

A Scalable Parallel PWTD-Accelerated SIE Solver for Analyzing Transient Scattering From Electrically Large Objects

Yang Liu, *Member, IEEE*, Abdulkadir C. Yücel, Hakan Bağcı, *Senior Member, IEEE*,
and Eric Michielssen, *Fellow, IEEE*

Abstract—A scalable parallel plane-wave time-domain (PWTD) algorithm for efficient and accurate analysis of transient scattering from electrically large objects is presented. The algorithm produces scalable communication patterns on very large numbers of processors by leveraging two mechanisms: 1) a hierarchical parallelization strategy to evenly distribute the computation and memory loads at all levels of the PWTD tree among processors and 2) a novel asynchronous communication scheme to reduce the cost and memory requirement of the communications between the processors. The efficiency and accuracy of the algorithm are demonstrated through its applications to the analysis of transient scattering from a perfect electrically conducting (PEC) sphere with a diameter of 70 wavelengths and a PEC square plate with a dimension of 160 wavelengths. Furthermore, the proposed algorithm is used to analyze transient fields scattered from realistic airplane and helicopter models under high frequency excitation.

Index Terms—Fast algorithms, marching-on-in-time (MOT), parallelization, plane-wave time-domain (PWTD) algorithm, time-domain surface integral equation (TD-SIE), transient scattering, very large-scale problems.

I. INTRODUCTION

TRANSIENT electromagnetic scattering from complex objects involving perfect electrically conducting (PEC) surfaces and piecewise inhomogeneous dielectric volumes can be analyzed by solving time-domain surface integral equations (TD-SIEs) [1]. Development of multilevel plane-wave time-domain (PWTD) algorithm has enabled marching-on-in-time (MOT) scheme a viable method for efficiently solving TD-SIEs [2]. The resulting PWTD-accelerated MOT-TD-SIE solvers have been shown to successfully compete with finite difference time-domain schemes in analyzing transient fields

scattered from electrically large (i.e., many wavelengths long) real-life objects [3], [4].

The PWTD-accelerated MOT-TD-SIE solvers are the time-domain counterparts of multilevel fast multipole (MLFMA)-accelerated frequency-domain SIE solvers. Their computational cost and memory requirement scale as $O(N_t N_s \log^2 N_s)$ and $O(N_s^{1.5})$, respectively. Here, N_s is the number of spatial unknowns and $N_t = O(N_s^{0.5})$ is the number of time steps. A comparison of these scaling estimates to those for the cost and memory requirements of unaccelerated MOT-TD-SIE solvers, which scale as $O(N_t N_s^2)$ and $O(N_s^2)$, respectively, demonstrates the benefits of multilevel PWTD acceleration. Having said that, the applicability of PWTD-accelerated MOT-TD-SIE solver in analyzing transient scattering from electrically large real-life objects can further be increased through effective parallelization schemes. Unfortunately, parallelizing multilevel PWTD algorithm on distributed memory clusters is a challenging task due to its complex nature and heterogeneous structure. Until recently, similar difficulties prevented efficient and scalable parallelization of MLFMA-accelerated frequency-domain SIE solvers [5].

Previously, schemes, which make use of spatial [6] and hybrid spatial/angular partitioning [7] strategies to efficiently parallelize the multilevel PWTD algorithm, were proposed. The PWTD algorithm with hybrid partitioning was incorporated into a TD-SIE solver and used for analyzing scattering from objects discretized with half a million spatial unknowns [7]. Unfortunately, neither of these parallelization schemes scales well on distributed memory clusters with large numbers of processors since their computational and communication costs and memory requirements per processor are not inversely proportional to the total processor count. This is expected since they make use of straight-forward time-domain extensions of spatial [8] and hybrid spatial/angular [9] partitioning schemes that were originally developed for parallelizing the MLFMA and are known to not scale well. The first provably scalable techniques for parallelizing the MLFMA did not appear in the archival literature until 2008 [10]–[14]. These techniques make use of a hierarchical partitioning strategy that simultaneously leverages spatial and angular partitioning at each level of the MLFMA tree [10]–[14].

Unfortunately, the direct extension of these MLFMA parallelization strategies to time domain does not produce a scalable parallel PWTD algorithm. This is simply because of

Manuscript received January 16, 2015; revised November 04, 2015; accepted December 07, 2015. Date of publication December 17, 2015; date of current version February 01, 2016. This work was supported in part by the AFOSR/NSSEFF Program under Grant FA9550-10-1-0180, and in part by the National Science Foundation (NSF) under Grant CCF 1116082.

Y. Liu, A. C. Yücel and E. Michielssen are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: liuyangz@umich.edu; acyucel@umich.edu; emichiel@umich.edu).

H. Bağcı is with the Division of Computer, Electrical, and Mathematical Sciences and Engineering and Center for Uncertainty Quantification in Computational Science and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia (e-mail: hakan.bagci@kaust.edu.sa).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAP.2015.2508483

the two important differences between MLFMA and PWTD: 1) MLFMAs call for only spatial and angular discretizations, while PWTD schemes require spatial, angular, and temporal discretizations. This results in different CPU and memory requirements for PWTD schemes. More specifically, from a parallelization perspective, communication costs in the translation stage of the PWTD schemes scale significantly different from those of the MLFMAs. 2) MLFMAs often use local (in angular dimension) schemes to interpolate and filter/interpolate far-fields, while PWTD implementations, for reasons relating to stability of the TD-SIE solvers, usually rely on exact global (in angular dimension) spherical interpolation/filtering schemes [4]. Load balancing these global schemes poses challenges, especially at the coarse PWTD levels involving many plane wave directions.

These differences between MLFMA and PWTD listed above clearly motivate the formulation and implementation of a scalable scheme to parallelize the multilevel PWTD algorithm. This paper describes such a scheme that makes use of a hierarchical parallelization strategy to quasi-optimally distribute computation and memory loads pertinent to spatial, angular, and temporal dimensions among processors. In addition, a novel asynchronous communication technique for reducing the cost and memory requirements of the communications at the translation stage of the PWTD algorithm is developed. By combining the hierarchical partitioning strategy and this asynchronous communication technique to achieve CPU and memory load balancing among processors, the proposed scheme produces scalable communication patterns among processors at all levels of the PWTD tree. The resulting parallel PWTD algorithm is incorporated into a TD-SIE solver to enable efficient and accurate analysis of transient scattering from electrically large PEC objects. Indeed, numerical results demonstrate that the proposed parallel PWTD-accelerated TD-SIE solver can be efficiently applied to real-life electromagnetics problems involving scatterers spanning over a 100 wavelengths and discretized with 10 million spatial unknowns.

II. FORMULATION

A. TD-SIEs and MOT Scheme

Let S denote an arbitrarily shaped (closed) PEC surface that resides in free space with permittivity ε_0 and permeability μ_0 . S is excited by the incident electromagnetic field $\{\mathbf{E}^i(\mathbf{r}, t), \mathbf{H}^i(\mathbf{r}, t)\}$. $\mathbf{E}^i(\mathbf{r}, t)$ and $\mathbf{H}^i(\mathbf{r}, t)$ are assumed essentially bandlimited to maximum frequency f_{max} and vanishingly small $\forall \mathbf{r} \in S$ for $t < 0$. In response to the excitation, the surface current density $\mathbf{J}(\mathbf{r}, t)$ is induced on S and $\mathbf{J}(\mathbf{r}, t)$ generates the scattered field $\{\mathbf{E}^s(\mathbf{r}, t), \mathbf{H}^s(\mathbf{r}, t)\}$. $\mathbf{J}(\mathbf{r}, t)$ satisfies the time-domain electric field and magnetic field integral equations (TD-EFIE and TD-MFIE) [15]

$$\begin{aligned} \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \partial_t \mathbf{E}^i(\mathbf{r}, t) &= -\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \partial_t \mathbf{E}^s(\mathbf{r}, t) \\ &= \mathcal{L}_e[\mathbf{J}](\mathbf{r}, t) \quad \forall \mathbf{r} \in S, S^+, S^- \end{aligned} \quad (1)$$

$$\begin{aligned} \hat{\mathbf{n}} \times \partial_t \mathbf{H}^i(\mathbf{r}, t) &= -\hat{\mathbf{n}} \times \partial_t \mathbf{H}^s(\mathbf{r}, t) \\ &= \mathcal{L}_h[\mathbf{J}](\mathbf{r}, t) \quad \forall \mathbf{r} \in S^-. \end{aligned} \quad (2)$$

Here, S^- and S^+ denote surfaces conformal to but just inside and outside S , $\hat{\mathbf{n}}$ denotes the outward pointing unit normal to S , and ∂_t represents the time derivative. The TD-EFIE and TD-MFIE operators \mathcal{L}_e and \mathcal{L}_h are

$$\mathcal{L}_e[\mathbf{J}](\mathbf{r}, t) = \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \frac{\mu_0}{4\pi} \int_S dS' (\partial_t^2 \mathcal{I} - c_0^2 \nabla \nabla) \cdot \frac{\mathbf{J}(\mathbf{r}', \tau)}{R} \quad (3)$$

$$\begin{aligned} \mathcal{L}_h[\mathbf{J}](\mathbf{r}, t) &= \frac{1}{4\pi} \hat{\mathbf{n}} \times \int_S dS' (\mathbf{r} - \mathbf{r}') \\ &\times \left[\frac{1}{c_0 R^2} \partial_t^2 \mathbf{J}(\mathbf{r}', \tau) + \frac{1}{R^3} \partial_t \mathbf{J}(\mathbf{r}', \tau) \right] \end{aligned} \quad (4)$$

where \mathcal{I} is the identity dyad, $R = |\mathbf{r} - \mathbf{r}'|$ is the distance between source point \mathbf{r}' and observer point \mathbf{r} , $c_0 = 1/\sqrt{\varepsilon_0 \mu_0}$ is the speed of light in free space, and $\tau = t - R/c_0$ denotes retarded time. The TD-EFIE applies to both open and closed surfaces [1]. When applied to closed surfaces, solutions to both the TD-EFIE and TD-MFIE are corrupted by oscillating currents that (approximately) reside in the null spaces of \mathcal{L}_e and \mathcal{L}_h [15], [16]. The time domain combined field integral equation (TD-CFIE) eliminates these spurious currents by linearly combining the TD-EFIE and TD-MFIE as

$$\begin{aligned} \hat{\mathbf{n}} \times \partial_t \mathbf{H}^i(\mathbf{r}, t) - \beta/\eta_0 \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \partial_t \mathbf{E}^i(\mathbf{r}, t) \\ = \mathcal{L}_h[\mathbf{J}](\mathbf{r}, t) - \beta/\eta_0 \mathcal{L}_e[\mathbf{J}](\mathbf{r}, t) \\ = \mathcal{L}_c[\mathbf{J}](\mathbf{r}, t) \quad \forall \mathbf{r} \in S^-. \end{aligned} \quad (5)$$

Here, $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$ is the wave impedance in free space. The TD-CFIE reduces to the TD-EFIE and TD-MFIE when $\beta = \infty$ and $\beta = 0$, respectively.

To numerically solve (5), $\mathbf{J}(\mathbf{r}, t)$ is discretized using spatial basis functions $\mathbf{S}_n(\mathbf{r})$, $n = 1, \dots, N_s$, and temporal basis functions $T_i(t)$, $i = 1, \dots, N_t$

$$\mathbf{J}(\mathbf{r}, t) = \sum_{n=1}^{N_s} \mathbf{S}_n(\mathbf{r}) f_n(t) = \sum_{n=1}^{N_s} \sum_{i=1}^{N_t} I_{n,i} \mathbf{S}_n(\mathbf{r}) T_i(t). \quad (6)$$

Here, $f_n(t)$ is the temporal signature associated with $\mathbf{S}_n(\mathbf{r})$ and $I_{n,i}$ is the expansion coefficient associated with space-time basis function $\mathbf{S}_n(\mathbf{r}) T_i(t)$. The temporal basis function $T_i(t) = T(t - i\Delta t)$ is a shifted Lagrange interpolant [17] that is nonzero for $t > -\Delta t$; $\Delta t = 1/(2\chi_t f_{max})$ is the time step size and $\chi_t > 1$ is the temporal oversampling factor. The spatial basis functions $\mathbf{S}_n(\mathbf{r})$ are Rao–Wilton–Glisson (RWG) functions [18]. Substituting (6) into (5) and testing the resulting equation with $\mathbf{S}_m(\mathbf{r})$, $m = 1, \dots, N_s$, at $t = j\Delta t$ yields the set of linear equations

$$\bar{\bar{\mathbf{Z}}}_0 \bar{\mathbf{I}}_j = \bar{\mathbf{F}}_j - \sum_{i=1}^{j-1} \bar{\bar{\mathbf{Z}}}_i \bar{\mathbf{I}}_{j-i}. \quad (7)$$

Here, the entries of the vectors $\bar{\mathbf{I}}_j$ and $\bar{\mathbf{F}}_j$ and the matrices $\bar{\bar{\mathbf{Z}}}_i$ are

$$\{\bar{\mathbf{I}}_j\}_n = I_{n,j}, \quad n = 1, \dots, N_s \quad (8)$$

$$\{\bar{\mathbf{F}}_j\}_m = \langle \mathbf{S}_m(\mathbf{r}), \hat{\mathbf{n}} \times \partial_t \mathbf{H}^i(\mathbf{r}, t) - \beta/\eta_0 \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \partial_t \mathbf{E}^i(\mathbf{r}, t) \rangle|_{t=j\Delta t}, \quad m = 1, \dots, N_s \quad (9)$$

$$\{\bar{\mathbf{Z}}_i\}_{mn} = \langle \mathbf{S}_m(\mathbf{r}), \mathcal{L}_c[\mathbf{S}_n T_{-i}](\mathbf{r}, t) \rangle|_{t=0}, \quad m, n = 1, \dots, N_s \quad (10)$$

where $\langle \cdot, \cdot \rangle$ denotes the standard spatial inner product. The set of linear equations (7) can be solved by MOT: first, $\bar{\mathbf{I}}_1$ is computed by solving (7) for $j = 1$. Then, for $j = 2$, the summation on the right-hand side (RHS) of (7) is computed and resulting system is solved for $\bar{\mathbf{I}}_2$. This process is repeated to compute $\bar{\mathbf{I}}_3$ and so on. At every time step, an iterative solver is used to efficiently solve (7) for $\bar{\mathbf{I}}_j$. At a given time step, the computationally most demanding operation is the evaluation of the sum on the RHS of (7). This computation requires $O(N_t N_s^2)$ operations for all N_t time steps and $O(N_s^2)$ memory. Classical MOT schemes, therefore, are prohibitively expensive when applied to the analysis of transient scattering from electrically large structures. The multilevel PWTD algorithm significantly reduces the computational cost of the MOT scheme [4] and is summarized next.

B. Multilevel PWTD Algorithm

Consider a rectangular box enclosing S ; it is recursively subdivided into eight boxes until the dimensions of the smallest boxes are on the order of the wavelength at the maximum frequency, $\lambda = c_0/f_{max}$. This recursive subdivision strategy gives rise to an N_L -level PWTD tree with levels labeled $v = 1, \dots, N_L = O(\log(N_s^{0.5}))$. The tree's finest level ($v = 1$) contains the smallest boxes while its coarsest level ($v = N_L$) contains the box enclosing S . Let N_g^v denote the number of nonempty boxes at level v . For (nonfractal) surface scatterers $N_g^1 = O(N_s)$ and $N_g^{v+1} \approx N_g^v/4$. The radius of a sphere enclosing a level v box is $R^v = 2^{(v-1)}R^1$ with $R^1 = O(1)$.

Upon constructing the PWTD tree, far-field box pairs at each level are identified starting with level $N_L - 2$. Two nonempty boxes at level v are labeled a level- v far-field pair if the distance between their centers is greater than γR^v ($3 \leq \gamma \leq 6$) and their respective parent boxes do not constitute a far-field pair [2]. Two nonempty boxes at the finest level, which are not labeled as a far-field pair are considered as a near-field pair; also each nonempty box at the finest level forms a near-field pair with itself. Interactions between spatial basis functions residing in near-field box pairs are computed using (10) and their contributions are directly added to the RHS of (7). Interactions between spatial basis functions contained in far-field box pairs are evaluated by the PWTD scheme.

Let α and α' denote a far-field box pair's source and observer boxes. Let $R_{c,\alpha\alpha'} = |\mathbf{R}_{c,\alpha\alpha'}| = |\mathbf{r}_o^c - \mathbf{r}_s^c|$ denote the distance between the source and observer box centers \mathbf{r}_s^c and \mathbf{r}_o^c . The source and observer boxes contain spatial basis functions $\mathbf{S}_n(\mathbf{r})$, $n \in \alpha$, and $\mathbf{S}_m(\mathbf{r})$, $m \in \alpha'$, respectively. For $\forall n \in \alpha$, the temporal signature $f_n(t)$ associated with $\mathbf{S}_n(\mathbf{r})$ is broken into N_l^v consecutive subsignals, $f_n^l(t)$, $l = 1, \dots, N_l^v$, using bandlimited local interpolants [19]. The subsignal $f_n^l(t)$ is bandlimited to $f_s = \chi_t f_{max}$ and has quasi-finite duration

$T^v \Delta t \leq (R_{c,\alpha\alpha'} - 2R^v)/c_0$, with $T^v N_l^v \approx N_t$. Fields due to $\mathbf{S}_n(\mathbf{r}) f_n^l(t)$, $n \in \alpha$ tested by $\mathbf{S}_m(\mathbf{r})$, $m \in \alpha'$ are expressed as

$$\begin{aligned} & \langle \mathbf{S}_m(\mathbf{r}), \mathcal{L}_c[\mathbf{S}_n f_n^l](\mathbf{r}, t) \rangle \\ &= \frac{1}{8\pi^2 c_0^2} \sum_{q=0}^{K^v} \sum_{p=-K^v}^{K^v} \omega_{qp}^v [-\beta \mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v) \\ & \quad + \mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{n}})]^\dagger * \mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t) * \mathbf{P}_n^+(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v) * f_n^l(t) \end{aligned} \quad (11)$$

where $K^v = \lfloor 4\pi f_s \chi_s R^v / c_0 \rfloor + 1$ represents the number of spherical harmonics effectively accounted for in the plane wave expansions, χ_s represents the spherical oversampling factor, ω_{qp}^v are quadrature weights on the unit sphere $\hat{\mathbf{k}}_{qp}^v$, $q = 0, \dots, K^v$ and $p = -K^v, \dots, K^v$ represent directions of outgoing/incoming rays with a total of $N_k^v = (K^v + 1)(2K^v + 1)$ directions [2], \dagger denotes the transpose, and $*$ represents time convolution. The projection function $\mathbf{P}_{\{m,n\}}^\pm(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{v}})$ is

$$\begin{aligned} \mathbf{P}_{\{m,n\}}^\pm(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{v}}) &= \int_{S_{\{m,n\}}} dS' \hat{\mathbf{v}} \\ & \quad \times \mathbf{S}_{\{m,n\}}(\mathbf{r}') \delta(t \pm \hat{\mathbf{k}}_{qp}^v \cdot (\mathbf{r}' - \mathbf{r}_{\{o,s\}}^c) / c_0). \end{aligned} \quad (12)$$

The translation function $\mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t)$ is

$$\begin{aligned} \mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t) &= \frac{c_0 \partial_t^3}{2R_{c,\alpha\alpha'}} \sum_{k=0}^{K^v} (2k+1) \Phi_k \left(\frac{c_0 t}{R_{c,\alpha\alpha'}} \right) \times \Phi_k \\ & \quad \left(\frac{\hat{\mathbf{k}}_{qp}^v \cdot \mathbf{R}_{c,\alpha\alpha'}}{R_{c,\alpha\alpha'}} \right) \end{aligned} \quad (13)$$

where $\Phi_k(\cdot)$ is the Legendre polynomial of degree k and $|t| \leq R_{c,\alpha\alpha'}/c_0$.

The PWTD algorithm is executed as follows. First, outgoing rays for all directions $\hat{\mathbf{k}}_{qp}^v$ are constructed by convolving the projection function $\mathbf{P}_n^+(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v)$ with the subsignal $f_n^l(t)$. Next, outgoing rays in box α are convolved with $\mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t)$ and are translated into incoming rays in box α' . Finally, the incoming rays are projected onto testing basis function $\mathbf{S}_m(\mathbf{r})$ by convolving the projection function $\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v)$ and $\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{n}})$ with the incoming rays and summing over all directions with quadrature weights ω_{qp}^v [2]. Note that only outgoing/incoming rays of the finest level boxes are constructed/projected directly from/onto basis functions using the projection function (12); those of higher level boxes are constructed/projected by an exact global vector spherical filtering technique described in [4]. The analysis in [4] showed that the computational cost and memory requirements of a multilevel PWTD-accelerated MOT scheme applied to surface scatterers scale as $O(N_t N_s \log^2 N_s)$ and $O(N_s^{1.5})$, respectively.

C. Parallelization of the PWTD-Accelerated TD-SIE Solver

This section describes a highly scalable scheme for parallelizing the multilevel PWTD algorithm briefly described in the previous section. The proposed strategy leverages hierarchical

partitioning of the multilevel PWTD tree among processors and an asynchronous scheme for memory and cost efficient communications between processors. This asynchronous scheme is implemented using message passing interface (MPI). In what follows, first, an overview of the proposed parallelization strategy is provided (Section II-C1). Then, the costs estimates for computations and communications required at different stages of the PWTD algorithm are derived (Sections II-C2–II-C4). Finally, overall computational and communication costs of the proposed parallelization strategy are provided and its scalability is theoretically proven (Section II-C5).

1) *A Pedestrian Description:* The effective parallelization of the multilevel PWTD scheme calls for a uniform distribution of the near-field MOT matrix elements (i.e., near-field data) and outgoing/incoming rays (i.e., ray data) and the pertinent workload among processors. The near-field data can be uniformly distributed among processors in a straightforward manner. That said, distributing the ray data uniformly among processors is a challenging task due to the PWTD algorithm's heterogeneous tree structure. That is, ray data at level v of the PWTD tree are computed for $N_g^v = O(N_s/4^v)$ boxes (or spatial samples), $N_k^v = O(4^v)$ angular samples and approximately $T^v = O(2^v)$ temporal samples at each PWTD stage and its partitioning along a single dimension results in poor load balance and/or congested communications at certain levels. This problem is observed with spatial partitioning at higher levels and angular or temporal partitioning at lower levels. A viable solution to this problem is adaptively partitioning ray data along more than one dimension.

To achieve this, the proposed scheme first identifies a “base” level v_b at which the number of boxes $N_g^{v_b}$ is no less than the number of processors N_p . For simplicity, v_b is chosen as the highest possible such level. It should be noted here that the proposed partitioning strategy can be easily adopted for other choices of v_b . Then, it uses two different partitioning strategies for levels $v \leq v_b$ and $v > v_b$.

- 1) At levels $v \leq v_b$, each processor computes and stores the complete ray and near-field data for approximately N_g^v/N_p boxes. Moreover, each processor stores the geometry information related to spatial basis functions $\mathbf{S}_n(\mathbf{r})$ in its N_g^1/N_p boxes and their near-field pairs at level $v = 1$. By doing so, the memory and computation loads are only partitioned along the spatial dimension.
- 2) At levels $v > v_b$, computation and storage of the ray data of one box are distributed among $N_r^v = \lfloor N_p/N_g^v \rfloor$ processors. Each processor is in charge of storing N_k^v/N_r^v angular samples of one box's ray data, hence the memory load is simultaneously partitioned along the spatial and angular dimensions. This memory partitioning leads to the following workload partitioning: each processor performs the translation operation for N_k^v/N_r^v angular samples and all, temporal samples of the ray data of one box; in contrast, each processor spherically interpolates/filters the ray data for $O(T^v/N_r^v)$ temporal samples and all angular samples of the ray data of one box. This approach ensures that computation load is simultaneously partitioned along the spatial and angular/temporal dimensions.

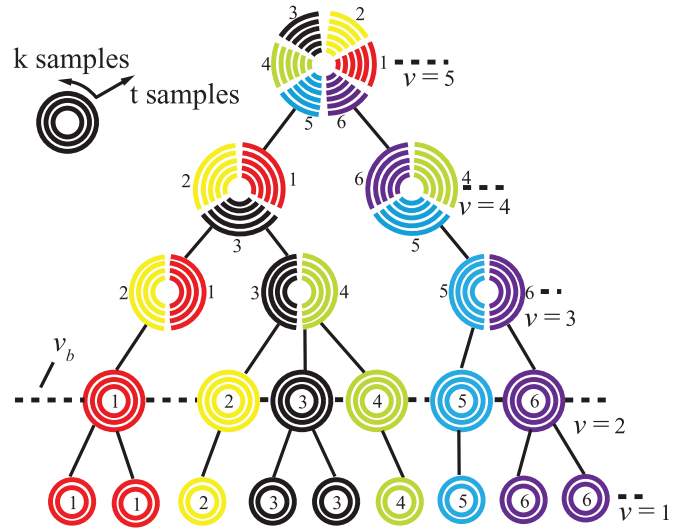


Fig. 1. Partitioning of boxes and their ray data in a five-level PWTD tree among six processors.

This partitioning strategy is perhaps best described by an example. Consider a five-level PWTD tree that is partitioned among six processors (Fig. 1). In Fig. 1, each set of concentric circles represents one box and its associated ray data. The angular and radial dimensions of the circles concern the angular and temporal samples of the ray data, respectively. The number printed near the concentric circles and arcs indicates the ID of the processor in charge of the data marked with a certain color. For this example, $N_g^v = 9, 6, 3, 2, 1$ for $v = 1, \dots, 5$, and $N_p = 6$, therefore, $v_b = 2$.

First, the proposed strategy assigns each box at base level $v = v_b$ and with its corresponding subtree(s) to one processor. Each processor is responsible for computing and storing the ray data of the source/observer boxes at levels lower than the base level, i.e., $v \leq v_b$, which it is in charge of. In the example in Fig. 1, processor 1 is in charge of computing and storing the ray data of the leftmost two boxes at the first level and those of the single leftmost box at the second level. More specifically, processor 1 constructs/projects the ray data at the second and first levels by spherical interpolation/filtering of the ray data at the first and second level. Since the ray data at both levels are stored in processor 1, no interprocessor communication is needed. Each processor performs the translation stage without interprocessor communications if both source and observer boxes are handled by the same processor. Otherwise, it carries out the translation operation after receiving the outgoing ray data from other processors. Similarly, each processor is in charge of computing and storing the near-field data pertinent to source boxes at the finest level in its corresponding subtree. For example, in Fig. 1, processor 1 only computes and stores the near-field data pertinent to the leftmost two source boxes at the finest level. This near-field data are related to self and mutual interactions between the leftmost two source boxes and the mutual interaction between the second box (source box) and the third box (observer box) from the left. Note that in this example it is assumed that only adjacent (and self) boxes constitute near-field pairs. Since the near-field data pertinent to many box

pairs resides on the same processor, the communication cost for the near-field calculation is very low.

Second, the proposed strategy partitions the ray data of the boxes at levels higher than the base level, i.e., $v > v_b$, among processors by considering the number of processors and the number of boxes at that level. For example, in Fig. 1, the processor 1 is responsible for computing and storing $1/2$, $1/3$, and $1/6$ of the ray data (i.e., $N_r^v = 2, 3, 6$) of the leftmost boxes at levels 3, 4, and 5, respectively. Translation of the ray data of one box is split among N_r^v processors in angular dimension. For example, processor 1 performs translation stage for half of the angular samples of the ray data of the leftmost box at level 3 after receiving from processor 6 the outgoing ray data of the rightmost box at that level. Note that in this example, it is assumed that, at level 3, only the leftmost and rightmost boxes constitute a far-field pair. In contrast, interpolating/filtering the ray data of one box is split among N_r^v processors in temporal dimension. For example, interpolating/filtering the ray data of the leftmost box at level 4 is carried out by processor 1, 2, and 3; the ray data are redistributed among these three processors in temporal dimension, each processor interpolates/filters $1/3$ of the temporal samples of that box.

The proposed strategy gives rise to quasi-optimal (uniform) distribution of the ray data of boxes at each level of the PWTD tree as well as the near-field data among processors. Consequently, the memory requirement of PWTD-accelerated TD-SIE solver per processor scales as $O(N_s^{1.5}/N_p)$. In addition, this partitioning strategy produces a uniform distribution of the computation load among processors at each level of the PWTD tree. However, the sizes of communication packets, which scale as $O(N_k^v T^v / N_r^v) = O(2^v N_s / N_p)$, can be rather large at higher levels. A memory efficient and asynchronous communication scheme is, therefore, used to alleviate this high cost helping the proposed partitioning strategy produce scalable communication patterns. In the following sections, the computational and communication costs per processor at different stages of the multilevel PWTD algorithm are analyzed in detail.

2) Construction/Projection of Outgoing/Incoming Rays: The proposed strategy constructs the outgoing rays and projects the incoming rays separately for the finest level ($v = 1$) and the other levels ($v > 1$). Here, we only detail the procedures for constructing outgoing rays as those for the incoming rays are similar.

At level $v = 1$, construction of outgoing rays proceeds directly from basis function data and requires no communication. At this level, construction of one outgoing ray for one box requires $O(T^1)$ operations. The computation of all ray data for one box (N_l^1 outgoing rays for N_k^v directions) requires $O(T^1 N_l^1 N_k^1)$ operations. As one processor is in charge of approximately N_g^1 / N_p $v = 1$ boxes, the computational cost per processor for constructing outgoing rays at this level scales as $O(T^1 N_l^1 N_k^1 N_g^1 / N_p) = O(N_t N_s / N_p)$.

At levels $v > 1$, construction of outgoing rays is carried out using spherical interpolation and may require interprocessor communication. Unlike the ray data of $v = 1$ boxes, that of each $v > 1$ box may be constructed and stored by more than one processor (see the example in Fig. 1). Analysis of the computational and communication costs associated with the construction of outgoing rays via spherical interpolation

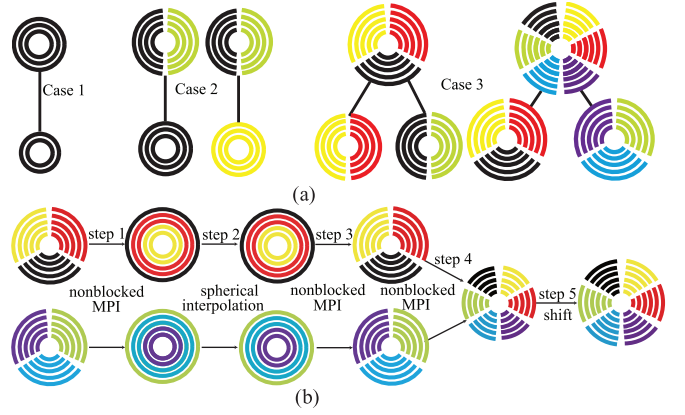


Fig. 2. (a) Three possible cases encountered during the construction of outgoing rays of boxes in PWTD tree. Each case requires different communication patterns. (b) Steps to construct the outgoing rays in case 3.

at levels $v > 1$ requires consideration of three distinct cases [Fig. 2(a)].

- 1) Case 1: Ray data of level $v \leq v_b$ boxes are directly interpolated and shifted from those of the child boxes, which the same the processor is in charge of. As the spherical interpolation of one temporal sample of ray data requires $O(N_k^v \log K^v)$ operations [4], [20], the computational cost of this operation scales as $O(T^v N_k^v \log K^v)$. This case requires no communication.
- 2) Case 2: Ray data of level $v = v_b + 1$ boxes are constructed by interpolating and shifting ray data of the child boxes, which are completely stored on another processor. The computational cost of this operation scales as $O(T^v N_k^v \log K^v)$. This case requires communication among processors following the interpolation step.
- 3) Case 3: Ray data of level $v > v_b + 1$ boxes are computed via interpolating and shifting the ray data of the child boxes, which invariably are stored on more than one processor. The construction of the outgoing rays [Fig. 2(b)] is performed in four steps. Step 1) Ray data of the child boxes stored in N_r^{v-1} processors is exchanged between them in such a way that each processor handles $O(T^v / N_r^{v-1})$ temporal samples of outgoing rays along all N_k^{v-1} directions [Fig. 2(b)]. Step 2) each processor performs its own spherical interpolation, requiring $O(T^v N_k^v \log K^v / N_r^{v-1})$ operations. Step 3) the interpolated ray data of each child box is split along the angular dimension and the resulting data is exchanged between N_r^{v-1} processors. Step 4) the interpolated ray data are sent to the processors in charge of the parent box via non-blocked MPI communication. Step 5) the transferred ray data are locally shifted to the center of the parent box.

For each processor, the computational cost CC_1 of constructing outgoing rays scales as

$$CC_1 = O\left(\frac{N_t N_s}{N_p}\right) + \sum_{v=2}^{v_b+1} \frac{N_g^{v-1}}{N_p} N_l^v O(T^v N_k^v \log K^v) + \sum_{v=v_b+2}^{N_L} N_l^v O\left(\frac{T^v N_k^v \log K^v}{N_r^{v-1}}\right) = O\left(\frac{N_t N_s \log^2 N_s}{N_p}\right). \quad (14)$$

In (14), three contributing terms represent the costs of constructing outgoing rays at the finest level, levels $2 < v \leq v_b + 1$ (cases 1 and 2), and levels $v > v_b + 1$ (case 3), respectively. Note that each processor is in charge of (completely or partially) interpolating ray data of approximately N_g^{v-1}/N_p child boxes for cases 1 and 2, and at most one child box for case 3.

The communication cost per processor is proportional to the total amount of data that it sends and receives. The cost of communications required at this stage of the PWT algorithm is dominated by the data exchange required in case 3 [depicted as Steps 1), 3), and 4) in Fig. 2(b)]. In Steps 1) and 3), each processor sends partial ray data of size $O(T^v N_k^v / N_r^{v-1})$ to $N_r^{v-1} - 1$ other processors and receives data of size $O(T^v N_k^v / (N_r^{v-1})^2)$ from each of the other $N_r^{v-1} - 1$ processors. Therefore, the total amount of data that one processor sends/receives in Steps 1) and 3) scales as $O(T^v N_k^v / N_r^{v-1})$. In Step 4), each processor sends/receives ray data in the amount of $O(T^v N_k^v / N_r^{v-1})$ to/from $O(1)$ other processors. Therefore, the communication cost CM_1 in this stage scales as

$$CM_1 = \sum_{v=v_b+2}^{N_L} N_l^v O\left(\frac{T^v N_k^v}{N_r^{v-1}}\right) = O\left(\frac{N_s N_t \log N_p}{N_p}\right). \quad (15)$$

Note that each term in the summation in (15) scales as $O(N_s N_t / N_p)$ and the total number of terms is $N_L - v_b - 1 = O(\log N_p)$.

The computational cost CC_2 associated with projecting incoming rays onto observers and the cost of communications required by this operation CM_2 are proportional to those associated with the construction of outgoing rays, i.e.

$$CC_2 \propto CC_1, CM_2 \propto CM_1. \quad (16)$$

3) Translation Stage: Translating one outgoing ray from a source box onto the incoming ray of an observer box requires the temporal convolution of the outgoing ray with the translation function in (13). This convolution is carried out in the Fourier domain, i.e., by inverse Fourier transforming the product of the ray's and translation function's Fourier transforms [2]. The computational cost associated with this operation scales as $O(T^v \log T^v)$. Note that for a level- v source box there exist N_l^v outgoing rays along each direction, and that each observer box interacts with $O(1)$ source boxes. At levels $v \leq v_b$, each processor performs translations along N_k^v directions for its N_g^v / N_p observer boxes; at levels $v > v_b$, each processor performs translations along N_k^v / N_r^v directions for one single observer box. The computational cost of the translation operation CC_3 for each processor can, therefore, be estimated as

$$\begin{aligned} CC_3 &= \sum_{v=1}^{v_b} \frac{N_g^v}{N_p} N_l^v O(1) N_k^v O(T^v \log T^v) \\ &\quad + \sum_{v=v_b+1}^{N_L} N_l^v O(1) \frac{N_k^v}{N_r^v} O(T^v \log T^v) \\ &= O\left(\frac{N_t N_s \log^2 N_s}{N_p}\right). \end{aligned} \quad (17)$$

Here, the first and second summations represent costs of translation operations at levels $v \leq v_b$ and $v > v_b$, respectively.

Note that during the translation stage outgoing ray data of a box residing in a processor is sent to at most $O(1)$ other processors. In fact, some translations do not require communications whatsoever since they involve source and observer boxes residing on the same processor. For each processor, the amount of ray data sent and received during the translation stage for one box scales as $O(N_k^v T^v)$ for levels $v \leq v_b$ and $O(N_k^v T^v / N_r^v)$ for levels $v > v_b$. Therefore, the communication cost for the translation stage CM_3 scales as

$$\begin{aligned} CM_3 &= \sum_{v=1}^{v_b} \frac{N_g^v}{N_p} N_l^v O(N_k^v T^v) + \sum_{v=v_b+1}^{N_L} N_l^v O\left(\frac{N_k^v T^v}{N_r^v}\right) \\ &= O\left(\frac{N_t N_s \log N_s}{N_p}\right). \end{aligned} \quad (18)$$

Oftentimes, the number of source boxes far-field paired with one observer box is large (e.g., exceeds one hundred) and the processor in charge of the observer box needs to allocate temporary memory for receiving all outgoing rays of source boxes it is interacting with. This temporary memory space may become excessively large, especially for translations at higher levels. To overcome this potential bottleneck, a novel, memory-efficient, and asynchronous communication scheme is proposed here. This scheme leverages concepts originally proposed in [21]. It no longer separates the computation and communication phases and limits the temporary memory to be allocated to that available to a single processor. The workflow of this scheme for one processor can be summarized as follows (Fig. 3). First, the processor allocates a “receiving” memory pool containing memory grains of size $O(T^v N_k^v)$ or $O(T^v N_k^v / N_r^v)$ to receive outgoing rays of one source box at level $v \leq v_b$ or $v > v_b$, respectively. Prior to translation, the processor sends out all/partial outgoing rays of source boxes needed by the far-field observer boxes that are held by different processors. The processor then iterates over the following four steps until the translation stage is complete. Step 1) The processor catches arriving data packets (i.e., outgoing ray data from source boxes). If a memory grain is available, the processor starts receiving the packet by putting it into the receiving queue. If not, it temporarily suspends reception of the packet until the next iteration. Step 2) The processor moves any completed packets in the receiving queue into a working queue, which now contains packets that are complete for translation. Step 3) The processor carries out the translation of complete outgoing rays. The working queue is a “priority queue” such that translations associated with “nonlocal” packets (as opposed to “local” packets that require no temporary memory space) and “higher level” packets that correspond to outgoing rays of boxes at higher levels, are executed first. Step 4) After translation, the memory grain associated with the packet is returned to the pool and becomes available again for Step 1). In this manner, the translation and communication are performed asynchronously and the maximum amount of temporary memory allocated is fully controlled by the processor.

4) Near-Field Calculation: Near-field calculations include: 1) matrix-vector multiplications on the left-hand side (LHS)

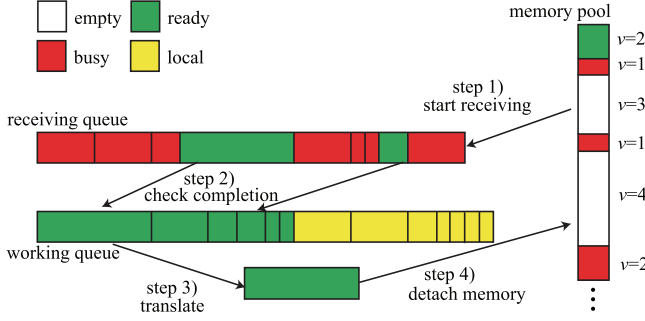


Fig. 3. Queue-based asynchronous communication during translation stage.

of (7) at each iteration of the solver executed for solving (7) at all time steps; and 2) partial matrix–vector multiplications on the RHS of (7). Here, it is assumed that the solution of (7) requires $O(1)$ iterations, a condition that typically is satisfied for nonresonant objects under high-frequency illumination. Since each box only participates in $O(1)$ near-field pairs, the near-field computational and communication costs per box scale as $O((N_s/N_g^1)^2) = O(1)$ and $O(N_s/N_g^1) = O(1)$, respectively. As one processor is in charge of approximately N_g^1/N_p source boxes at the finest level, the computational cost and communication cost of the near-field calculations CC_4 and CM_4 scale as

$$CC_4 \propto CM_4 = \frac{N_g^1}{N_p} N_t O(1) = O\left(\frac{N_s N_t}{N_p}\right). \quad (19)$$

5) *Total Computational and Communication Costs:* By summing the computational and communication costs in (14)–(19) and noting that CC_4 is bounded by $CC_{\{1,2,3\}}$ and $CM_{\{1,4\}}$ is bounded by $CM_{\{2,3\}}$, the total computational cost and total communication cost of the PWTD-accelerated TD-SIE solver CM and CC scale as

$$CC = O\left(\frac{N_t N_s \log^2 N_s}{N_p}\right), \quad CM = O\left(\frac{N_t N_s \log N_s}{N_p}\right). \quad (20)$$

These costs are inversely proportional to the number of processors N_p . As the memory requirements of the solver $O(N_s^{1.5}/N_p)$ are also inversely proportional to N_p , it can be concluded that the proposed parallel PWTD-accelerated TD-SIE solver is scalable.

III. NUMERICAL RESULTS

This section presents numerical examples that demonstrate the efficiency, accuracy, and applicability of the proposed parallel PWTD-accelerated TD-SIE solver. In all examples considered here, the scatterers are excited by a plane wave with electric field given by

$$\mathbf{E}^i(\mathbf{r}, t) = \hat{\mathbf{p}} G(t - \mathbf{r} \cdot \hat{\mathbf{k}}/c_0). \quad (21)$$

$G(t) = \cos(2\pi f_0(t - t_0)) \exp(-(t - t_0)^2/2\sigma^2)$ is a modulated and quasi-bandlimited Gaussian pulse, f_0 is the modulation frequency, $t_0 = 6\sigma$ is the delay, $\sigma = 3/(2\pi f_{bw})$ is a

measure of pulse duration, and $\hat{\mathbf{p}}$ and $\hat{\mathbf{k}}$ denote the polarization and propagation direction of the plane wave. The parameter f_{bw} represents the “half bandwidth” of the pulse. The minimum and maximum frequencies are $f_{min} = f_0 - f_{bw}$ and $f_{max} = f_0 + f_{bw}$; energy outside this frequency band is only 0.0022% of the pulse’s total energy. A parallel generalized minimal residual (GMRES) algorithm and a diagonal preconditioner are used to iteratively solve (7) at each time step. The GMRES iteration is terminated when the condition

$$\|\bar{\mathbf{V}}_j - \bar{\mathbf{Z}}_0 \bar{\mathbf{I}}_j^{(n)}\| < \varepsilon \|\bar{\mathbf{V}}_j\| \quad (22)$$

is satisfied. Here, $\bar{\mathbf{I}}_j^{(n)}$ represents the vector of current coefficients in the n^{th} iteration, $\bar{\mathbf{V}}_j = \bar{\mathbf{F}}_j - \sum_{i=1}^{j-1} \bar{\mathbf{Z}}_i \bar{\mathbf{I}}_{j-i}$ is total RHS at time step j , and ε is the desired residual error. At a given time step, the GMRES solver’s initial guess is constructed by extrapolating current coefficients from those obtained in previous time steps. All frequency-domain (i.e., time harmonic) data presented in this section are obtained by dividing the Fourier transform of the time-domain waveforms (whose samples are recorded during MOT) by that of $G(t)$.

All simulations were executed on a cluster of Quad-Core 850 MHz PowerPC CPUs with 4 GB/CPU memory, which is located at the King Abdullah University of Science and Technology (KAUST) Supercomputing Laboratory. The proposed scheme solved either the TD-CFIE ($\beta = 1$) or TD-EFIE leveraging a hybrid MPI and OpenMP parallelization strategy: one MPI process was launched per CPU and OpenMP utilized four cores on each CPU to parallelize the computationally most intensive loops associated with translation, near-field calculation, and construction/projection of outgoing/incoming rays at level $v = 1$.

A. Parallelization Performance and Solution for Canonical Problems

1) *Plate:* First, parallelization performance of the PWTD-accelerated TD-EFIE solver is demonstrated by analyzing scattering from a $120 \text{ m} \times 120 \text{ m}$ PEC plate. The plate is centered at the origin and positioned parallel to the xy plane. It is illuminated by $\mathbf{E}^i(\mathbf{r}, t)$ in (21) with $f_0 = 150 \text{ MHz}$, $f_{bw} = 50 \text{ MHz}$, $\hat{\mathbf{p}} = \hat{\mathbf{z}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{y}}$. The current induced on the plate is discretized using $N_s = 731,247$ spatial basis functions and fourth-order temporal basis function. The simulation is executed for $N_t = 500$ time steps with $\Delta t = 250 \text{ ps}$. A nine-level PWTD tree is constructed upon setting the side length of boxes at the finest level to 0.64λ and $\gamma = 3$. The number of boxes at each level is $N_g^v = \{65\,536, 16\,384, 4096, 1024, 256, 64, 16, 4, 1\}$.

The parallel efficiencies are computed using $\kappa = N_{ref} T_{N_{ref}} / N_p T_{N_p}$, where $T_{N_{ref}}$ and T_{N_p} are the measured execution times (including computation and communication times) on N_{ref} and N_p processors, respectively. Note that N_{ref} is chosen as the minimum number of processors (usually greater than one) required to execute the simulation in a reasonable time. This is a valid measure of the *true* parallel efficiency as the efficiency only starts to (noticeably) degrade when N_p becomes very large. κ is computed for different PWTD

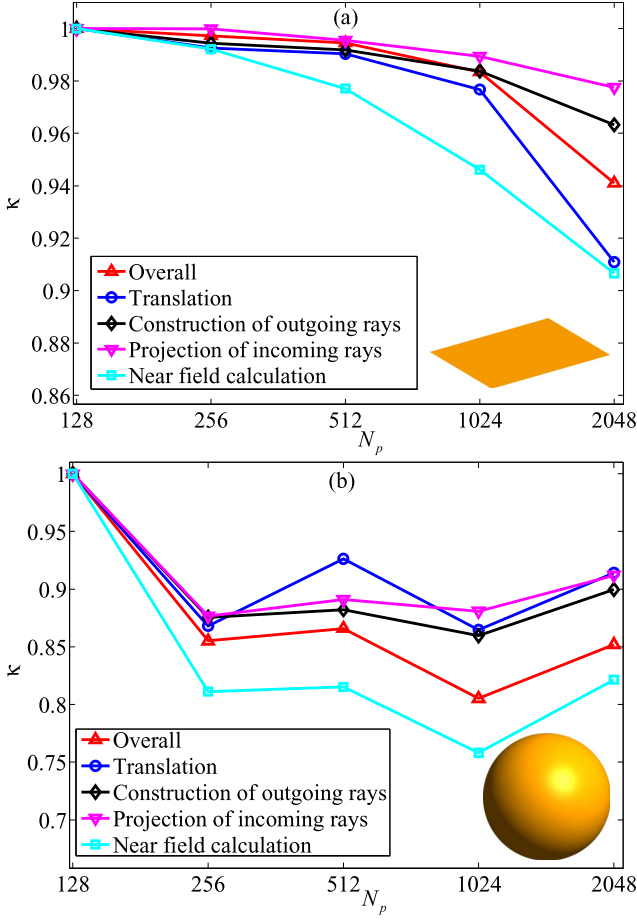


Fig. 4. Parallel efficiencies (κ) of PWT stages for the canonical problem involving (a) square PEC plate and (b) PEC sphere while N_p is changed from 128 to 2048 ($N_{ref} = 128$).

stages for $N_p = \{128, 256, 512, 1024, 2048\}$ and $N_{ref} = 128$ [Fig. 4(a)]. The base level is set to $v_b = \{4, 4, 3, 3, 2\}$, respectively. As discussed in Section II-C2, three different cases exist when constructing/projecting outgoing/incoming rays. For case 1, parallel efficiency is 100% since there is no communication between processors. As expected, the same efficiency cannot be achieved for cases 2 and 3. However, still, over 96% overall parallel efficiency is achieved for construction/projection of outgoing/incoming rays even when $N_p = 2048$. Moreover, efficiencies of 90% and 91% are achieved for the near-field calculation and the translation stages. Note that lower efficiency is observed for the near-field calculation stage compared to the other stages. This is due to nonideal load balancing when each processor is in charge of different numbers of finest level boxes, especially if the subtree has many levels. Nevertheless, the overall solver achieves 94% parallel efficiency (i.e., 15-fold speedup) when $N_p = 2048$. It is worth pointing out that this high efficiency may demonstrate the scalability in (20) as it is difficult to separately measure the computation/communication time due to extensive usage of nonblocked and asynchronous communication techniques. Fig. 5(a) plots the execution time on each processor when $N_p = 2048$; excellent computation load balance is observed. Moreover, the maximum memory usages among all processors when $N_p = 128$ and $N_p = 2048$

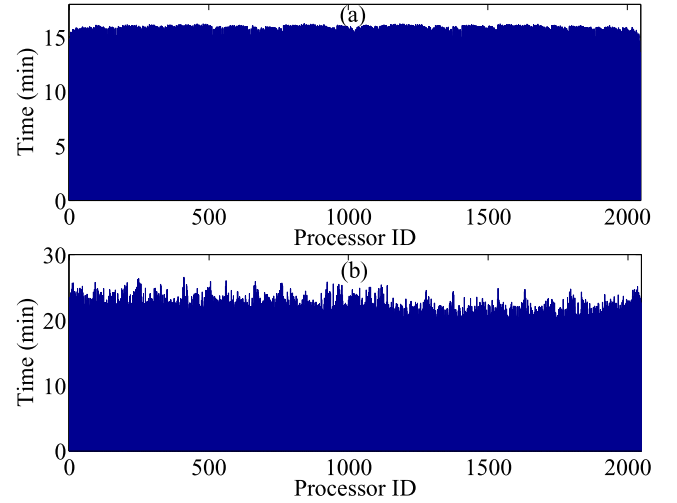


Fig. 5. Execution time on each processor for the canonical problem involving (a) square PEC plate and (b) PEC sphere when $N_p = 2048$.

TABLE I
MAXIMUM MEMORY COSTS AMONG ALL PROCESSORS FOR THE
CANONICAL PROBLEM INVOLVING A SQUARE PEC PLATE AND A PEC
SPHERE WHEN $N_p = 128$ AND 2048

	PWTD (plate)	Near-field (plate)	PWTD (sphere)	Near-field (sphere)
$N_p = 128$	1.96 GB	671.7 MB	388 MB	1.45 GB
$N_p = 2048$	135 MB	47.6 MB	26.3 MB	109.5 MB
Ratio	14.8	14.1	14.7	13.5

are listed in Table I; memory reduction ratios of 14.8 and 14.1 are achieved (compared to the ideal reduction ratio of 16) for the storage of the PWTD ray data and the near-field data, respectively. Note that lower memory reduction ratio is observed for the near-field data compared to the PWTD ray data, due to similar reasons causing the lower parallel efficiency for the near-field calculation.

The PWTD-accelerated TD-EFIE solver is then applied to the analysis of scattering from the PEC plate illuminated by $\mathbf{E}^i(\mathbf{r}, t)$ in (21) with $f_0 = 300$ MHz, $f_{bw} = 100$ MHz, $\hat{\mathbf{p}} = -\hat{\mathbf{y}}$, and $\hat{\mathbf{k}} = -\sin(5^\circ)\hat{\mathbf{x}} - \cos(5^\circ)\hat{\mathbf{z}}$. The current induced on the plate is discretized using $N_s = 2,920,476$ spatial basis functions and fourth-order temporal basis function. The simulation is executed for $N_t = 1000$ time steps with $\Delta t = 125$ ps. A ten-level PWTD tree is constructed upon setting the side length of boxes at the finest level to 0.64λ and $\gamma = 3.5$. Table II presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires 3.5 TB of memory and 7 h of CPU time when $N_p = 2048$. The solution at each time step is obtained in just one GMRES iteration, which yields $\varepsilon = 10^{-12}$.

The broadband RCS of the plate along the $+z$ direction ($\theta = 0$) obtained using the proposed solver is compared the approximate analytical solution (which only accounts for physical optics) in Fig. 6; good agreement is observed.

2) *Sphere*: Next, parallelization performance of the PWTD-accelerated TD-CFIE solver is demonstrated via the analysis of scattering from a PEC sphere of radius 1 m

TABLE II
TECHNICAL DATA FOR THE SETUPS AND SOLUTIONS OF SCATTERING PROBLEMS INVOLVING CANONICAL EXAMPLES

	Plate	Sphere
Maximum dimension	120 m (160 λ)	2 m (70 λ)
Frequency band (f_{min}, f_{max})	(200, 400) MHz	(5.12, 10.24) GHz
Number of unknowns N_s	2 920 476	9 433 437
Time step size Δt	125 ps	5 ps
Number of time steps N_t	1 000	2 300
Number of processors N_p	2 048	8 192
Memory for $\bar{\mathbf{Z}}_t$ /PWTD	0.46/3.02 TB	3.28/3.78 TB
Setup time	1.63 h	3 h
RHS time (near field)	628 s	1.38 h
RHS time (PWTD)	4 h	11.2 h
LHS time (GMRES)	108 s	3.8 h
Number of GMRES iterations	1	1

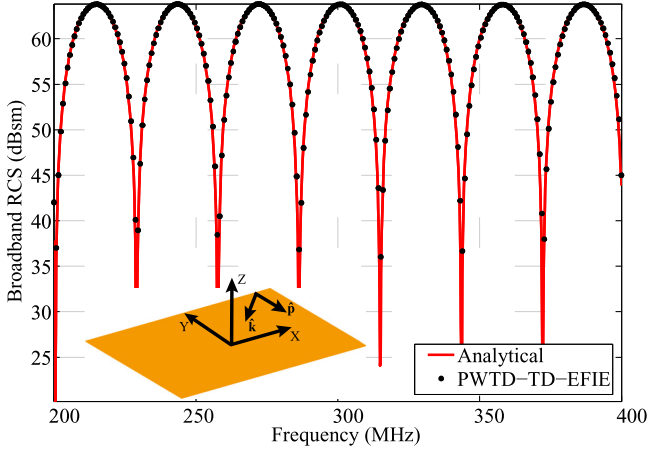


Fig. 6. broadband RCS of the PEC plate along $+z$ direction computed by the PWTD-accelerated TD-EFIE solver and analytical formula.

centered at the origin. It is illuminated by $\mathbf{E}^i(\mathbf{r}, t)$ in (21) with $f_0 = 2.3\text{GHz}$, $f_{bw} = 0.77\text{GHz}$, $\hat{\mathbf{p}} = \hat{\mathbf{x}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{z}}$. The current induced on the sphere is discretized using $N_s = 992,766$ spatial basis functions and fourth-order temporal basis function. The simulation is executed for $N_t = 900$ time steps with $\Delta t = 16.7\text{ps}$. A seven-level PWTD tree is constructed upon setting the side length of boxes at the finest level to 0.479λ and $\gamma = 3.5$. The number of boxes at each level is $N_g^v = \{18481, 4739, 1160, 272, 56, 8, 1\}$.

The parallel efficiency κ is computed for different PWTD stages for $N_p = \{128, 256, 512, 1024, 2048\}$ and $N_{ref} = 128$ [Fig. 4(b)]. The base level is set to $v_b = \{3, 3, 2, 2, 1\}$, respectively. Fig. 4(b) shows that there is a higher efficiency degradation in comparison to efficiency curves presented in Fig. 4(a). This is due to load imbalance, which is observed because the number of boxes at each level is not an integer power of two. For the same reason, the efficiencies decrease in a “staircase” fashion and exhibit noticeable local increases whenever v_b is incremented [Fig. 4(b)]. That said, the overall parallel efficiency is still rather high around 85% (corresponding to 13.6-fold speedup) even when $N_p = 2048$. Fig. 5(b) plots the execution time on each processor when $N_p = 2048$. For similar

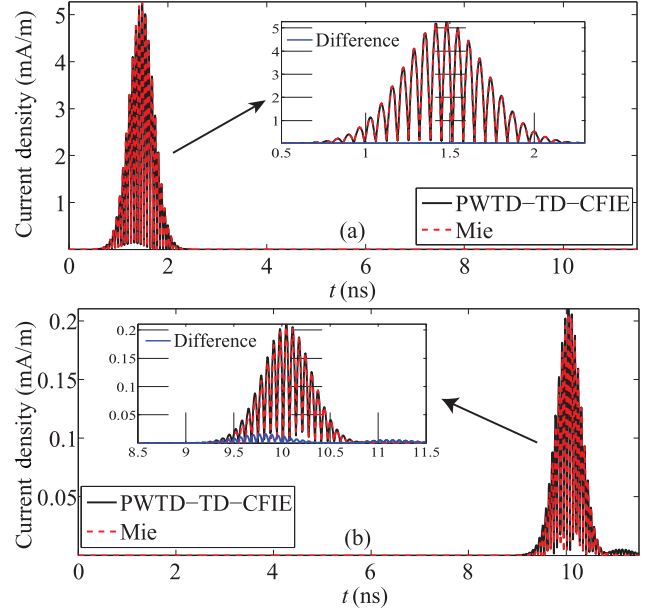


Fig. 7. Magnitudes of the current density induced at (a) ($r = 1, \theta = 180^\circ, \phi = 0$) and (b) ($r = 1, \theta = 0, \phi = 0$) on the sphere computed using the proposed solver and Mie series. Note that the maximum current magnitude at (b) is 25 times smaller than that at (a).

TABLE III
THE TECHNICAL DATA FOR THE SETUPS AND SOLUTIONS OF SCATTERING PROBLEMS INVOLVING REAL-LIFE TARGETS

	Helicopter	Airbus A-320
Maximum dimension	22 m (117.3 λ)	37.5 m (33 λ)
Frequency band (f_{min}, f_{max})	(0.8, 1.6) GHz	(133.5, 266.5) MHz
Number of unknowns N_s	2 436 813	1 020 069
Time step size Δt	31.25 ps	187 ps
Number of time steps N_t	3 000	1 600
Number of processors N_p	8 192	2 048
Memory for $\bar{\mathbf{Z}}_t$ /PWTD	907/820 GB	1152/70 GB
Setup time	2.8 h	5.6 h
RHS time (near field)	42 min	3 h
RHS time (PWTD)	14 h	2.7 h
LHS time (GMRES)	21 min	800 s
Number of GMRES iterations	1–3	2–3
RCS calculation time	2.53 h	55 min

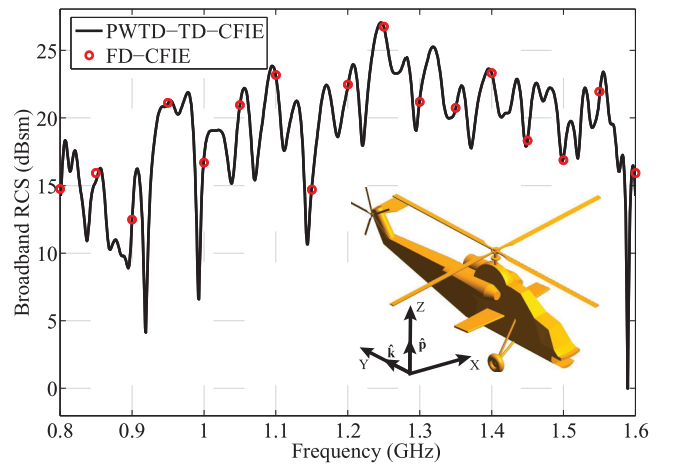


Fig. 8. Broadband RCS of the Rooivalk helicopter along the $+z$ direction computed by PWTD-accelerated TD-CFIE solver and FD-CFIE solver.

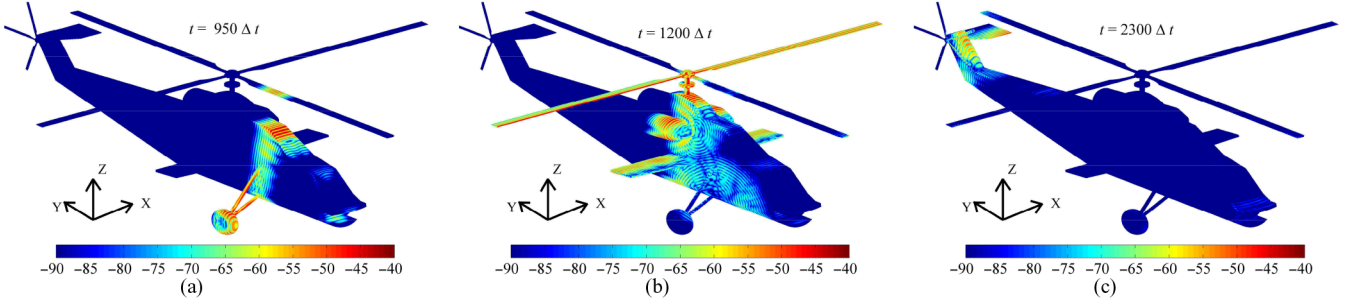


Fig. 9. Snapshots of the current density (in dB) induced on the helicopter obtained by the PWTD-accelerated TD-CFIE solver at (a) $t = 950 \Delta t$, (b) $t = 1200 \Delta t$, and (c) $t = 2300 \Delta t$.

reasons as explained above, nonideal computation load balance is observed in Fig. 5(b) compared with that in Fig. 5(a). Moreover, the maximum memory usages among all processors when $N_p = 128$ and $N_p = 2048$ are listed in Table I; memory reduction ratios of 14.7 and 13.5 are achieved (compared to the ideal reduction ratio of 16) for the storage of the PWTD ray data and the near-field data, respectively.

The PWTD-accelerated TD-CFIE solver is then applied to the analysis of scattering from the PEC sphere illuminated by $\mathbf{E}^i(\mathbf{r}, t)$ in (21) with $f_0 = 7.68$ GHz, $f_{bw} = 2.56$ GHz, $\hat{\mathbf{p}} = \hat{\mathbf{x}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{z}}$. The current induced on the sphere is discretized using $N_s = 9,433,437$ spatial basis functions and fourth-order temporal basis function. The simulation is executed for $N_t = 2,300$ time steps with $\Delta t = 5$ ps. A nine-level PWTD tree is constructed upon setting the side length of boxes at the finest level to 0.404λ and $\gamma = 4.5$. Table II presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires 7 TB of memory and 20 h of CPU time when $N_p = 8192$. The solution at each time step is obtained in just one GMRES iteration, which yields $\varepsilon = 10^{-12}$.

Current densities induced at $(\theta = 180^\circ, \phi = 0)$ and $(\theta = 0, \phi = 0)$ obtained using the proposed solver are compared with the exact Mie series solution in Fig. 7.

B. Solution for Real-World Problems

1) *Rooivalk Helicopter*: The PWTD-accelerated TD-CFIE solver is applied to the analysis of scattering from a Rooivalk helicopter model, which fits in a fictitious box of dimensions $7.4 \text{ m} \times 22 \text{ m} \times 7.1 \text{ m}$. The helicopter is illuminated by $\mathbf{E}^i(\mathbf{r}, t)$ in (21) with $f_0 = 1.2$ GHz, $f_{bw} = 0.4$ GHz, $\hat{\mathbf{p}} = \hat{\mathbf{z}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{y}}$. The current induced on the helicopter is discretized using $N_s = 2,436,813$ spatial basis functions and fourth-order temporal basis function. The simulation is executed for $N_t = 3000$ time steps with $\Delta t = 31.25$ ps. A ten-level PWTD tree is constructed upon setting the side length of boxes at the finest level to 0.48λ and $\gamma = 4$. Table III presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires around 2 TB of memory and 21 h of CPU time when $N_p = 8192$. The solution at each time step is obtained in maximum three GMRES iterations, which yields $\varepsilon = 5 \times 10^{-14}$.

The broadband RCS of the helicopter along the $+z$ direction ($\theta = 0$) is computed using the proposed solver and a

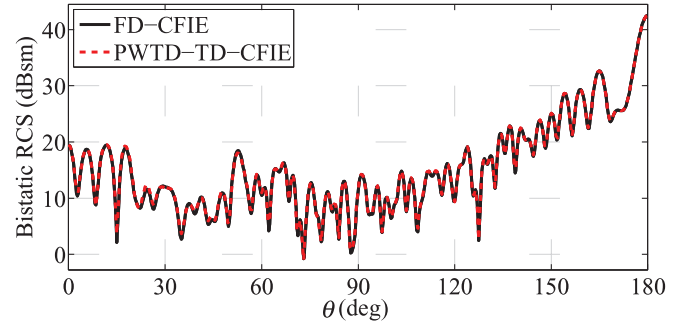


Fig. 10. Bistatic RCS of the Airbus-A320 airplane at 250 MHz computed at $\phi = 0^\circ$ and $\theta = [0, 180]^\circ$ via PWTD-accelerated TD-CFIE solver and FD-CFIE solver.

frequency-domain combined field integral equation (FD-CFIE) solver (Fig. 8). Results are in good agreement. Moreover, snapshots of the current induced on the helicopter at times $950 \Delta t$, $1200 \Delta t$, and $2300 \Delta t$ are shown in Fig. 9. The amplitudes of currents on edges and regions where multiple reflections exist are larger than those elsewhere.

2) *Airbus-A320*: Finally, the PWTD-accelerated PWTD solver is applied to the analysis of transient scattering from an Airbus-A320 model, which fits in a hypothetical box with dimensions $34.2 \text{ m} \times 11.7 \text{ m} \times 37.5 \text{ m}$. The airplane is illuminated by $\mathbf{E}^i(\mathbf{r}, t)$ in (21) with $f_0 = 200$ MHz, $f_{bw} = 67.5$ MHz, $\hat{\mathbf{p}} = \hat{\mathbf{y}}$, and $\hat{\mathbf{k}} = -\hat{\mathbf{z}}$. The current induced on the airplane is discretized using $N_s = 1,020,069$ spatial basis functions and fourth-order temporal basis function. The simulation is executed for $N_t = 1600$ time steps with $\Delta t = 187$ ps. An eight-level PWTD tree is constructed upon setting the side length of boxes at the finest level to 0.38λ and $\gamma = 4.5$. Table III presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires about 1.2 TB of memory and 12.6 h of CPU time when $N_p = 2048$. The solution at each time step is obtained in maximum three GMRES iterations, which yields $\varepsilon = 5 \times 10^{-15}$.

The bistatic RCS of the airplane is computed at 250 MHz and compared to those obtained using a FD-CFIE solver (Fig. 10); results are in good agreement. In addition, snapshots of the current induced on the airplane at times $640 \Delta t$, $800 \Delta t$, and $1140 \Delta t$ are shown in Fig. 11. The maximum current densities are induced on edges of antennas placed on top of the airplane and the engine intake.

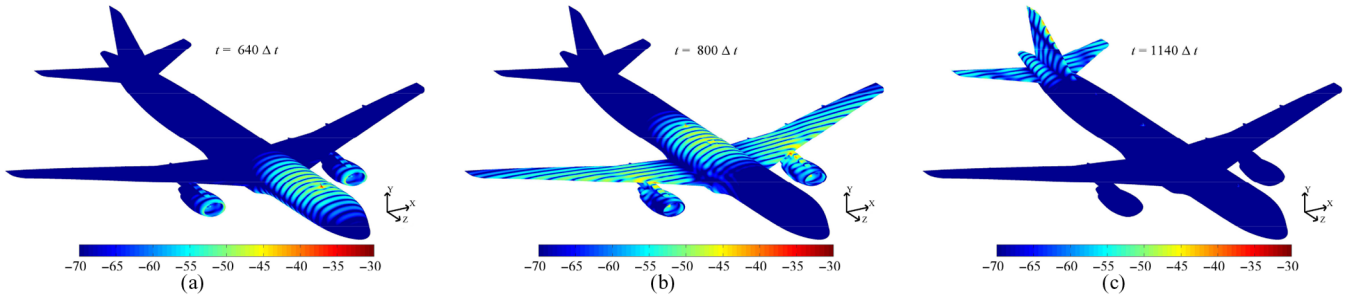


Fig. 11. Snapshots of the current density (in dB) induced on the Airbus-A320 obtained by the PWTD-accelerated TD-CFIE solver at (a) $t = 640 \Delta t$, (b) $t = 800 \Delta t$, and (c) $t = 1140 \Delta t$.

IV. CONCLUSION

A scheme for efficiently parallelizing PWTD-accelerated TD-SIE solvers is presented. Its efficiency is achieved by hierarchically partitioning the computation and memory loads along the spatial, angular, and temporal dimensions. The resulting partitioned memory, CPU, and communication costs scale as $O(N_s^{1.5}/N_p)$, $O(N_t N_s \log^2 N_s/N_p)$, and $O(N_t N_s \log N_s/N_p)$, respectively. Furthermore, to enable scalable communications among processors, the parallelization scheme employs a mix of nonblocked MPI and queue-based, memory-efficient, and asynchronous communication techniques. Indeed, numerical results validate the scalability and parallel efficiency of the proposed scheme up to 2048 processors. The resulting PWTD-accelerated TD-SIE solver is successfully applied to analysis of transient scattering from objects measuring well over 100 wavelengths in size and discretized using 10 million spatial unknowns.

ACKNOWLEDGMENT

The authors would like to thank the King Abdullah University of Science and Technology (KAUST) Supercomputing Laboratory for providing the required computational resources.

REFERENCES

- [1] W. C. Chew, E. Michielssen, J. Song, and J. Jin, *Fast and Efficient Algorithms in Computational Electromagnetics*. Norwood, MA, USA: Artech House, 2001.
- [2] A. A. Ergin, B. Shanker, and E. Michielssen, "The plane-wave time-domain algorithm for the fast analysis of transient wave phenomena," *IEEE Antennas Propag. Mag.*, vol. 41, no. 4, pp. 39–52, Sep. 1999.
- [3] B. Shanker, A. A. Ergin, and E. Michielssen, "Plane-wave-time-domain-enhanced marching-on-in-time scheme for analyzing scattering from homogeneous dielectric structures," *J. Opt. Soc. Amer. A*, vol. 19, pp. 716–726, 2002.
- [4] B. Shanker, A. A. Ergin, M. Lu, and E. Michielssen, "Fast analysis of transient electromagnetic scattering phenomena using the multilevel plane wave time domain algorithm," *IEEE Trans. Antennas Propag.*, vol. 51, no. 3, pp. 628–641, Mar. 2003.
- [5] J. Song, C.-C. Lu, and W. C. Chew, "Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1488–1493, Oct. 1997.
- [6] K. Aygun, M. Lu, N. Liu, A. Yilmaz, and E. Michielssen, "A parallel PWTD accelerated time marching scheme for analysis of EMC/EMI problems," in *Proc. IEEE Int. Symp. Electromagn. Compat.*, 2003, pp. 863–866.
- [7] N. Liu, M. Lu, B. Shanker, and E. Michielssen, "The parallel plane wave time domain algorithm-accelerated marching on in time solvers for large-scale electromagnetic scattering problems," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2004, pp. 4212–4215.
- [8] S. Velampambal, W. C. Chew, and J. Song, "10 million unknowns: Is it that big?," *IEEE Trans. Antennas Propag.*, vol. 45, no. 2, pp. 43–58, Apr. 2003.
- [9] S. Velampambal and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2719–2727, Aug. 2005.
- [10] J. Fostier and F. Olyslager, "Provably scalable parallel multilevel fast multipole algorithm," *Electron. Lett.*, vol. 44, pp. 1111–1113, 2008.
- [11] O. Ergul and L. Gurel, "A hierarchical partitioning strategy for an efficient parallelization of the multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.*, vol. 57, no. 6, pp. 1740–1750, Jun. 2009.
- [12] J. Fostier and F. Olyslager, "An open-source implementation for full-wave 2D scattering by million-wavelength-size objects," *IEEE Antennas Propag. Mag.*, vol. 52, no. 5, pp. 23–34, Oct. 2010.
- [13] O. Ergul and L. Gurel, "Rigorous solutions of electromagnetic problems involving hundreds of millions of unknowns," *IEEE Antennas Propag. Mag.*, vol. 53, no. 1, pp. 18–27, Feb. 2011.
- [14] V. Melapudi, B. Shanker, S. Seal, and S. Aluru, "A scalable parallel wide-band MLFMA for efficient electromagnetic simulations on large scale clusters," *IEEE Trans. Antennas Propag.*, vol. 59, no. 7, pp. 2565–2577, Jul. 2011.
- [15] B. Shanker, A. A. Ergin, K. Aygun, and E. Michielssen, "Analysis of transient electromagnetic scattering from closed surfaces using a combined field integral equation," *IEEE Trans. Antennas Propag.*, vol. 48, no. 7, pp. 1064–1074, Jul. 2000.
- [16] Y. Shi, H. Bagci, and M. Lu, "On the internal resonant modes in marching-on-in-time solution of the time domain electric field integral equation," *IEEE Trans. Antennas Propag.*, vol. 61, no. 8, pp. 4389–4392, Aug. 2013.
- [17] G. Manara, A. Monorchio, and R. Reggiannini, "A space-time discretization criterion for a stable time-marching solution of the electric field integral equation," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 527–532, Mar. 1997.
- [18] S. M. Rao, D. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. 30, no. 3, pp. 409–418, May 1982.
- [19] J. Knab, "Interpolation of band-limited functions using the approximate prolate series," *IEEE Trans. Inf. Theory*, vol. 25, no. 6, pp. 717–720, Nov. 1979.
- [20] R. Jakob-Chien and B. K. Alpert, "A fast spherical filter with uniform resolution," *J. Comput. Phys.*, vol. 136, pp. 580–584, 1997.
- [21] J. Fostier and F. Olyslager, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2346–2355, Aug. 2008.



Yang Liu (S'11–M'15) received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, the M.S. degrees in electrical engineering and mathematics, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2010, 2013, 2014, and 2015, respectively.

Since August 2010, he has been a Graduate Research Assistant with the Radiation Laboratory, University of Michigan. Since June 2015, he has been working as a Research Fellow at the Radiation Laboratory, University of Michigan. His research interests include computational electromagnetics, with focus on fast time-domain and frequency-domain integral equation methods.

Dr. Liu authored the second place paper of the student paper competition of the 28th International Review of Progress in Applied Computational Electromagnetics, 2012. He also co-authored the first place paper of the student paper competition of the 12th International Workshop on Finite Elements for Microwave Engineering, 2014.



Abdulkadir C. Yücel received the B.S. degree (*summa cum laude*) in electronics engineering from Gebze Institute of Technology, Kocaeli, Turkey, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2005, 2008, and 2013, respectively.

From September 2005 to August 2006, he worked as a Research and Teaching Assistant with the Electronics Engineering Department, Gebze Institute of Technology, Kocaeli, Turkey. From August 2006 to May 2013, he was a Graduate Student Research

Assistant with the Radiation Laboratory, University of Michigan. Since May 2013, he has been working as a Research Fellow with the Radiation Laboratory, University of Michigan. His research interests include various aspects of computational electromagnetics with emphasis on uncertainty quantification for electromagnetic analysis on complex platforms, electromagnetic compatibility and interference analysis, nature-based design of electromagnetic devices, and integral-equation-based frequency- and time-domain solvers and their accelerators.

Dr. Yücel was the recipient of the Fulbright Fellowship in 2006 and Electrical Engineering and Computer Science Departmental Fellowship of University of Michigan in 2007.



Hakan Bağcı (S'98–M'07–SM'14) received the B.S. degree in electrical and electronics engineering from the Bilkent University, Ankara, Turkey, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 2001, 2003, and 2007, respectively.

From June 1999 to July 2001, he worked as an Undergraduate Researcher with the Computational Electromagnetics Group, Bilkent University, Ankara, Turkey. From August 2001 to December 2006, he

was a Research Assistant with the Center for Computational Electromagnetics and Electromagnetics Laboratory, UIUC. From January 2007 to August 2009, he worked as a Research Fellow with the Radiation Laboratory, University of Michigan, Ann Arbor, MI, USA. In August 2009, he joined the Division of Physical Sciences and Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, as an Assistant Professor of Electrical Engineering. His research interests include various aspects of computational electromagnetics with emphasis on time-domain integral equations and their fast marching-on-in-time-based solutions, well-conditioned integral-equation formulations, and development of fast hybrid methods for analyzing statistical EMC/EMI phenomena on complex and fully loaded platforms.

Dr. Bağcı was the recipient of the 2008 International Union of Radio Scientists (URSI) Young Scientist Award and the 2004–2005 Interdisciplinary Graduate Fellowship from the Computational Science and Engineering Department, UIUC. His paper titled *Fast and Rigorous Analysis of EMC/EMI Phenomena on Electrically Large and Complex Structures Loaded With Coaxial Cables* was one of the three finalists (with honorable mention) for the 2008 Richard B. Schulz Best Transactions Paper Award given by the IEEE Electromagnetic Compatibility Society. He has authored and coauthored three finalist papers and another paper, which is awarded honorable mention, in the Student Paper Competitions at the 2005, 2008, and 2010 IEEE Antennas and Propagation Society International Symposiums.



Eric Michielssen (M'95–SM'99–F'02) received the M.S. degree (*summa cum laude*) in electrical engineering from the Katholieke Universiteit Leuven (KUL), Leuven, Belgium, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 1987 and 1992, respectively.

He joined as the Faculty of the UIUC Department of Electrical and Computer Engineering in 1993, reaching the rank of Full Professor in 2002. In 2005, he joined the University of Michigan (UM), Ann Arbor, MI, USA, as a Professor of Electrical Engineering and Computer Science. Since 2009, he has been directing the University of Michigan Computational Science Certificate Program. He has authored or coauthored more than 160 journal papers and book chapters and more than 300 papers in conference proceedings. His research interests include all aspects of theoretical and applied computational electromagnetics, the development of fast frequency and time-domain integral-equation-based techniques for analyzing electromagnetic phenomena, and the development of robust optimizers for the synthesis of electromagnetic/optical devices.

Dr. Michielssen served as the Technical Chairman of the 1997 Applied Computational Electromagnetics Society (ACES) Symposium (Review of Progress in Applied Computational Electromagnetics, March 1997, Monterey, CA, USA), and served on the ACES Board of Directors (1998–2001 and 2002–2003) and as ACES Vice-President (1998–2001). From 1997 to 1999, he was an Associate Editor for *Radio Science*, and from 1998 to 2008, he served as an Associate Editor for the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION. He is a member of URSI Commission B. He was the recipient of the Belgian American Educational Foundation Fellowship in 1988, a Schlumberger Fellowship in 1990, the 1994 International Union of Radio Scientists (URSI) Young Scientist Fellowship, a 1995 National Science Foundation CAREER Award, and the 1998 Applied Computational Electromagnetics Society (ACES) Valued Service Award. In addition, he was named 1999 URSI United States National Committee Henry G. Booker Fellow and selected as the recipient of the 1999 URSI Koga Gold Medal. He was also the recipient of the UIUC's 2001 Xerox Award for Faculty Research, appointed 2002 Beckman Fellow in the UIUC Center for Advanced Studies, named 2003 Scholar in the Tel Aviv University Sackler Center for Advanced Studies, selected as UIUC 2003 University and Sony Scholar; in 2011, the UM College of Engineering David E. Liddle Research Excellence Award.