1971

[3] BLESZYNSKI E., BLESZYNSKI M., JAROZEWICZ T.: 'AIM: adaptive integral method for solving large-scale electromagnetic

scattering and radiation problems', *Radio Sci.*, 1996, **31**, pp. 1225–1251

[4] KAPUR S., LONG D.E.: 'IES3: a fast integral equation solver for efficient 3-dimensional extraction'. IEEE/ACM Int. Conf. Comput.-Aided Design Dig., November 1997, pp. 448–455

[5] SEO M., LEE J.F.: 'A single-level low rank IE-QR algorithm for PEC scattering problems using EFIE formulation', *IEEE Trans. Antennas Propag.*, 2004, **52**, (8), pp. 2141–2146

[6] GOPE D., JANDHYALA V.: 'Efficient solution of EFIE via low-rank compression of multilevel predetermined interactions', *IEEE Trans. Antennas Propag.*, 2005, **53**, (10), pp. 3324–3333

[7] HACKBUSCH W., KHOROMSKIJ B.: 'A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: introduction to $\mathcal{H}$-matrices', *Computing*, 1999, **62**, pp. 89–108

[8] HACKBUSCH W., KHOROMSKIJ B.N.: 'A sparse $\mathcal{H}$-matrix arithmetic. Part II: application to multi-dimensional problems', *Computing*, 2000, **64**, pp. 21–47

[9] BORM S.: 'Introduction to hierarchical matrices with applications', *EABE*, 2003, **27**, pp. 403–564

[10] HACKBUSCH W., KHOROMSKIJ B., SAUTER S.: 'On $\mathcal{H}^2$-matrices', in BUNGARTZ H., HOPPE R., ZENGER C. (EDS.): 'Lecture on applied mathematics', Springer-Verlag, Berlin, 2000, pp. 9–29

[11] BORM S.: 'Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices', *Computing*, 2002, **69**, pp. 1–35

[12] BORM S.: '$\mathcal{H}^2$-matrices – multilevel methods for the approximation of integral operators', *Comput. Visual. Sci.*, 2004, **7**, pp. 173–181

[13] CHAI W., JIAO D.: 'An $\mathcal{H}$-matrix-based method for reducing the complexity of integral-equation-based solutions of electromagnetic problems'. IEEE Int. Symp. on Antennas and Propagation, June 2008, p. 4

[14] CHAI W., JIAO D.: 'An $\mathcal{H}^2$-matrix-based integral-equation solver of linear-complexity for large-scale electromagnetic analysis'. 2008 Asia Pacific Microwave Conf., December 2008, p. 4

[15] CHAI W., JIAO D.: 'An $\mathcal{H}^2$-matrix-based integral-equation solver of reduced complexity and controlled accuracy for solving electrodynamic problems', *IEEE Trans. Antennas Propag.*, 2009, **57**, (10), pp. 3147–3159

[16] YAN S., SAREN V., SHI W.: 'Sparse transformations and preconditioners for hierarchical 3-D capacitance extraction with multiple dielectrics'. ACM/EDAC/IEEE Design Automation Conf. (DAC), 2004, pp. 788–793

[17] RAO S.M., WILTON D.R.: 'Electromagnetic scattering by surfaces of arbitrary shape', *IEEE Trans. Antennas Propag.*, 1982, **30**, (3), pp. 409–418

[18] BORM S., GRASEDYCK L., HACKBUSCH W.: 'Hierarchical matrices' (Lecture note 21 of the Max Planck Institute for Mathematics in the Sciences, 2003)

[19] BORM S.: '$\mathcal{H}^2$-matrix approximation of integral operators by interpolation', *Appl. Numer. Math.*, 2002, **43**, pp. 129–143

[20] GRASEDYCK L.: 'Adaptive geometrically balanced clustering of $\mathcal{H}$-matrices', *Computing*, 2004, **73**, pp. 1–23

[21] BORM S., GRASEDYCK L.: 'Low-rank approximation of integral operators by interpolation', *Computing*, 2004, **72**, pp. 325–332

[22] WANG H.G., CHAN C.H.: 'The implementation of multilevel Green's function interpolation method for full-wave electromagnetic problems', *IEEE Trans. Antennas Propag.*, 2007, **55**, (5), pp. 1348–1358

[23] NABORS K., WHITE J.: 'FastCap: A multipole accelerated 3-d capacitance extraction program', *IEEE Trans. CAD*, 1991, **10**, pp. 1447–1459

doi: 10.1049/iet-map.2009.0229