SOLVING ELECTRICALLY VERY LARGE TRANSIENT ELECTROMAGNETIC PROBLEMS USING PLANE-WAVE TIME-DOMAIN ALGORITHMS

by
Yang Liu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in The University of Michigan
2015

Doctoral Committee:

Professor Eric Michielssen, Chair
Associate Professor Anthony Grbic
Professor Robert Krasny
Professor Kenneth Powell
Professor Kamal Sarabandi

To My Family,

ACKNOWLEDGMENTS

I would like to express my greatest gratitude to Prof. Eric Michielssen for his constant intellectual, financial and spiritual support of the development of this work. Most importantly, his scientific enthusiasm and sincerity will keep inspiring me to do high-quality work in the future career. I would like to thank Professors Anthony Grbic, Kamal Sarabandi, Robert Krasny, and Kenneth Powell for serving in my thesis committee and providing valuable comments on this work. I want to thank Prof. Hakan Bağcı, Vitaliy Lomakin and Anna C. Gilbert for their contributions to the publications related to this work.

Sincere thanks go to Dr. Abdulkadir C. Yücel, Felipe Valdés, and Ahmed Al-Jarro for working closely with me for preparation of this dissertation. Full access to their research work has greatly accelerated the process of my PhD study. For similar reasons I want to thank my past and current colleagues in Prof. Eric Michielssen's research group for indirectly contributing to this work: Han Guo, Mohsen Ghaffari-Miab, Onur Bakir, Xi Lin, Luis Gomez, Zhongming Qu and Weitian Sheng. I would also like to thank my friends and fellow students from the Radiation Lab: Jiangfeng Wu, Amr Ibrahim, Morteza Sheikhsofla, Behzad Yektakhah, Armin Jam, Sangjo Choi, Hamid Nejati, Meysam Moallem, Wei Tian, Yiting Zhang, Peng Tian, Jun-Chieh Wang, Brian Tierney, Gurkan Gok, Xiaoyu Wang, Seungku Lee, Milad Zolfagharloo, Seyit Sis, Fatemeh Akbar, Guanbo Chen, Ruzbeh Akbar, Xueyang Duan, Ning Wang, Shanghua Yang, etc.

I owe special thanks to my undergraduate research advisors in Shanghai Jiao Tong University: Prof. Zhigang Zhang (embedded system), Xinbing Wang (wireless communication), Junping Geng (electromagnetics), Ronghong Jin (electromagnetics), Xinen Zhu (electromagnetics), Jianping Chen (optical communication), and Dr. Guangyue Sun (embedded system). They were always more than willing to give pertinent suggestions and guide me through any difficulties I encountered. These research experiences finally inspired me to start the journey of pursuing a PhD in computational electromagnetics.

All this work would not be possible without the endless support from my family, including but not limited to, my parents Yaozhong Liu, Junjing Cheng, my brother Guang Liu, and my wife's parents Hang Lin, Qiongyu Xia, etc. I especially want to thank my grandfather, Xinguo Liu, who always taught me to be a good man with the right attitude. Last but not least, I would sincerely thank my lovely wife, Lin Lin. Her graceful patience and unwavering love have ensured the successful completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Background

The marching-on-in-time (MOT)-based time domain (TD) integral equation (IE) methods provide an appealing avenue for solving various transient electromagnetic problems arising in scattering analysis [1, 2], electromagnetic interface/compatibility (EMI/EMC) analysis [3-5] and new antenna/metamaterial design [6, 7], etc. Compared to differential equation (DE) methods, they enjoy several advantages: first, IE methods implicitly impose the radiation boundary condition as opposed to DE methods that artificially truncate the computation domain; second, for surface scatterers, IE methods only require discretization of the scatterer surfaces as opposed to DE methods that discretize the volume enclosing the scatterers. However, the widespread use of MOT-TDIE solvers was oftentimes hindered by their numerical instability and computational inefficiency, which has led to extensive research works. In fact, the last decade has witnessed unprecedented developments in rapidly converging, accurate, stable and fast MOT-TDIE solvers capable of solving large and complex electromagnetic problems involving perfect electrically conducting (PEC) and dielectric objects, etc.

However, these newly developed MOT-TDIE solvers, despite of their efficiencies in solving broadband or transient electromagnetic problems, still lag behind the frequency domain (FD) IE solvers in their capabilities of simulating real-life electromagnetic problems that oftentimes involve millions of spatial unknowns. In fact, the MOT-TDIE solvers (with fast algorithm accelerated) typically require at least one order of magnitude more computational and memory resources than the FDIE solvers due to the presence of extra temporal dimension in the MOT-TDIE solvers. Therefore, this thesis focuses on advancing the capabilities of one specific class of fast MOT-TDIE solvers, viz., the multilevel plane-wave time-domain (PWTD) algorithm-accelerated MOT-TDIE solvers.

The multilevel PWTD algorithm constitutes the time domain counterpart of the frequency domain multilevel fast multipole algorithm (MLFMA) [8]. Compared to other fast (TD) algorithms such as the time domain adaptive integral method (TD-AIM) [9] and nonuniform-grid time-domain algorithm (NGTD) [10], PWTD enjoys favorably low computational complexity. Furthermore, PWTD and its various extensions have been applied to transient analyses that involve various types of objects, background media and frequency regimes. In the past, PWTD-accelerated MOT-TDIE solvers have been applied to a broad class of complex and large-scale transient electromagnetic problems. Indeed, transient scattering problems that involve half million spatial unknowns have been solved using PWTD-accelerated MOT-TDIE solvers. Having said that, in order to solve real-life transient magnetic problems via the PWTD-accelerated MOT-TDIE solvers, both efficient parallelization and algorithmic improvements of the PWTD algorithms are called for.

1.2 Review of Previous Work

This section provides a review of MOT-TDIE solvers. Section 1.2.1 describes the formulation and space-time discretization of TDIEs for solving transient problems involving PEC and dielectric objects. Schemes that improve the convergence, accuracy, stability and computational efficiency of the MOT-TDIE solvers are reviewed in Section 1.2.2-1.2.5, with an emphasis on the PWTD-accelerated MOT-TDIE solvers in Section 1.2.5.3.

1.2.1 MOT-TDIE Solvers for Problems Involving PEC and Dielectric Objects

1.2.1.1 PEC Scatterers in lossless medium

Transient scattering from PEC scatterers can be efficiently analyzed using TD surface integral equations (SIE). Consider a closed PEC surface S that resides in a lossless and unbounded background medium. The surface is illuminated by an incident electromagnetic field $\{\mathbf{E}^i(\mathbf{r},t),\mathbf{H}^i(\mathbf{r},t)\}$, which is assumed to be temporally bandlimited to maximum frequency f_{max} and vanishingly small for. $t<0$. The incident field induces on S a current density $\mathbf{J}(\mathbf{r},t)$ that, in turn, generates a scattered field $\{\mathbf{E}^s(\mathbf{r},t),\mathbf{H}^s(\mathbf{r},t)\}$. By enforcing the boundary condition for the total electric or magnetic field tangential to S, the problem can be formulated by time domain electric field integral equation (TD-EFIE) and time domain magnetic field integral equation (TD-MFIE) respectively as

$$\begin{aligned}-\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\mathbf{E}^i(\mathbf{r},t) &= \hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\mathbf{E}^s(\mathbf{r},t)\\ &= \hat{\mathbf{n}}\times\eta_0\mathcal{L}_e[\mathbf{J}](\mathbf{r},t) \quad \forall\mathbf{r}\in S,S^+,S^-\end{aligned} \tag{1.1}$$

$$\begin{aligned}\hat{\mathbf{n}}\times\mathbf{H}^i(\mathbf{r},t) &= -\hat{\mathbf{n}}\times\mathbf{H}^s(\mathbf{r},t)\\ &= \mathcal{L}_h[\mathbf{J}](\mathbf{r},t) \quad \forall\mathbf{r}\in S^-\end{aligned} \tag{1.2}$$

Here, $\hat{\mathbf{n}}$ is the outward unit normal to S, η_0 is the characteristic impedance of the background medium, S^- and S^+ denote the surfaces conformal to but just inside and outside S, respectively. Note: the TD-EFIE in (1.1) is also valid for open PEC surfaces. The TD-EFIE operator $\mathcal{L}_e=\mathcal{L}_{es}+\mathcal{L}_{eh}$ composes of a singular (vector potential) component $\mathcal{L}_{es}$ and a hypersingular (scalar potential) component $\mathcal{L}_{eh}$ as

$$\mathcal{L}_{es}[\mathbf{J}](\mathbf{r},t) = -\frac{1}{4\pi c_0}\hat{\mathbf{n}}\times\int_S ds'\frac{\dot{\mathbf{J}}(\mathbf{r}',\tau)}{R} \tag{1.3}$$

$$\mathcal{L}_{eh}[\mathbf{J}](\mathbf{r},t) = \frac{c_0}{4\pi}\hat{\mathbf{n}}\times\int_S ds'\nabla\frac{\int_0^\tau dt'\nabla'\cdot\mathbf{J}(\mathbf{r}',t')}{R} \tag{1.4}$$

The TD-MFIE operator is

$$\mathcal{L}_h[\mathbf{J}](\mathbf{r},t) = -\frac{1}{4\pi}\hat{\mathbf{n}}\times\int_S ds'\nabla\times\frac{\mathbf{J}(\mathbf{r}',\tau)}{R} \tag{1.5}$$

Here $R = |\mathbf{r}-\mathbf{r}'|$, c_0 is the speed of light in the background medium $\tau = t - R/c_0$, represents the delayed time and the dot on a symbol denotes time derivative. Since both $\mathcal{L}_e$ and $\mathcal{L}_h$ have a null space that permits the presence of nonphysical oscillating currents, the solutions of TD-EFIE and TD-MFIE are oftentimes corrupted by spurious resonance modes. The time domain combined field integral equation (TD-CFIE), free of these spurious resonance modes, can be constructed by linearly combining TD-EFIE (1.1) and TD-MFIE (1.2) with a combination constant β as [1]

$$\begin{aligned}&\hat{\mathbf{n}}\times\mathbf{H}^i(\mathbf{r},t) - \beta/\eta_0\,\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\mathbf{E}^i(\mathbf{r},t)\\ &= \mathcal{L}_h[\mathbf{J}](\mathbf{r},t) + \beta\hat{\mathbf{n}}\times\mathcal{L}_e[\mathbf{J}](\mathbf{r},t)\\ &= \mathcal{L}_c[\mathbf{J}](\mathbf{r},t) \qquad\qquad \forall\mathbf{r}\in S^-\end{aligned} \tag{1.6}$$

Here, $\mathcal{L}_c$ is the TD-CFIE operator. Note that the TD-CFIE reduces to the TD-EFIE and TD-MFIE when $\beta = \infty$ and $\beta = 0$, respectively.

To numerically solve (1.6), the surface S is discretized by a planar triangle mesh. Note: the minimum edge length Δs is properly chosen to resolve both the geometrical details and the wavelength corresponding to maximum frequency, $\lambda = c_0 / f_{max}$. The current density $\mathbf{J}(\mathbf{r},t)$ is expanded using N_s spatial basis functions and N_t temporal basis functions as

$$\mathbf{J}(\mathbf{r},t) = \sum_{n=1}^{N_s} f_n(t)\mathbf{S}_n(\mathbf{r}) = \sum_{j=1}^{N_t}\sum_{n=1}^{N_s} I_{j,n}T_j(t)\mathbf{S}_n(\mathbf{r}) \tag{1.7}$$

Here $f_n(t)$ is the current time signature associated with spatial basis function $\mathbf{S}_n(\mathbf{r})$ and $I_{j,n}$ is the current expansion coefficient associated with the space-time basis function $T_j(t)\mathbf{S}_n(\mathbf{r})$. $\mathbf{S}_n(\mathbf{r})$ is often chosen as the Rao-Wilton-Glisson (RWG) basis function defined on the n^{th} internal edge of the mesh [11]. $T_j(t) = T(t - j\Delta t)$ is the time shifted local Lagrange interpolant with time step size $\Delta t = 1/(2\chi_t f_{max})$; $\chi_t > 1$ is the temporal oversampling factor [12]. As an example, the local Lagrange polynomials of order

$d = 1,2,3,4$ are shown in Figure 1.1(a)-(d). Note that they are piecewise smooth for $(k-1)\Delta t \le t \le k\Delta t$, $k = 0,...,d$ and are nonzero for $-\Delta t < t < d\Delta t$.

Upon substituting (1.7) into (1.6), spatially testing (1.6) with spatial basis functions $\mathbf{S}_m(\mathbf{r})$, $m = 1,...,N_s$ and enforcing (1.6) at discrete times $i\Delta t$, $i = 1,...,N_t$, the following set of linear equations are obtained:

$$\bar{\bar{\mathbf{Z}}}_0 \bar{\mathbf{I}}_i = \bar{\mathbf{V}}_i - \sum_{k=1}^{\min\{i-1,k_{max}\}} \bar{\bar{\mathbf{Z}}}_k \bar{\mathbf{I}}_{i-k}, \tag{1.8}$$

Here, the entries of the excitation vectors $\bar{\mathbf{V}}_i$, current coefficient vectors $\bar{\mathbf{I}}_i$ and matrices $\bar{\bar{\mathbf{Z}}}_k$ are

$$\{\bar{\bar{\mathbf{Z}}}_k\}_{mn} =< \mathbf{S}_m(\mathbf{r}), \mathcal{L}_c[\mathbf{S}_n T_{-k}](\mathbf{r},t) >|_{t=0} \tag{1.9}$$

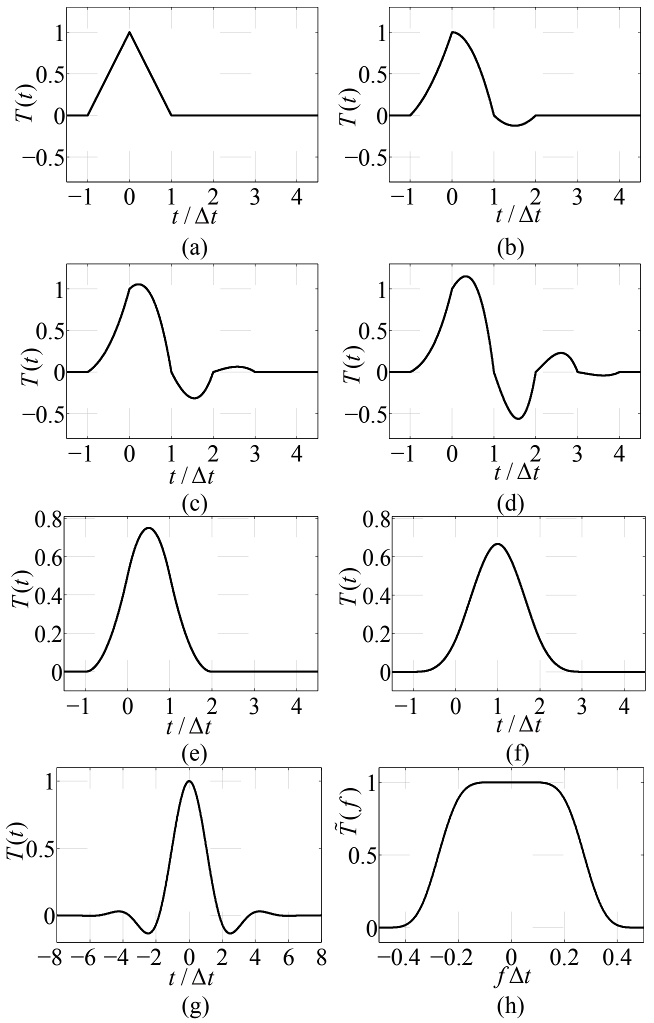
$$\{\bar{\mathbf{V}}_i\}_m =< \mathbf{S}_m(\mathbf{r}), \hat{\mathbf{n}} \times \mathbf{H}^i(\mathbf{r},t) - \frac{\beta}{\eta_0} \hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \mathbf{E}^i(\mathbf{r},t) >|_{t=i\Delta t} \tag{1.10}$$

and $\{\bar{\mathbf{I}}_i\}_n = I_{i,n}$. The number of nonzero impedance matrices $\bar{\bar{\mathbf{Z}}}_k$ is approximately $k_{max} = \lceil D_{max} / c_0 \Delta t \rceil$ where D_{max} denotes the maximum linear dimension of the scatterer. The above set of linear equations (1.8) can be solved by MOT: First, $\bar{\mathbf{I}}_1$ is computed by solving (1.8) for $i = 1$ using iterative methods. Then, for $i = 2$, the summation on the right hand side (RHS) of (1.8) is computed and the resulting system is solved for $\bar{\mathbf{I}}_2$. This process is repeated to compute $\bar{\mathbf{I}}_3$ and so on.

The computation of MOT matrices (1.9) in the above-described MOT scheme, involves evaluation of the temporal integral in the TD-EFIE operator (1.4), which can be computationally expensive and prone to error. An alternative form of the TD-EFIE that avoids numerical temporal integration is

$$-\hat{\mathbf{n}} \times \hat{\mathbf{n}} \times \dot{\mathbf{E}}^i(\mathbf{r},t) = \hat{\mathbf{n}} \times \eta_0 \dot{\mathcal{L}}_e[\mathbf{J}](\mathbf{r},t) \tag{1.11}$$

Equation (1.11) is called differentiated TD-EFIE. Similarly, differentiated TD-MFIE/CFIE can be formulated. The MOT scheme for differentiated TDIEs is obtained by replacing $\mathcal{L}_h/\mathcal{L}_e$ in (1.9) by $\dot{\mathcal{L}}_h/\dot{\mathcal{L}}_e$, and $\mathbf{E}^i/\mathbf{H}^i$ in (1.10) by $\dot{\mathbf{E}}^i/\dot{\mathbf{H}}^i$ [13].



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1.1 Temporal basis functions. (a)-(d) Lagrange interpolants of order, $d = 1, 2, 3, 4,$ respectively. (e) Quadratic B-Spline function. (f) Cubic B-Spline function. (g) APS function for $t \in [-p_f \Delta t, p_f \Delta t]$, $p_f = 8$, oversampling factor $\chi_t = 8$. (h) Fourier transform of (g).

1.2.1.2 Homogeneous dielectrics in lossless medium

Transient scattering from homogeneous (or piecewise homogeneous) dielectrics in lossless unbounded medium can be analyzed by solving a coupled pair of TD-SIEs involving both equivalent surface electric and magnetic currents radiating in the interior and exterior regions. Among many choices of how the equations are coupled, the most popular ones are the time domain Müller [14, 15] and Poggio-Miller-Chang-Harrington-Wu-Tsai (PMCHWT) formulations [16-18]. Upon discretizing the electric/magnetic currents using space-time basis functions and spatially testing the equations at discrete times, MOT systems similar to (1.8) can be obtained [15].

1.2.1.3 Inhomogeneous dielectrics in lossless medium

Transient scattering from inhomogeneous dielectrics in lossless unbounded medium can be analyzed using TD volume integral equations (VIE) [19] in terms of equivalent volume electric polarization current $\mathbf{J}^{P}(\mathbf{r},t)$

$$\mathbf{J}^{P}(\mathbf{r},t)=\dot{\mathbf{D}}(\mathbf{r},t)-\varepsilon_0\dot{\mathbf{E}}(\mathbf{r},t) \tag{1.12}$$

where $\mathbf{D}(\mathbf{r},t)$ is the electric flux density, $\mathbf{E}(\mathbf{r},t)$ is the total electric field and ε_0 is the permittivity of the background medium. The flux density and total field are related by the following formulas.

- Lossless dielectrics: $\mathbf{D}(\mathbf{r},t)=\varepsilon(\mathbf{r})\mathbf{E}(\mathbf{r},t)$. Here $\varepsilon(\mathbf{r})$ denotes the frequency independent permittivities of the scatterers.
- Lossy dielectrics: $\dot{\mathbf{D}}(\mathbf{r},t)=\sigma(\mathbf{r})\mathbf{E}(\mathbf{r},t)+\varepsilon(\mathbf{r})\dot{\mathbf{E}}(\mathbf{r},t)$. Here the flux density has a conduction current contribution [20] and $\sigma(\mathbf{r})$ denotes frequency independent conductivities of the scatterers.
- Dispersive dielectrics: $\mathbf{D}(\mathbf{r},t)=\varepsilon(\mathbf{r},t)*\mathbf{E}(\mathbf{r},t)$, where $\varepsilon(\mathbf{r},t)$ denotes the permittivities of the dispersive scatterers, and $*$ denotes temporal convolution.

The unknown electric flux density can be solved using MOT schemes similar to (1.8). Note that for lossy and dispersive dielectrics, the total field can be updated from the electric flux density via recursive computation [20, 21].

In addition to the abovementioned MOT-TDIE solvers, those applicable to problems involving surface scatterers embedded in lossy/structured medium [22, 23], 2D objects [24] and wire structures [25], etc., have been developed.

1.2.2 Well-conditioned MOT-TDIE Solvers

When solving the matrix equation (8) using iterative methods, the number of iterations for the solution to converge is typically proportional to the condition number (the ratio between the largest and smallest singular value) of $\bar{\bar{\mathbf{Z}}}_0$. Unfortunately, the standard MOT-TD-EFIE/CFIE solvers suffer from two types of breakdowns, viz., the condition number of $\bar{\bar{\mathbf{Z}}}_0$ grows without bound when time step size $\Delta t \to 0$ (i.e., when the excitation is a low frequency pulse) or mesh size $\Delta s \to 0$ (i.e., when the intricate geometry feature results in a dense/mixed-scaled mesh).

1.2.2.1 Low frequency breakdown

The low frequency breakdown occurs in MOT-TD-EFIE solvers for reasons relating to the inconsistent asymptotic behaviors of the (RWG basis) discretized vector potential component $\mathcal{L}_{es}$ and scalar potential component $\mathcal{L}_{eh}$ of the TD-EFIE operator in the low frequency regime. As a result, the vector potential component will lose its contribution as its magnitude becomes too small compared to that of the scalar potential component; on the other hand, the dominating scalar potential component $\mathcal{L}_{eh}$ has a null space for solenoidal (divergence free) current that results in an ill-conditioned MOT system. This low frequency breakdown can be avoided by discretizing the current using (properly weighted) solenoidal/nonsolenoidal sub-domain spatial basis functions. Specifically, let $\bar{\bar{\mathbf{P}}}$ denote an invertible transformation matrix (Gram matrix) between the RWG basis and the (weighted) solenoidal/nonsolenoidal basis, let $\bar{\mathbf{V}}_i^* = \bar{\bar{\mathbf{P}}}^{?}\bar{\mathbf{V}}_i$, $\bar{\mathbf{I}}_i^* = \bar{\bar{\mathbf{P}}}^{-1}\bar{\mathbf{I}}_i$

and $\bar{\bar{\mathbf{Z}}}_k^* = \bar{\bar{\mathbf{P}}}^? \bar{\bar{\mathbf{Z}}}_k \bar{\bar{\mathbf{P}}}$ denote the excitation vector, current coefficient vector and MOT matrix after basis transformation, a (low frequency) well-conditioned MOT system can be constructed as

$$\bar{\bar{\mathbf{Z}}}_0^* \bar{\mathbf{I}}_i^* = \bar{\mathbf{V}}_i^* - \sum_{k=1}^{\min\{i-1,k_{max}\}} \bar{\bar{\mathbf{Z}}}_k^* \bar{\mathbf{I}}_{i-k}^* \tag{1.13}$$

In the past, Loop-Star/Tree basis transformation techniques [26-28] were used to efficiently cap the condition number of $\bar{\bar{\mathbf{Z}}}_0^*$ (see the example in Figure 1.2(a)). However, these bases typically yield constant but still high condition numbers when applied to mixed-scale meshes. More recently, hierarchically constructed nonsolenoidal basis is proposed to further reduce the condition number of $\bar{\bar{\mathbf{Z}}}_0^*$ [29-31].

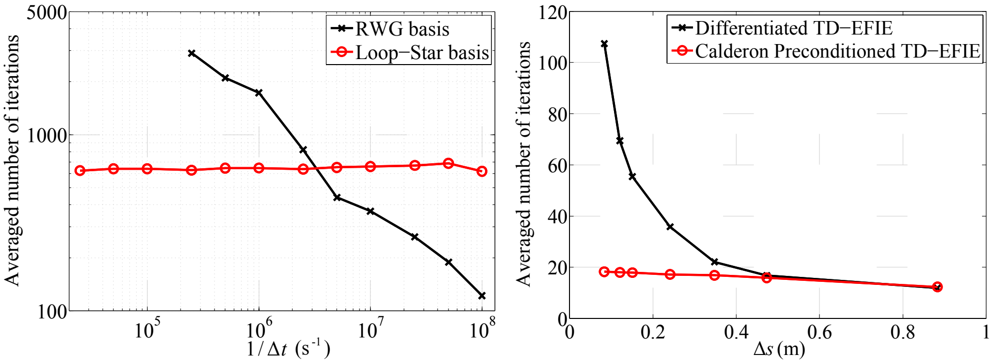
1.2.2.2 Dense mesh breakdown

The dense mesh breakdown occurs in MOT-TD-EFIE solvers for reasons relating to the inconsistent asymptotic behaviors of the EFIE operator’s singular values associated with solenoidal and nonsolenoidal singular functions that can be supported by a dense mesh. As the mesh size decreases, the singular values associated with solenoidal currents go to zero, and those associated with nonsolenoidal currents go to infinity. As a result, the condition number of the discretized TD-EFIE system grows without bound. This type of breakdown is cured by leveraging the self-regularization property of the time domain Calderón identity, viz., the square of the TD-EFIE operator has a well bounded spectrum [32, 33]. The Calderón preconditioned TD-EFIE is

$$-\mathcal{L}_e[\hat{\mathbf{n}} \times \mathbf{E}^i](\mathbf{r},t) = \eta_0 \mathcal{L}_e^2[\mathbf{J}](\mathbf{r},t), \tag{1.14}$$

and can be discretized using the RWG functions and the divergence (and quasi-curl) conforming Buffa-Christiansen (BC) functions [32-35]. The effect of eliminating dense mesh breakdown using the Calderón preconditioned TD-EFIE is demonstrated by the example in Figure 1.2(b).

The MOT-TD-MFIE is free from dense mesh breakdown as the MFIE operator has a bounded singular spectrum, however, the MOT-TD-CFIE is not, due to the presence of the EFIE operator. A Calderón preconditioned TD-CFIE that gives rise to bounded condition number irrespective of mesh density is proposed in [32]. Moreover, a Calderón preconditioned single source TDIE for analyzing transient scattering from homogeneous dielectric has been developed [36].



(a) (b)

Figure 1.2 (a) Number of iterations using different time step sizes and spatial basis functions for analyzing scattering from a PEC sphere discretized with 32 planner triangles. (b) Number of iterations using differentiated TD-EFIE and Calderón preconditioned TD-EFIE for analyzing scattering from a PEC sphere discretized with different mesh sizes.

1.2.3 Accurate MOT-TDIE Solvers

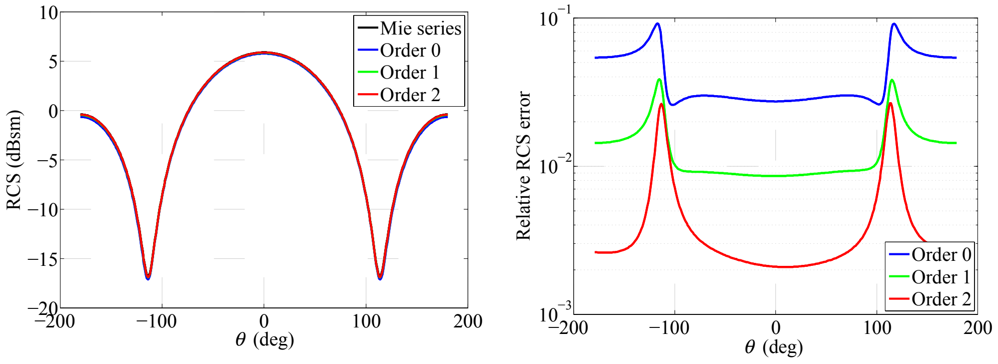
Inaccurate discretization of the TDIEs can cause erroneous solutions and late-time instabilities. Specifically, the discretization of the TDIEs includes the space-time current discretization in (1.7) and the computation of the MOT matrices in (1.9). The former is discussed in this subsection as it has direct impact on the accuracy of the MOT-TDIE solvers. In contrast, the latter affects more the stability of the solvers and is discussed in the next subsection.

1.2.3.1 Accurate temporal discretization

The local Lagrange temporal basis functions, though commonly used, can cause numerical errors due to their discontinuous derivatives at integer multiples of Δt [Figure 1.1(a)-(d)]. Alternatively, other temporal basis functions such as the first order continuous cosine square function [37], all order continuous exponential function [38] and smooth B-spline functions [39-41] have been proposed [Figure 1.1 (e),(f)]. Note that these functions satisfy the discrete causality condition required for the time marching, viz., $T(t)=0$ for $t \leq -\Delta t$. In addition, the approximate prolate spheroidal wave function (APS) [42] has been used as the temporal basis function [43] as its bandlimitedness property permits interpolation with spectral accuracy [Figure 1.1 (g) and 1(h)]. However, the APS function, locally supported on $[-p\Delta t, p\Delta t]$ with $p = 5 \sim 10$, is noncausal and requires carefully designed extrapolation technique to retrieve the form of MOT in (1.8) [43].

1.2.3.2 High order spatial discretization

High order spatial basis function (and high order geometry modeling) is an efficient method to avoid refined mesh to achieve the prescribed solution accuracy, or equivalently, it improves accuracy super-linearly given a fixed mesh. Among many choices of high order spatial basis functions, the most popular one is the divergence conforming Graglia-Wilton-Peterson (*GWP*) function [44] constructed by the product of a scalar polynomial of the given order and the RWG basis function. Authors in [27] developed higher order TD-EFIE/MFIE/CFIE solvers with *GWP* functions and the Loop-Tree decomposition of the pertinent function space [45]. As an example, the bistatic radar cross section (RCS) of a PEC sphere at 30 MHz computed using a TD-CFIE solver with fourth-order Lagrange temporal basis and *GWP* spatial basis of orders 0, 1, 2 is compared with the Mie series solution [Figure 1.3]. The effect of accuracy improvement using *GWP* basis functions is clearly demonstrated. More recently, high order Calderón preconditioned TD-EFIEs [46] are developed that leverages *GWP* functions and high order divergence (and quasi-curl) conforming BC functions [47]. Other methods to improve the spatial discretization accuracy in TDIE solvers include high order Nyström method [48] and Generalized Method of Moments using mixed spatial basis functions [49], etc.



(a) (b)

Figure 1.3 Bistatic RCS at 30 MHz of a PEC sphere of radius 1 m discretized with 32 curvilinear triangles obtained from the TDIE solver with 4^{th} order Lagrange temporal basis and *GWP* spatial basis of orders 0,1,2. (a) Bistatic RCS. (b) Relative RCS errors compared to the Mie series solution

1.2.4 Stabilized MOT-TDIE Solvers

The MOT-TDIE solvers are oftentimes plagued by instabilities, viz., the presence of non-decaying solutions $\bar{\mathbf{I}}_i$ with respect to decaying excitation vectors $\bar{\mathbf{V}}_i$. The stability behavior of a MOT-TDIE solver can be studied by rewriting equation (1.8) in the form of a difference equation [50]

$$\bar{\mathbf{I}}_i^c = \bar{\bar{\mathbf{Z}}}^c \bar{\mathbf{I}}_{i-1}^c \tag{1.15}$$

where $\bar{\mathbf{I}}_i^c = (\bar{\mathbf{I}}_i^T, \bar{\mathbf{I}}_{i-1}^T, \ldots, \bar{\mathbf{I}}_{i-k_{max}+1}^T)^T$ and the companion matrix $\bar{\bar{\mathbf{Z}}}^c$ consists of matrix blocks $\bar{\bar{\mathbf{Z}}}_0^{-1}\bar{\bar{\mathbf{Z}}}_1, \bar{\bar{\mathbf{Z}}}_0^{-1}\bar{\bar{\mathbf{Z}}}_2, \ldots, \bar{\bar{\mathbf{Z}}}_0^{-1}\bar{\bar{\mathbf{Z}}}_{k_{max}}$, and zero/identity blocks of size $N_s \times N_s$. According to the eigenvalue distribution of $\bar{\bar{\mathbf{Z}}}^c$, instabilities of the MOT-TDIE solvers can be classified into three categories: (i) eigenvalues outside the unit circle lead to wildly oscillating and exponentially growing solutions (high frequency instability); (ii) eigenvalue of $1+i0$ corresponds to constant or slowly growing solutions (DC instability); (iii) eigenvalues of form $e^{i\theta}$, $\theta \neq 0$ relate to harmonic solutions with oscillating frequencies corresponding to the interior resonance modes (resonant instability). Among these three types of instabilities, high frequency instability is mainly due to the numerical discretization errors. In contrast, DC and resonance instabilities are more rooted in the spectral property of the pertinent TDIE operators.

1.2.4.1 High frequency instability

In the past, high frequency instability in MOT-TDIE solvers was partially remedied using methods such as temporal/spatial averaging [51-56], space-time Galerkin testing [57, 58], specific collocation-in-time scheme [59], implicit time stepping [60, 61], the Laplace/Z-transform-based scheme [62-64] and accurate space-time discretization schemes already discussed in Section 1.2.3. More recently, methods that permit highly accurate evaluation of the MOT matrix elements and hence prevent undesirable eigenvalue shifting of $\bar{\bar{\mathbf{Z}}}^c$, are developed to stabilize the MOT-TDIE solvers.

To be specific, the MOT matrix element $\{\bar{\bar{\mathbf{Z}}}_k\}_{mn}$ in (1.9) represents the electric/magnetic field generated by space-time basis function $T_{-k}(t)\mathbf{S}_n(\mathbf{r})$ and tested by spatial basis function $\mathbf{S}_m(\mathbf{r})$ at time $t=0$, which involves two source spatial integrals and two test spatial integrals. Accurate evaluation of these four spatial integrals can be very challenging using pure numerical quadrature rules. Recently, semi-analytical methods that analytically evaluate the two [18, 65] or three [66] out of the four spatial integrals in (1.9) and numerically evaluate the rest ones by standard quadrature rules, are developed. These methods assume the usage of RWG spatial basis functions and Lagrange temporal basis functions. Methods that allow more flexible choices of temporal basis functions are developed leveraging closed-form evaluation of the electric [67], magnetic [68, 69] and combined fields [70] due to impulse excited RWG spatial basis functions. Other methods for accurate evaluation of the MOT matrix include polar integration [71], fully numerical integration based on a separable approximation of the convolution kernel [72] and radial source integration/smoothed test integration [73].

1.2.4.2 DC instability

DC instability, which occurs mostly in MOT-TD-EFIE solvers, is caused by the presence of static (or linear-in-time) solenoidal currents that reside in the null space of TD-EFIE operator $\mathcal{L}_e$ (or its differentiated form $\dot{\mathcal{L}}_e$). The first effort at eliminating this type of instability was reported in [27, 43] that leverages the Loop-Tree decomposition to extract static solenoidal current and add it back to the solution after solving the system equation (1.8). Later, the same authors proposed augmentation of the standard TD-EFIE with a differentiated normal field MFIE [28, 74]

$$\hat{\mathbf{n}} \cdot \dot{\mathbf{H}}^{i}(\mathbf{r},t) = -\hat{\mathbf{n}} \cdot \dot{\mathbf{H}}^{s}(\mathbf{r},t), \tag{1.16}$$

as the enforcement of (1.16) suffices to, in practice, induce zero normal magnetic fields and zero static static solenoidal current during time marching.

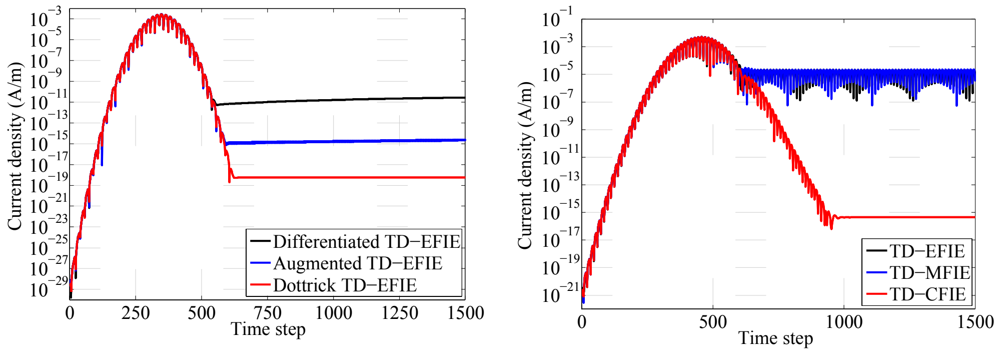
More recently, a "Dottrick" scheme that totally removes the eigenvalue $1+i0$ of $\bar{\bar{\mathbf{Z}}}^{c}$ is proposed based on the Calderón preconditioned TD-EFIE (1.14) and a judicious rearrangement of the time differentiation and integration operations in the operator $\mathcal{L}_{e}^{2}$ [32]

$$\mathcal{L}_{e}^{2}=\mathcal{L}_{es}^{2}+\dot{\mathcal{L}}_{eh}\tilde{\mathcal{L}}_{es}+\tilde{\mathcal{L}}_{es}\dot{\mathcal{L}}_{eh} \tag{1.17}$$

where $\tilde{\mathcal{L}}_{es}[\mathbf{J}](\mathbf{r},t)=-\left(\hat{\mathbf{n}}\times\int_{S}ds'\mathbf{J}(\mathbf{r}',\tau)/R\right)/4\pi c_{0}$. Figure 1.4(a) compares the (DC) stability behaviors of the standard, normal magnetic field augmented and Dottrick TD-EFIEs.

1.2.4.3 Resonant instability

The solutions of TD-EFIE and MEIE are oftentimes corrupted by resonant currents that reside in the null space of the TD-EFIE/MFIE operators. Although in theory these resonant modes are not supported by the zero initial current condition, in practice they can be excited due to numerical errors accumulated during the time marching. Not surprisingly, accuracy improvement in the iterative solver and evaluation of the MOT matrix elements can suppress the resonant instability [75]. The TD-CFIE, as already mentioned in Section 1.2.1.1, eliminates the eigenvalues $e^{i\theta}$, $\theta \neq 0$ of $\bar{\bar{\mathbf{Z}}}^{c}$ and is free from resonant instability [1]. The stability behaviors of TD-EFIE/MFIE/CFIE are compared via an example in Figure 1.4(b).



(a)

(b)

Figure 1.4 (a) Currents on a PEC sphere of radius 1 m obtained using standard TD-EFIE and DC stabilized TD-EFIEs. The excitation waveform is a modulated Gaussian pulse with central frequency $f_0 = 8$ MHz and bandwidth $f_{bw} = 3$ MHz. (b) Currents on a PEC sphere of radius 1 m obtained using standard TD-EFIE, TD-MFIE and TD-CFIEs. The excitation waveform is a modulated Gaussian pulse with central frequency $f_0 = 120$ MHz and bandwidth $f_{bw} = 80$ MHz..

1.2.5 Fast MOT-TDIE Solvers

The computationally most demanding operation in abovementioned MOT-TDIE solvers is the evaluation of the sum on the RHS of (1.8) during time marching, which requires computation of tested fields at N_s observers due to N_s sources for N_t time steps. The computational cost of this operation, if performed directly, is prohibitively high and hinders the application of MOT-TDIE solvers to transient problems involving electrically large objects. Indeed, when applied to problems that involves 3D objects residing in unbounded lossless medium, the computational and memory costs of this operation scale as $O(N_t N_s^2)$ and $O(N_s^2)$; when applied to problems involving 3D objects embedded in dissipative or structured environment, or 2D objects, these costs raise to $O(N_t^2 N_s^2)$ and $O(N_t N_s^2)$, respectively. Note: the latter estimates are valid due to the infinite temporal tail of the Green's function in these media.

In the past, the computational efficiency of MOT-TDIE solvers has been significantly improved by various fast algorithms. The most popular ones among them, viz, TD-AIM, NGTD and PWTD, are discussed in this subsection. Other methods to reduce the computational and/or memory costs of the MOT scheme include the accelerated Cartesian expansion (ACE)-based algorithm that is well-suited for accelerating low frequency integral kernels [76, 77], wavelet-based adaptive MOT scheme [78], envelope tracking technique that permits large time step size in the high frequency regime [79, 80] and hybridization of TDIE methods with physical optics (PO) methods [81-85] and DE methods [86-90].

1.2.5.1 TD-AIM

Just like its frequency domain counterpart [91], TD-AIM permits fast evaluation of the radiated fields by projecting them onto auxiliary uniform spatial grids and propagating them using fast Fourier transforms (FFTs). However, unlike the frequency domain AIM that leverages space-only FFTs, TD-AIM often utilizes multilevel/blocked space-time FFTs as the sparse structure of the MOT matrices need to be accounted for [9, 92]. When applied to transient analyses involving quasi-planar surface scatterers in unbounded lossless medium under high frequency excitations, the computational and memory costs of TD-AIM scale as $O(N_t N_s \log^2 N_s)$ and $O(N_s^{1.5})$, respectively [9]; for more general surfaces, these costs become $O(N_t N_s^{1.5} \log^2 N_s)$ and $O(N_s^2)$ [9]. Moreover, TD-AIM can be applied, with minimal modifications, to transient analyses involving surfaces embedded in lossy medium [22] or half space [23]. In [93], TD-AIM is extended to the low frequency regime through accelerating the computation of both the RHS and left hand side of (1.8) by the space-time and space-only FFTs, respectively. The computational costs of these TD-AIM-accelerated MOT-TDIE solvers are listed in Table 1.1. Note: although TD-AIM is asymptotically inferior to other fast algorithms such as PWTD and NGTD, it remains very competitive for many practical problems. Moreover, TD-AIM has been applied to accelerate a hybrid field-circuit simulator that couples the surface-volume TDIE solver with the modified nodal analysis (MNA)-based circuit solver [94]. More recently, a TD-AIM-accelerated multiconductor transmission line (MTL) simulator that further combines the TDIE solver, TDIE-based MTL solver and

MNA-based circuit solver is developed for electromagnetic characterization of complex structures [4, 5, 95].

1.2.5.2 NGTD

The NGTD algorithm accelerates computation of fields produced by temporally bandlimited, space-confined sources by representing the delay- and amplitude-compensated fields at a sparse grid surrounding the observers and evaluating the true fields through interpolation and delay/amplitude restoration [96]. The two-level NGTD algorithm was first developed using the spherical nonuniform grid [96]. Later, multilevel NGTD based on the Cartesian nonuniform grid was developed [10]. When applied to field computation due to either surface-bound or volumetrically distributed sources that reside in unbounded lossless medium, the computational costs of multilevel NGTD scale as $O(N_t N_s \log^{\mu} N_s)$. Here, $\mu = 2$ in the high frequency regime, $\mu = 1$ in the low frequency regime and $1 < \mu < 2$ for mixed-scale mesh [Table 1.1]. Moreover, NGTD is remarkably simple to implement compared with other fast algorithms.

	Surface			Volume; high frequency
	Unbounded, lossless medium; high frequency	Unbounded, lossless medium; low frequency	Structured/ dissipative medium; high frequency	
Direct	$N_t N_s^2$	$N_t N_s^2$	$N_t^2 N_s^2$	$N_t N_s^2$
TD-AIM	$N_t N_s \log^2 N_s$	$N_t N_s \log^2 N_s$	$N_t N_s \log N_t N_s \log N_t$	$N_t N_s \log^2 N_s$
NGTD	$N_t N_s \log^2 N_s$	$N_t N_s \log N_s$	–	$N_t N_s \log^2 N_s$
PWTD	$N_t N_s \log^2 N_s$	$N_t N_s \log N_s$	$N_t N_s \log N_t \log N_s$	$N_t N_s$

Table 1.1 Best achievable estimates (multiplicative constants omitted) of the computational costs for computing the RHS sum of (1.8) using the direct scheme and fast algorithms.

1.2.5.3 PWTD

This section reviews the advances in PWTD algorithms and their applications to various transient electromagnetic problems. First, the PWTD algorithm and its extensions applicable to problems involving different objects, background media and frequency regimes are summarized. Next, a few application examples of the PWTD-accelerated MOT-TDIE solvers are provided.

- *PEC objects in lossless medium.* PWTD permits fast evaluation of interactions in (1.9) between spatial basis functions that reside in well-separated group pairs. It expands the fields due to time-limited and bandlimited current density in the source group into a set of homogeneous plane waves propagating in the background medium. When implemented in a two-level framework (i.e., all groups have equal sizes), the PWTD algorithm permits efficient evaluation of RHS of (1.8) using $O(N_t N_s^{1.5} \log N_s)$ CPU and $O(N_s^{1.5})$ memory resources [97]. Moreover, when implemented in a multilevel framework, (i.e., a so-called PWTD tree needs to be constructed to efficiently account for interactions between group pairs of different sizes), the computational cost of the PWTD-accelerated TD-SIE solver can be further reduced to $O(N_t N_s \log^2 N_s)$ [13].
- *Homogeneous dielectrics in lossless medium.* In the (multilevel) PWTD-accelerated TD-SIE solvers for analyzing transient scattering from (piecewise) homogeneous dielectrics residing in unbounded lossless medium, the PWTD algorithm accelerates computation of the fields in each dielectric region due to surface electric and magnetic currents [15]. Furthermore, due to different wave speed in each region, multiple PWTD trees need to be constructed. The computational and memory costs of these PWTD-accelerated TD-SIE solvers scale as $O(N_t N_s \log^2 N_s)$ and $O(N_s^{1.5})$. These cost estimates, though seemingly similar to those of the abovementioned solvers for analysis scattering from PEC

scatterers, have larger leading constants due to the presence of double surface currents and multiple PWTD trees. In [98], a PWTD-accelerated TD-SIE solver applicable to composite scatterers that involve piecewise homogeneous dielectrics and PEC structures, is developed.

- *Inhomogeneous dielectrics in lossless medium.* In the PWTD-accelerated TD-VIE solvers for analyzing transient scattering from inhomogeneous dielectrics residing in unbounded lossless medium, the PWTD algorithm permits fast computation of the fields due to both the electric flux density and the total field contributions in the polarization current $\mathbf{J}^P(\mathbf{r},t)$ [19-21]. Unlike the above-described PWTD algorithms, here the dielectric volume (instead of the surfaces) of the scatterers is subdivided using the PWTD tree. The computational and memory costs of the PWTD-accelerated volume TD-VIE solvers scale as $O(N_t N_s)$ and $O(N_s \log N_s)$, respectively [Table 1.1].
- *Surface scatterers in lossy medium.* As discussed at the beginning of Section 1.2.5, the computational cost of the direct TD-SIE solvers for the analysis of scattering from surface scatterers embedded in unbounded lossy medium scales as $O(N_t^2 N_s^2)$ due to the infinite temporal tail of the lossy medium Green's function. This cost can be reduced leveraging two mechanisms: (i) a scalar lossy-medium PWTD algorithm that permits rapid computation of the far field interactions [99], and (ii) a Prony series-based scheme that permits fast temporal convolution of the lossy-medium Green's function with space-time basis functions during the evaluation of near field interactions [100]. Later, the pertinent PWTD-accelerated

TD-EFIE [101] and TD-CFIE solvers [102] are developed. The computational costs of these solvers scale as $O(N_t N_s \log N_t \log N_s)$ [Table 1.1].

- *Mixed-scale scatterers.* The abovementioned PWTD algorithms will lose their computational efficiency when directly applied to mixed-scale scatterers (residing in unbounded lossless medium) due to their inefficiency of field computation in the dense mesh region. An adaptive MOT-TDIE solver that leverages the standard PWTD algorithm in the electrically large region and a low frequency PWTD algorithm in the dense mesh region, was developed [103, 104]. As the computational cost of the low frequency PWTD algorithm scales at most as $O(N_t N_s \log N_s)$, that of the overall solver can be efficiently capped by $O(N_t N_s \log^2 N_s)$.

In addition to the abovementioned PWTD-accelerated MOT-TDIE solvers, those applicable to transient scattering problems that involve surface scatterers embedded in half space or layered medium [105], periodic structures [106], 2D objects [24, 107, 108] have been developed.

These PWTD algorithms have been applied successfully to many large-scale transient electromagnetic problems including large transient scattering problems, broadband antenna problems and EMC problems. In addition, the PWTD algorithm has been applied to construct fast boundary kernels for the FDTD algorithms [109-111], and to accelerate large-scale acoustic scattering [112, 113] and electrodynamics [114] problems, etc.

- *Transient scattering problems.* The PWTD-accelerated TDIE solvers have been widely applied to transient scattering problems that involve large and complex targets. In the past, scattering from real-life targets (e.g., aircrafts and vessels) that involve 10^5 spatial unknowns was analyzed using serial implementation of these solvers [13, 97]. A parallel PWTD-accelerated TDIE solver was used to solve electrically large problems involving half million spatial unknowns [115].

- *Broadband antenna problems.* TDIE methods are well-suited to analyze electromagnetic scattering and radiation from broadband antennas. In the past, direct TDIE solvers were applied to thin wire antennas [25, 116, 117] and small 3D antennas [118, 119], etc. In addition, radiation from antennas mounted on electrically large (yet geometrically simple) platforms have been studied using hybrid TDIE-PO solvers [84, 85]. The PWTD-accelerated MOT-TDIE solvers, when augmented with the aforementioned stability improvement techniques, can be used to analyze large and complex antenna radiation problems. EMC/EMI problems
- *EMC/EMI problems.* Real-life EMC/EMI problems often involve complex and multi-scale structures, e.g., radiation components, cables and microwave circuits that reside in (electrically large) shielding enclosures. MOT-TDIE solvers capable of accurately and efficiently modeling PEC surfaces/wires/junctions, homogeneous/inhomogeneous dielectrics, and linear/nonlinear lumped elements are required. In [120], a PWTD-accelerated TDIE solver was applied to the EMC/EMI analysis involving PEC surfaces and wires. Parallel implementation of this solver was developed in [121]. In addition, a PWTD-accelerated hybrid surface-volume TDIE solver capable of modeling composite structures was developed [122]. More recently, EMC/EMI problems involving microwave circuits have been analyzed using a field-circuit simulator that couples the

PWTD-accelerated TDIE solvers capable of effectively modeling volumes/surfaces/wires/junctions, and the MNA-based circuit solver [123].

1.3 Advances Proposed by This Work

Although PWTD-accelerated MOT-TDIE solver have been successfully applied to the transient analysis of electromagnetic scattering from various objects, that involves up to half million spatial unknowns, they still lag behind the MLFMA-accelerated FDIE solvers in their capabilities of solving real-life electromagnetic problems that involve millions of spatial unknowns. The main limitations of the existing PWTD-accelerated TDIE solvers are:

- Parallel PWTD-accelerated TDIE solvers do not scale well when executed using massive computing resources, mainly due to the lack of an efficient parallelization scheme for the heterogeneous computation and memory loads in PWTD.
- PWTD-accelerated TDIE solvers require much more memory and computational resources than the MLFMA-accelerated FDIE solvers when applied to scattering problems that involves same number of spatial unknowns. To reduce the gap between these two classes of solvers, the temporal, spatial and angular sparsity pertinent to the PWTD algorithm need to be exploited.
- Compared to the widely used TD-SIE solvers, the TD-VIE solvers still remain under development. The capability of existing PWTD-accelerated TD-VIE solvers is limited by their poor stability and/or computational inefficiency.

This work presents the following contributions:

- A provably scalable parallelization scheme for the multilevel PWTD algorithm. The proposed parallel PWTD algorithm scales well on thousands of CPU processors and permits fast evaluation transient fields due to tens of millions of surface-bounded and over one hundred million volumetrically distributed source constellations (Chapter 2).

- A Graphics Processing Unit (GPU) implementation of multilevel PWTD algorithm that achieves significant speedups when compared to serial CPU implementations of the PWTD algorithm (Chapter 3).
- A parallel PWTD-accelerated TD-SIE solver capable of analyzing transient scattering from canonical and real-life PEC objects that involve 10 million spatial unknowns (Chapter 4).
- A wavelet-enhanced PWTD-accelerated TD-SIE solver that exploits the temporal sparsity in the PWTD algorithm. The proposed solver is capable of analyzing transient scattering from smooth quasi-planar PEC objects spanning well over one hundred wavelengths (Chapter 5).
- An explicit and PWTD-accelerated TD-VIE solver capable of analyzing transient scattering from canonical and real-life dielectric objects that involve 25 million spatial unknowns (Chapter 6).

CHAPTER 2

A Provably Scalable Parallel Multilevel PWTD Algorithm

2.1 Chapter Introduction

The analysis of transient fields produced by temporally bandlimited and space-confined source constellations is of paramount importance in various disciplines including acoustics, electromagnetics (EM), elastodynamics, and quantum dynamics. The computational complexity and memory requirement of performing this analysis via classical schemes scale as $O(N_t N_s^2)$ and $O(N_s^2)$, respectively, for N_s sources active for N_t time-steps. The multilevel PWTD algorithm reduces the computational and memory requirements of this analysis to $O(N_t N_s \log^2 N_s)$ and $O(N_s^{1.5})$ for the surface-bound sources [8], as well as to $O(N_t N_s)$ and $O(N_s \log N_s)$ for the volumetrically-distributed sources [19]. These computational cost and memory requirement of the PWTD algorithm permit its applications to the fast and accurate analyses of transient acoustic/elastodynamic/EM scattered fields when it is used in conjunction with classical MOT schemes [13, 15, 19-21, 113, 114]. Despite of its favorable computational and memory requirements, the application of serial implementation of the PWTD algorithm cannot be extended beyond the transient analyses involving medium scale source constellations. To this end, parallelization of the PWTD algorithm for executing on distributed memory supercomputers is imperative for its applications to the transient analyses involving (very) large-scale source constellations, that is the case in real-world transient problems.

Parallelization of the PWTD algorithm for distributed memory supercomputers is a non-trivial task since the computation and memory loads pertinent to spatial, angular, and temporal dimensions of the PWTD algorithm differ at each level of the PWTD tree. To date, a spatial partitioning-based [121] and a hybrid spatial/angular partitioning-based [115] parallelization schemes have been proposed for the PWTD algorithm. Unfortunately, these parallelization schemes are not rigorously proven to be scalable and consequently exhibit unfavorable load balance and low computation-to-communication ratios (CCRs) at certain levels of the PWTD tree when executed on a large number of processors. Similar spatial and hybrid spatial/angular partitioning-based parallelization schemes were proposed for MLFMA, viz. frequency domain counterpart of PWTD algorithm [124-126], yet not proven to be scalable. The first provably scalable techniques for parallelizing the MLFMA did not appear in the archival literature until 2008 [127-131]. These techniques make use of a hierarchical partitioning strategy that simultaneously leverages spatial *and* angular partitioning at each level of the MLFMA tree [127-131].

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 the two important differences between MLFMA and PWTD: (i) 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. (ii) MLFMAs often use local schemes to interpolate and filter/anterpolate far fields, while PWTD implementations usually rely on exact global spherical interpolation/filtering schemes [13]. 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 chapter 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 load balancing among processors, the proposed scheme produces scalable communication patterns among processors at all levels of the PWTD tree. The load balance and scalability of the proposed scheme are theoretically proved and numerically validated through the tests performed on thousands of multi-core processors. The efficacy and capability of the proposed scheme are demonstrated through its applications to the evaluation of transient scalar/vector fields generated by tens of millions of surface-bound and volumetrically-distributed scalar/vector sources.

The rest of the chapter is organized as follows: In Section 2.2, PWTD algorithm for rapidly evaluating transient scalar and vector fields generated by temporally bandlimited and space-confined source constellations is summarized. The proposed parallelization scheme of the PWTD algorithm is expounded in Section 2.3. In Section 2.4, numerical examples that show the scalability and capability of the proposed scheme are presented.

2.2 Formulation

2.2.1 PWTD for Evaluating Transient Scalar Fields

Let $q(\mathbf{r},t)$ denote a scalar source distribution defined on a surface S or in a volume V. The transient scalar field generated by $q(\mathbf{r},t)$ is

$$u(\mathbf{r},t)=\int_{\Omega}q(\mathbf{r}',t)\,\delta(t-R/c_0)/R\,d\mathbf{r}' \qquad (2.1)$$

where $R=|\mathbf{r}-\mathbf{r}'|$ is the distance between the source point $\mathbf{r}'$ and the observation point $\mathbf{r}$, $\delta(\cdot)$ is the Dirac function, c_0 is the speed of wave propagating in the medium, and $\Omega=\{S,V\}$. Assume that $q(\mathbf{r},t)$ consists of N_s point sources as

$$q(\mathbf{r},t)=\sum_{n=1}^{N_s} f_n(t)\delta(\mathbf{r}-\mathbf{r}_n) \tag{2.2}$$

where $\mathbf{r}_n$ is the position of the n^{th} point source with the time signature $f_n(t)$, which is temporally bandlimited to maximum frequency f_{max} and quasi-time-limited to $0<t<T$.

To numerically evaluate transient scalar fields, the time signature $f_n(t)$ is oftentimes expressed in terms of Lagrange polynomials $T_j(t)$, $j=1,\ldots,N_t$ [12], as

$$f_n(t)=\sum_{j=1}^{N_t} q_{n,j}T_j(t) \tag{2.3}$$

where $T_j(t)=T_0(t-j\Delta t)$ (time-shifted), $\Delta t=1/(2\chi_t f_{max})$ is the time-step size with oversampling ratio χ_t, $5<\chi_t<20$, $N_t=T/\Delta t$, and $q_{n,j}=f_n(j\Delta t)$ is the time signature at the j^{th} time-step. Substituting (2.2) and (2.3) into (2.1) and computing the scalar fields at $t=i\Delta t$ results in a matrix system as

$$\bar{\mathbf{U}}_i=\sum_{j=0}^{i}\bar{\bar{\mathbf{Z}}}_j^{sca}\bar{\mathbf{Q}}_{i-j},\qquad i=1,\ldots,N_t \tag{2.4}$$

where the entries of $\bar{\mathbf{U}}_i$, $\bar{\mathbf{Q}}_i$, and $\bar{\bar{\mathbf{Z}}}_j^{sca}$ are $u(\mathbf{r}_m,i\Delta t)$ and $q(\mathbf{r}_m,i\Delta t)$, $m=1,\ldots,N_s$, and

$$Z_{j,mn}^{sca}=\begin{cases} T_{-j}(-|\mathbf{r}_m-\mathbf{r}_n|/c_0)/(4\pi|\mathbf{r}_m-\mathbf{r}_n|), & m\neq n\\ 0 & m=n\end{cases} \tag{2.5}$$

Direct evaluation of scalar fields on N_s point sources [via (2.5)] requires $O(N_tN_s^2)$ operations and $O(N_s^2)$ memory when $\bar{\bar{\mathbf{Z}}}_j^{sca}$ is pre-computed and stored. These computational requirements can be reduced drastically via the PWTD algorithm.

In multilevel PWTD algorithm, a hypothetical box that encloses the source distribution is recursively subdivided into eight smaller boxes until the edge length of the smallest boxes becomes a prescribed fraction of the wavelength at the maximum frequency, $\lambda=c_0/f_{max}$. This recursive subdivision generates a N_L-level PWTD tree, where $N_L=\log O(N_s^{1/\mathrm{d}})$, $d=\{2,3\}$ for $\Omega=\{S,V\}$; the boxes with the same edge length belong to the same level v, $v = 1,...,N_L$. At level v of the PWTD tree, there exist approximately $N_g^v = 2^{d(N_L - v)}$ nonempty boxes, each of which can be enclosed by a sphere of radius $R^v = 2^{v-1}R^1$ with $R^1 = O(1)$. By starting from level N_L (coarsest level), a pair of source and observer boxes, labeled as α and α', centered at $\mathbf{r}_\alpha^c$ and $\mathbf{r}_{\alpha'}^c$, respectively, is identified as a "far field" pair if (i) the distance between their centers, $R_{c,\alpha\alpha'} =| \mathbf{R}_{c,\alpha\alpha'} |=| \mathbf{r}_\alpha^c - \mathbf{r}_{\alpha'}^c |$, is greater than a prescribed threshold, $R_{c,\alpha\alpha'} > \gamma R^v$ ($3 \le \gamma \le 6$), and (ii) their parent boxes are not distinguished as a far field pair. The box pairs at level 1 (finest level) that are not identified as "far field" pairs are labeled as "near field" pairs. The contributions to (2.4) due to the interactions between point sources within the near field pairs are directly evaluated by (2.5). To compute the contributions to (2.4) stemming from the interactions between point sources within a far field box pair (α,α'), first, the time signature of the n^{th} point source in the box α, $f_n(t)$, is broken into N_l^v consecutive subsignals using the APS function as [42]

$$f_n(t) = \sum_l^{N_l^v} f_n^l(t) = \sum_l^{N_l^v} \sum_{j=(l-1)M^v+1}^{lM^v} q_{n,j} T_j^{APS}(t) \tag{2.6}$$

Here, $T_j^{APS}(t) = T^{APS}(t - j\Delta t)$, $N_l^v M^v = N_t$, $T^{APS}(t)$ denotes the APS function, which is temporally bandlimited to $f_s = \chi_t f_{max}$ and approximately time-limited to $-p_f \Delta t < t < p_f \Delta t$, $5 \le p_f \le 10$, M^v is set to an appropriate integer so that the duration of each subsignal $f_n^l(t)$, $T^v = (M^v + 2p_f)\Delta t$, becomes less than $(R_{c,\alpha\alpha'} - 2R^v)/c_0$. The scalar field on the m^{th} point source (in box α') generated by the n^{th} point source (in box α), for the l^{th} subsignal, can be evaluated in three stages. First, a set of outgoing rays along directions $\hat{\mathbf{k}}_{pq}^v$, $q_{l,\alpha}^+(\hat{\mathbf{k}}_{pq}^v, t)$, are constructed by projecting the subsignal $f_n^l(t)$, $n \in \alpha$, onto these directions as

$$q_{l,\alpha}^+\left(\hat{\mathbf{k}}_{pq}^v, t\right) = \sum_{n\in\alpha} \delta\left[t + \hat{\mathbf{k}}_{pq}^v \cdot \left(\mathbf{r}_n - \mathbf{r}_\alpha^c\right)/c_0\right] * f_n^l(t) \tag{2.7}$$

where $*$ denotes temporal convolution and directions $\hat{\mathbf{k}}_{pq}^{v}$, $0 \le p \le K^{v}$, $-K^{v} \le q \le K^{v}$ are selected as quadrature points on a unit sphere [8]. $K^{v} = \lfloor 4\chi_s \pi f_s R^{v} / c_0 \rfloor + 1$ is the number of spherical harmonics with the oversampling factor χ_s. Indeed, the total number of plane wave directions required is $N_k^{v} = (K^{v}+1)(2K^{v}+1)$. Second, the outgoing rays of box α, $q_{l,\alpha}^{+}(\hat{\mathbf{k}}_{pq}^{v},t)$, are translated into the incoming rays of box α', $q_{l,\alpha'}^{-}(\hat{\mathbf{k}}_{pq}^{v},t)$, by convolving the outgoing rays with the translation function $\mathcal{T}^{sca}(\hat{\mathbf{k}}_{pq}^{v},t,K^{v})$, i.e.,

$$q_{l,\alpha'}^{-}\left(\hat{\mathbf{k}}_{pq}^{v},t\right) = \mathcal{T}^{sca}\left(\hat{\mathbf{k}}_{pq}^{v},t,K^{v}\right) * q_{l,\alpha}^{+}\left(\hat{\mathbf{k}}_{pq}^{v},t\right) \tag{2.8}$$

where

$$\mathcal{T}^{sca}\left(\hat{\mathbf{k}}_{pq}^{v},t,K^{v}\right) = -\frac{\partial_t}{16\pi^2 R_{c,\alpha\alpha'}} \sum_{k=0}^{K^{v}} (2k+1)\Phi_k\left(\frac{c_0 t}{R_{c,\alpha\alpha'}}\right)\Phi_k\left(\frac{\hat{\mathbf{k}}_{pq}^{v} \cdot \mathbf{R}_{c,\alpha\alpha'}}{R_{c,\alpha\alpha'}}\right) \tag{2.9}$$

Here, $\Phi_k(\cdot)$ is the Legendre polynomial of degree k, ∂_t denotes time derivative, and $|t| \le R_{c,\alpha\alpha'}/c$. Finally, the scalar field on the m^{th} point source (in box α'), $u^{l}\left(\mathbf{r}_m,t\right)$, is computed by projecting the incoming rays, $q_{l,\alpha'}^{-}(\hat{\mathbf{k}}_{pq}^{v},t)$, onto the m^{th} point source and summing over all directions with the quadrature weights ω_{pq} as

$$u^{l}\left(\mathbf{r}_m,t\right) = \sum_{p=0}^{K^{v}} \sum_{q=-K^{v}}^{K^{v}} \omega_{pq} \delta\left[t - \hat{\mathbf{k}}_{pq}^{v} \cdot \left(\mathbf{r}_m - \mathbf{r}_{\alpha'}^{c}\right)/c_0\right] * q_{l,\alpha'}^{-}\left(\hat{\mathbf{k}}_{pq}^{v},t\right) \tag{2.10}$$

In practice, only outgoing/incoming rays of the boxes at level 1 are computed using (2.8) /(2.10) while those of higher level boxes are obtained using a scalar global spherical interpolation/filtering scheme [132] and performing a local shifting operation. It was proven that the computational complexity and memory requirement of the multilevel PWTD scheme scale as $O(N_t N_s \log^{\kappa} N_s)$ and $O(N_s^{3/d} \log^{\mu} N_s)$, where $\kappa = 2$, $\mu = 0$ for $d = 2$ ($\Omega = S$) and $\kappa = 0$, $\mu = 1$ for $d = 3$ ($\Omega = V$).

2.2.2 PWTD for Evaluating Transient Vector Fields

The PWTD scheme is often used for rapidly evaluating transient vector fields generated by vector sources pertinent to EM scattering problems. Here, we briefly summarize the PWTD algorithm for evaluating vector fields, with a focus on its difference compared with the abovementioned scalar-field PWTD algorithm. Let $\mathbf{J}(\mathbf{r},t)$ represent a current density defined on a surface S or in a volume V. The differentiated electric field $\dot{\mathbf{E}}(\mathbf{r},t)$ generated by $\mathbf{J}(\mathbf{r},t)$ is

$$\dot{\mathbf{E}}(\mathbf{r},t) = \left(\mu_0/(4\pi)\right)\left(\partial_t^2 \mathcal{I} - c_0^2 \nabla\nabla\right) \cdot \int_\Omega \mathbf{J}\left(\mathbf{r}',t\right)\delta\left(t - R/c_0\right)/R \, d\mathbf{r}' \tag{2.11}$$

where μ_0 is the free space permeability and $\mathcal{I}$ is the identity dyad. Assume that $\mathbf{J}(\mathbf{r},t)$ is approximated by N_s surface-bound point dipoles as

$$\mathbf{J}\left(\mathbf{r},t\right) = \sum_{n=1}^{N_s} f_n\left(t\right)\delta\left(\mathbf{r} - \mathbf{r}_n\right)\hat{\mathbf{u}}_n \tag{2.12}$$

Here $\mathbf{r}_n$ and $\hat{\mathbf{u}}_n$ are the n^{th} dipole's position and direction, and $f_n(t)$, just as in (2.2), is its temporal signature, which is band-limited to maximum frequency f_{max} and quasi time-limited to $0 < t < T$. To evaluate interactions between these dipoles, the temporal signature $f_n(t)$ is oftentimes discretized as

$$f_n\left(t\right) = \sum\nolimits_{j=1}^{N_t} I_{n,j} T_j\left(t\right) \tag{2.13}$$

where $I_{n,j} = f_n(j\Delta t)$. Substituting (2.13) and (2.12) into (2.11) and computing fields (excluding self-interactions) at $t_i = i\Delta t$ yields

$$\overline{\mathbf{F}}_i = \sum\nolimits_{j=0}^{i} \overline{\overline{\mathbf{Z}}}_j^{vec} \overline{\mathbf{I}}_{i-j}, \qquad i = 1,...,N_t \tag{2.14}$$

where the entries of $\overline{\mathbf{F}}_i$**,** $\overline{\mathbf{I}}_i$ and $\overline{\overline{\mathbf{Z}}}_j^{vec}$ are $\hat{\mathbf{u}}_m \cdot \dot{\mathbf{E}}(\mathbf{r}_m, i\Delta t)$, $I_{n,i}$, $m,n = 1,\ldots,N_s$ and

$$Z_{j,mn}^{vec} = \left(\mu_0/(4\pi)\right)\hat{\mathbf{u}}_m \cdot \left(\partial_t^2 \mathcal{I} - c_0^2 \nabla\nabla\right) \cdot \hat{\mathbf{u}}_n T\left(t - \left|\mathbf{r} - \mathbf{r}_n\right|/c_0\right) / \left|\mathbf{r} - \mathbf{r}_n\right| \Big|_{t=t_j,\, \mathbf{r}=\mathbf{r}_m}, \quad m \neq n \tag{2.15}$$

with $Z_{j,mn}^{vec}=0$. Direct computation of fields along N_s dipoles via (2.15) requires $O(N_s^2)$ memory and $O(N_t N_s^2)$ operations for N_t time steps. These computational requirements can be reduced significantly via the multilevel vector-field PWTD algorithm.

First, a hypothetical box enclosing the point dipoles is recursively subdivided into smaller boxes and the box pairs are classified as near field and far field ones by following the procedure described in Section 2.2.1. Next, contributions to (2.14) stemming from interactions between dipoles belonging to near field box pairs are computed directly by (2.15). To evaluate contributions to (2.14) due to interactions between dipoles belonging to a far field box pair at level v, the time signature of the n^{th} dipole in box α at level v is broken into N_l^v consecutive band-limited subsignals as

$$f_n(t)=\sum\nolimits_l^{N_l^v} f_n^l(t)=\sum\nolimits_l^{N_l^v}\sum\nolimits_{j=(l-1)M^v+1}^{lM^v} I_{n,j}T_j^{APS}(t) \tag{2.16}$$

where $T_j^{APS}(t)=T^{APS}(t-j\Delta t)$ and $N_l^v M^v=N_t$; M^v is chosen such that the duration of each subsignal, $T^v=(M^v+2p_f)\Delta t$, is less than $(R_{c,\alpha\alpha'}-2R^v)/c_0$. To compute the differentiated electric field along the m^{th} dipole (in box α') generated by the n^{th} dipole (in box α) for the l^{th} subsignal, first, a set of outgoing rays (of box α) in directions $\hat{\mathbf{k}}_{pq}^v$ is constructed by the convolution of the projection function $\mathbf{P}_n^+(\hat{\mathbf{k}}_{pq}^v,t,\hat{\mathbf{u}}_n)$ with the subsignal $f_n^l(t)$ as

$$\mathbf{G}_{l,\alpha}^+\left(\hat{\mathbf{k}}_{pq}^v,t\right)=\sum\nolimits_{n\in\alpha}\mathbf{P}_n^+\left(\hat{\mathbf{k}}_{pq}^v,t,\hat{\mathbf{u}}_n\right)*f_n^l(t) \tag{2.17}$$

Next, the outgoing rays (of box α) are translated into incoming rays (of box α') by the convolution of $\mathbf{G}_{l,\alpha}^+(\hat{\mathbf{k}}_{pq}^v,t)$ with the translator $\mathcal{T}^{vec}(\hat{\mathbf{k}}_{pq}^v,t,K^v)$ as

$$\mathbf{G}_{l,\alpha'}^-\left(\hat{\mathbf{k}}_{pq}^v,t\right)=\mathcal{T}^{vec}\left(\hat{\mathbf{k}}_{pq}^v,t,K^v\right)*\mathbf{G}_{l,\alpha}^+\left(\hat{\mathbf{k}}_{pq}^v,t\right) \tag{2.18}$$

Finally, incoming rays are projected onto the m^{th} dipole by convolving the projection function $\mathbf{P}_m^-(\hat{\mathbf{k}}_{pq}^v, t, \hat{\mathbf{u}}_m)$ with the incoming rays and summing over all directions with quadrature weights ω_{pq} as

$$\hat{\mathbf{u}}_m \cdot \dot{\mathbf{E}}\left(\mathbf{r}_m, t\right) = \sum_{p=0}^{K^v} \sum_{q=-K^v}^{K^v} \omega_{pq} [\mathbf{P}_m^-\left(\hat{\mathbf{k}}_{pq}^v, t, \hat{\mathbf{u}}_m\right)]^T \mathbf{G}_{l,\alpha'}^-\left(\hat{\mathbf{k}}_{pq}^v, t\right). \tag{2.19}$$

In (2.17)-(2.19),

$$\mathbf{P}_{\{m,n\}}^{\pm}(\hat{\mathbf{k}}_{pq}^v, t, \hat{\mathbf{v}}) = \hat{\mathbf{k}}_{pq}^v \times \hat{\mathbf{v}} \delta(t \pm \hat{\mathbf{k}}_{pq}^v \cdot (\mathbf{r}_{\{m,n\}} - \mathbf{r}_{\{\alpha',\alpha\}}^c) / c_0) / 4\pi, \tag{2.20}$$

$$\mathcal{T}^{vec}(\hat{\mathbf{k}}_{pq}^v, t, K^v) = \frac{\mu_0 \partial_t^3}{R_{c,\alpha\alpha'}} \sum_{k=0}^{K^v} (2k+1) \Phi_k \left(\frac{c_0 t}{R_{c,\alpha\alpha'}} \right) \Phi_k \left(\frac{\hat{\mathbf{k}}_{pq}^v \cdot \mathbf{R}_{c,\alpha\alpha'}}{R_{c,\alpha\alpha'}} \right), \tag{2.21}$$

In (2.17), the *vector* outgoing rays, $\mathbf{G}_{l,\alpha}^+(\hat{\mathbf{k}}_{pq}^v, t)$ [as opposed to the *scalar* outgoing rays, $q_{l,\alpha}^+(\hat{\mathbf{k}}_{pq}^v, t)$, in (2.7)], have two transverse components. Consequently, the translation stage is performed for each transverse component separately and the incoming rays are projected onto the m^{th} dipole (in box α') oriented along $\hat{\mathbf{u}}_m$. In addition, the outgoing/incoming rays of boxes at level $v > 1$ are computed via a *vector* global spherical interpolation/filtering scheme [133] instead of the global *scalar* interpolation/filtering scheme [13]. Needless to say, computational and memory resources required by the vector-field PWTD algorithm are twice as large as those required by the scalar-field PWTD algorithm.

2.3 Parallelization of the PWTD Algorithm

This section describes a highly scalable parallelization scheme for the PWTD algorithm. 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 2.2). The memory cost of the proposed parallelization strategy is discussed (Section 2.3). Next, the costs estimates for computations and communications required at different stages of the PWTD algorithm are derived (Sections 2.3.3-2.3.5). Finally, overall computational and communication costs of the proposed parallelization strategy are provided and its scalability is theoretically proven (Section 2.3.6).

2.3.1 A Pedestrian Description

The effective parallelization of the multilevel PWTD scheme calls for a uniform distribution of the near field matrix elements (i.e. near field data) in (2.5)/(2.15) 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 is computed for $N_g^v = O(N_s / 2^{dv})$ boxes (or spatial samples), $N_k^v = O(2^{dv})$ 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 as the highest possible level at which the number of boxes $N_g^{v_b}$ is no less than the number of processors N_p. Then it uses two different partitioning strategies for levels $v \le v_b$ and $v \le v_b$:

- At levels $v \le v_b$, each processor computes and stores the complete ray and near field data for approximately N_g^v / N_p boxes. By doing so, the memory and computation loads are only partitioned along the spatial dimension.

- At levels $v > v_b$, computation and storage of the ray data of one box are distributed among $N_r^v = \lceil N_p / N_g^v \rceil$ 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.

This partitioning strategy is perhaps best described by an example. Consider a five-level PWTD tree that is partitioned among six processors [Figure 2.1]. In Figure 2.1, 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 $v = 1,...,5$, 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., $v \le v_b$, which it is in charge of. In the example in Figure 2.1, 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 is stored in processor 1, no inter-processor communication is needed. Each processor performs the translation stage without inter-processor 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 the example in Figure 2.1, 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 is 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 the example in Figure 2.1, 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 is redistributed among these three processors in temporal dimension, each processor interpolates/filters 1/3 of the temporal samples of that box.

2.3.2 Overall Memory Cost

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. In the proposed strategy, the memory load at each processor can be estimated as follows. For levels $v \le v_b$, each processor stores the ray data of N_g^v / N_p boxes for all $O(N_k^v)$ directions. For levels $v > v_b$, each processor stores the ray data of one box for $O(N_k^v / N_r^v)$ directions, where $N_r^v = \lceil N_p / N_g^v \rceil$ denote the number of processors in charge of partially storing one box's ray data. As the proposed strategy is based on partitioning ray data along its spatial and angular dimensions, its temporal dimension is not split among processors. Consequently, each processor stores $O(M^v) = O(N_s^{1/d} / 2^{N_v - v})$ temporal samples of the ray data for each direction and box that it is responsible for. Hence, the memory requirement for the ray data at each processor, MR, is

$$\begin{aligned} MR &= \sum_{v=1}^{v_b} \frac{N_g^v}{N_p} O\left(N_k^v\right) O\left(M^v\right) + \sum_{v=v_b+1}^{N_L} O\left(\frac{N_k^v}{N_r^v}\right) O\left(M^v\right) \\ &= \frac{N_s}{N_p} O\left(\sum_{v=1}^{N_L} 2^{(3-d)v}\right) = O\left(\frac{N_s^{3/d} \log^{\mu} N_s}{N_p}\right) \end{aligned} \qquad (2.22)$$

It's worthwhile to note here that the memory required to store the information about all boxes' parents, children, far field partners, and near field partners may become considerably large at each processor as the number of sources/boxes (and the dimensions of the hypothetical box enclosing the source constellation) increases. Albeit a fraction of the memory requirement of ray data, the memory required to store this data at each processor can be reduced by its partition among processors. In the proposed strategy, each processor stores only a portion of this data pertinent to the boxes that it is in charge of. Therefore, the memory required to store this information at each processor is negligible.

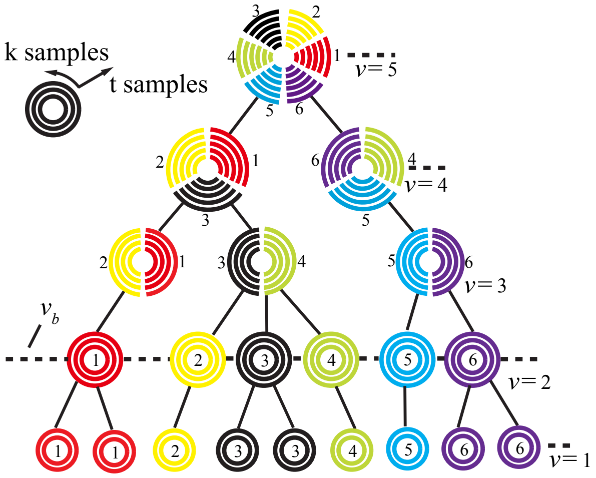


Figure 2.1 Partitioning of boxes and their ray data in a five-level PWTD tree among six processors.

2.3.3 Construction/Projection of Outgoing/Incoming Rays

In the proposed scheme, outgoing ray construction and incoming ray projection are performed via similar procedures. For the sake of brevity, this subsection only details the parallelization of outgoing ray construction .As discussed in Section 2.2, such operations are performed separately at the finest level $v=1$ and at levels higher than finest level (i.e. $v>1$).

At the finest level $v=1$, constructing outgoing rays directly from sources requires no communication. Specifically, one processor is responsible for computing N_l^1 outgoing rays along all N_k^1 directions for approximately N_g^1/N_p boxes. Each outgoing ray is computed at $O(M^1)$ temporal samples. The computational cost of outgoing ray construction at the finest level scales as $O(M^1N_l^1N_k^1\,N_g^1/N_p)=O(N_t\,N_s/N_p)$.

At levels $v>1$, construction of outgoing rays is performed by spherical interpolation and can require communication. Unlike the ray data of boxes at the finest level, the ray data of one box at higher levels can be stored by multiple processors [as in the example in Figure 2.1]. The construction of outgoing rays at higher levels via spherical interpolation can be carried out by considering three possible cases [Figure 2.2 (a)].

- Case 1: the ray data of the box at level $v \le v_b$ is directly interpolated and shifted from the children boxes with no communication. As the spherical interpolation of one temporal sample of the ray data requires $O(N_k^v \log K^v)$ operations, the computational cost for one outgoing ray at level v scales as $O(M^v N_k^v \log K^v)$.
- Case 2: the ray data of the box at $v = v_b + 1$ needs to be stored partially by one processor, while that of the each child box is completely stored by the same or a different processor. This case requires similar operations for the interpolation as in case 1. However, it requires communication before the shifting operation.
- Case 3: ray data of level $v > v_b + 1$ boxes is computed via interpolating and shifting the ray data of the child boxes, which invariably is stored on more than one processor. The construction of the outgoing rays [Figure 2.2(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 [Figure 2.2(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 is sent to the processors in charge of the parent box via non-blocked MPI communication. Step 5: the transferred ray data is locally shifted to the center of the parent box.

For each processor, the computational cost of outgoing ray construction, CC_1, scales as

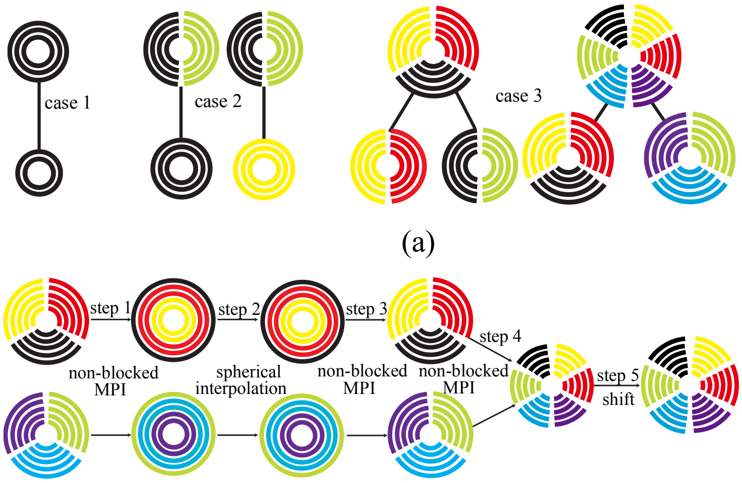
$$\begin{aligned} CC_1 &= O\left(\frac{N_t N_s}{N_p}\right) + \sum_{v=2}^{v_b+1} \frac{N_g^{v-1}}{N_p} N_l^v O\left(M^v N_k^v \log K^v\right) + \sum_{v=v_b+2}^{N_L} N_l^v O\left(\frac{M^v N_k^v \log K^v}{N_r^{v-1}}\right) \\ &= O\left(\frac{N_t N_s}{N_p}\right) + \frac{N_t N_s}{N_p} O\left(\sum_{v=2}^{N_L} v 2^{(2-d)v}\right) = O\left(\frac{N_t N_s \log^{\kappa} N_s}{N_p}\right) \end{aligned} \tag{2.23}$$

where the three contributing terms represent the costs of construction of outgoing rays at finest level, in case 1 and 2 (together), and in case 3, respectively. Note that each processor is in charge of interpolating the ray data of approximately N_g^{v-1} / N_p boxes at level $2 \le v \le v_b + 1$ and at most one child box at level $v > v_b + 1$. Assume that the communication cost per processor is proportional to the total data amount that each processor sends and receives. The communication in this stage is dominated by the data exchange required in steps 1, 3, and 4 of case 3 [Figure 2.2(b)]. At steps 1 and 3, each processor sends/receives partial ray data of size $O(M^v N_k^v / (N_r^{v-1})^2)$ to/from each of the other $N_r^{v-1} - 1$ processors. Therefore, the total data amount that one processor sends/receives at step 1 and 3 is of size $O(M^v N_k^v / N_r^{v-1})$. At step 4, each processor sends/receives ray data of size $O(M^v N_k^v / N_r^{v-1})$ to/from $O(1)$ other processors. Therefore, the cost of communication in this stage, CM_1, scales as

$$\begin{aligned} CM_1 &= \sum_{v=v_b+2}^{N_L} N_l^v O\left(\frac{M^v N_k^v}{N_r^{v-1}}\right) \\ &= \frac{N_t N_s}{N_p} O\left(\sum_{v=v_b+2}^{N_L} 2^{(2-d)v}\right) \le \frac{N_t N_s}{N_p} O\left(\sum_{v=2}^{N_L} 2^{(2-d)v}\right) = O\left(\frac{N_t N_s \log^{\kappa/2} N_s}{N_p}\right) \end{aligned} \tag{2.24}$$

Similarly, the computational cost of the projection of incoming rays, CC_2, and the communication cost in such stage, CM_2, are proportional to those of construction of outgoing rays, i.e.

$$CC_2 \propto CC_1,\ CM_2 \propto CM_1 \tag{2.25}$$



(a)

(b)

Figure 2.2 (a) Three possible cases encountered during the construction of outgoing rays of boxes in PWTD tree. Each case requires different communication patterns. (b) The steps to construct the outgoing rays in case 3.

2.3.4 Translation

The translation between one far field box pair (formulated by (2.8)/(2.18)) is performed by multiplying the Fourier transforms of the translation function and outgoing ray, as well as computing the inverse Fourier transform of the result [8]. The computational cost of this operation (i.e., the fast forward and backward Fourier transforms) scales as $O(M^{v}\log M^{v})$. Note that there exist N_{l}^{v} outgoing rays along each direction for one source box at level v and each observer box only interacts with $O(1)$ source boxes. At level $v \le v_{b}$, each processor performs translation along $O(N_{k}^{v})$ directions for its N_{g}^{v}/N_{p} observer boxes. At the level $v > v_{b}$, each processor carries out translation along $O(N_{k}^{v}/N_{r}^{v})$ directions for at most one observer box. Therefore, the computation cost of the translation stage for each processor, CC_{3}, can be computed as

$$
\begin{aligned}
CC_3 &= \sum_{v=1}^{v_b} \frac{N_g^v}{N_p} N_l^v O(1) O\left(N_k^v\right) O\left(M^v \log M^v\right) \\
&+ \sum_{v=v_b+1}^{N_L} N_l^v O(1) O\left(\frac{N_k^v}{N_r^v}\right) O\left(M^v \log M^v\right) \\
&= \frac{N_t N_s}{N_p} O\left(\sum_{v=1}^{N_L} v 2^{(2-d)v}\right) = O\left(\frac{N_t N_s \log^\kappa N_s}{N_p}\right)
\end{aligned}
\tag{2.26}
$$

where the first and second summations relate to the computation costs of translations at levels $v \le v_b$ and $v > v_b$, respectively. Note that the outgoing ray data of a box at each processor is sent to at most $O(1)$ processors in the translation stage. In fact, some translations do not require communications as those are performed between source and observer boxes possessed by the same processor. For each processor, the ray data sent and received during the translation stage for one box is of size $O(N_k^v M^v)$ for the level $v \le v_b$ and $O(N_k^v M^v / N_r^v)$ for the level $v > v_b$. Therefore, the communication cost in this stage, CM_3, scales as

$$
\begin{aligned}
CM_3 &= \sum_{v=1}^{v_b} \frac{N_g^v}{N_p} N_l^v O\left(N_k^v M^v\right) + \sum_{v=v_b+1}^{N_L} N_l^v O\left(\frac{N_k^v M^v}{N_r^v}\right) \\
&= \frac{N_t N_s}{N_p} O\left(\sum_{v=1}^{N_L} 2^{(2-d)v}\right) = O\left(\frac{N_t N_s \log^{\kappa/2} N_s}{N_p}\right)
\end{aligned}
\tag{2.27}
$$

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 [134]. 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 work flow of this scheme for one processor can be summarized as follows [Figure 2.3]. 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 $v \le v_b$ 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.

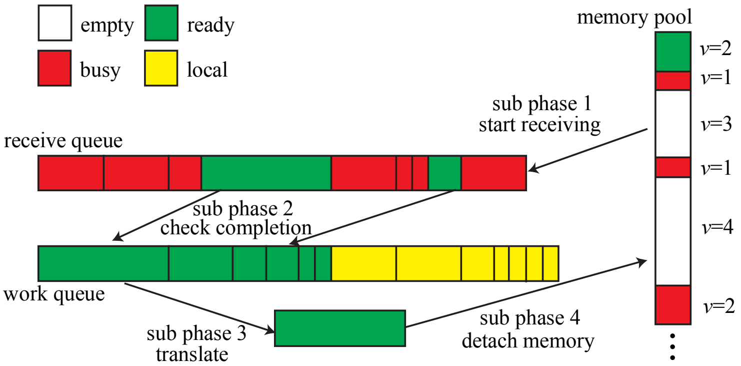


Figure 2.3 Queue-based asynchronous communication during translation stage.

2.3.5 Near field calculation

The interactions between near field box pairs are performed by (2.4) every time-step. In this work, the near field portion of the interaction matrix $\bar{\bar{\mathbf{Z}}}_j^{sca}$ or $\bar{\bar{\mathbf{Z}}}_j^{vec}$ is calculated on-the-fly. At each time-step, since one source box only interacts with $O(1)$ near field observer boxes, the computational and communication costs of near field calculation for one source box scales 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 CC_4 and communication cost CM_4 in near field calculation stage scale as

$$CC_4 \propto CM_4 = \frac{N_g^1}{N_p} N_t O(1) = O\left(\frac{N_s N_t}{N_p}\right). \quad (2.28)$$

2.3.6 Overall computational and communication cost

The above computational and communication cost analysis of PWTD stages (from (2.23) to (2.28)) shows that the overall computational cost CC and communication cost CM of the proposed parallel PWTD algorithm scale as

$$CC = \sum_{n=1}^{4} CC_n = O\left(\frac{N_t N_s \log^{\kappa} N_s}{N_p}\right), \quad CM = \sum_{n=1}^{4} CM_n = O\left(\frac{N_t N_s \log^{\kappa/2} N_s}{N_p}\right). \quad (2.29)$$

It's apparent from (2.22) and (2.29) that the overall memory, computational, and communication costs are inversely proportional to the number of processors N_p. Furthermore, CCR, which equals to $CC / CM = O\left(\log^{\kappa/2} N_s\right)$, is proportional to $O(\log N_s)$ for $\Omega = S$ and $O(1)$ for $\Omega = V$. In other words, as N_s and N_p increase, the parallel PWTD scheme for evaluating fields generated by surface-bound and volumetrically-distributed sources yields non-decreasing CCR. Hence, the scalability of the proposed parallelization strategy is proved.

Test	N_s	Edge length of plate/cube (λ)	R^1 (λ)	Memory per processor (GB)	CPU Time (hour)

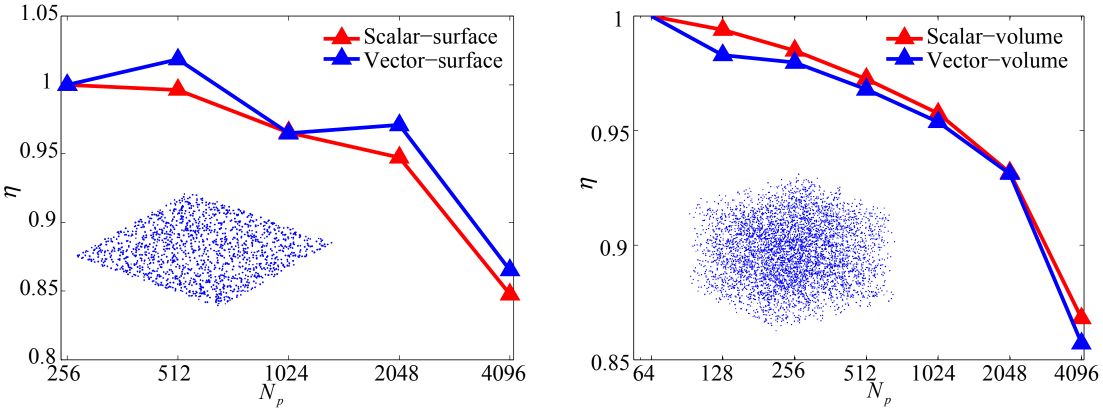
Vector-surface	10,240,000	320	0.54	1.8	2.43
Scalar-surface	20,480,000	452	0.38	2.5	3.04
Vector-volume	64,000,000	40	0.27	0.63	16.2
Scalar-volume	128,000,000	50	0.33	0.75	19.6

Table 2.1 The specifications, memory requirements, and computational times of the tests performed to study load balance.

2.4 Numerical Results

This section presents several numerical tests that demonstrate the scalability, efficacy, and applicability of the proposed parallel PWTD algorithm. In all tests below, a set of point sources or dipoles is randomly located on square plates (surface-bound source, $\Omega = S$) or inside cubes (volumetrically-distributed source, $\Omega = V$). The tests pertinent to evaluating transient scalar and vector fields of surface-bound sources (i.e. point sources or dipoles) are labeled as “scalar-surface” and “vector-surface”, respectively, while those pertinent to evaluating transient scalar and vector fields of volumetrically distributed sources are termed as “scalar-volume” and “vector-volume”, respectively. In all tests below, the temporal signature of each source is assigned to $f_n(t) = m_n \exp(-(t - 6\sigma)^2 / 2\sigma^2)$, where $\sigma = 4/(2\pi f_{max})$, $f_{max} = 1\,\text{GHz}$, and m_n is a random real number between 0 and 1, while the sources are assumed to be active for $N_t = 1000$ time-steps with $\Delta t = 0.0625\,\text{ns}$ in surface tests and $N_t = 500$ time-steps with $\Delta t = 0.0333\,\text{ns}$ for volume tests. All simulations were carried out on a cluster of Quad-Core 850 MHz PowerPC CPUs with 4 GB memory located at Supercomputing Laboratory at the King Abdullah University of Science and Technology. To minimize intra-node communication costs and memory duplications, the parallel PWTD algorithm was implemented using hybrid message passing interface/open multiprocessing (MPI/OpenMP) standards; one MPI process was launched per processor and four cores of each processor were used by OpenMP process.

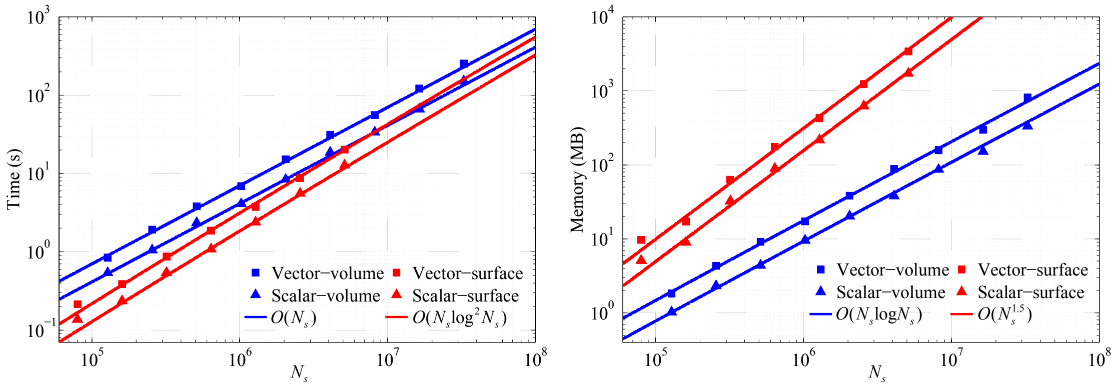
First, parallel efficiency of the proposed parallel PWTD algorithm is demonstrated. The parallel efficiency is defined as $\eta = N_{ref} T_{ref} / N_p T_p$, where T_{ref} is the reference execution time using N_{ref} processors and T_p is the execution time using N_p processors. For surface tests, a set of $N_s = 1{,}000{,}000$ sources are randomly located on a plate with 100λ edge length. The PWTD tree is generated by setting the edge length of the boxes at the finest level to $R^1 = 0.677\,\lambda$. The parallel efficiency of the proposed algorithm is computed when N_p is changed from 256 to 4096 and N_{ref} is set to 256 [Figure 2.4(a)]. For volume tests, a set of $N_s = 1{,}000{,}000$ sources is randomly distributed inside a cube with 10λ edge length. The edge length of the boxes at the finest level of PWTD tree is set to $R^1 = 0.27\,\lambda$. The parallel efficiency of the proposed algorithm is computed when N_p is changed from 64 to 4096 and N_{ref} is set to 64 [Figure 2.4(b)]. It is apparent from Figure 2.4 that over 85% efficiencies have been achieved in all tests. Moreover, the proposed algorithm yields similar parallel efficiencies for evaluating transient scalar and vector fields due to the same load partitioning patterns and CCRs.



(a) (b)

Figure 2.4 Parallel efficiencies when $N_s = 1{,}000{,}000$ sources distributed (a) on a square plate with edge length 100 λ and (b) inside a cube with edge length 10 λ.

Next, the computational time and memory required by the proposed parallel PWTD scheme are examined for increasing N_s when $N_p = 1024$. For surface tests, N_s is increased from $80{,}000$ to $5{,}120{,}000$ while the edge length of the plate is changed from $28.3\,\lambda$ to $226\,\lambda$ and edge length of the boxes at the finest level of PWTD tree is varied from 0.46λ to 0.54λ .The computational costs for a time-step and memory requirements are plotted [Figure 2.5(a)]; those comply with theoretical estimates of $O(N_s \log^2 N_s)$ and $O(N_s^{1.5})$, respectively. For volume tests, N_s is increased from $128,000$ to $32,768,000$ as the edge length of the cube is varied from 5λ to 31.7λ and edge length of the boxes at the finest level is changed from 0.21λ to 0.34λ .The computational costs for a time-step and memory requirements are plotted [Figure 2.5(b)]; those are in good agreement with theoretical predictions of $O(N_s)$ and $O(N_s \log N_s)$, respectively. It's worthwhile to note here that the computational resources required by parallel PWTD scheme for evaluating transient vector fields is two times more than those for evaluating transient scalar fields. In addition, the memory required by scheme in volume tests is significantly lower than that in surface tests.

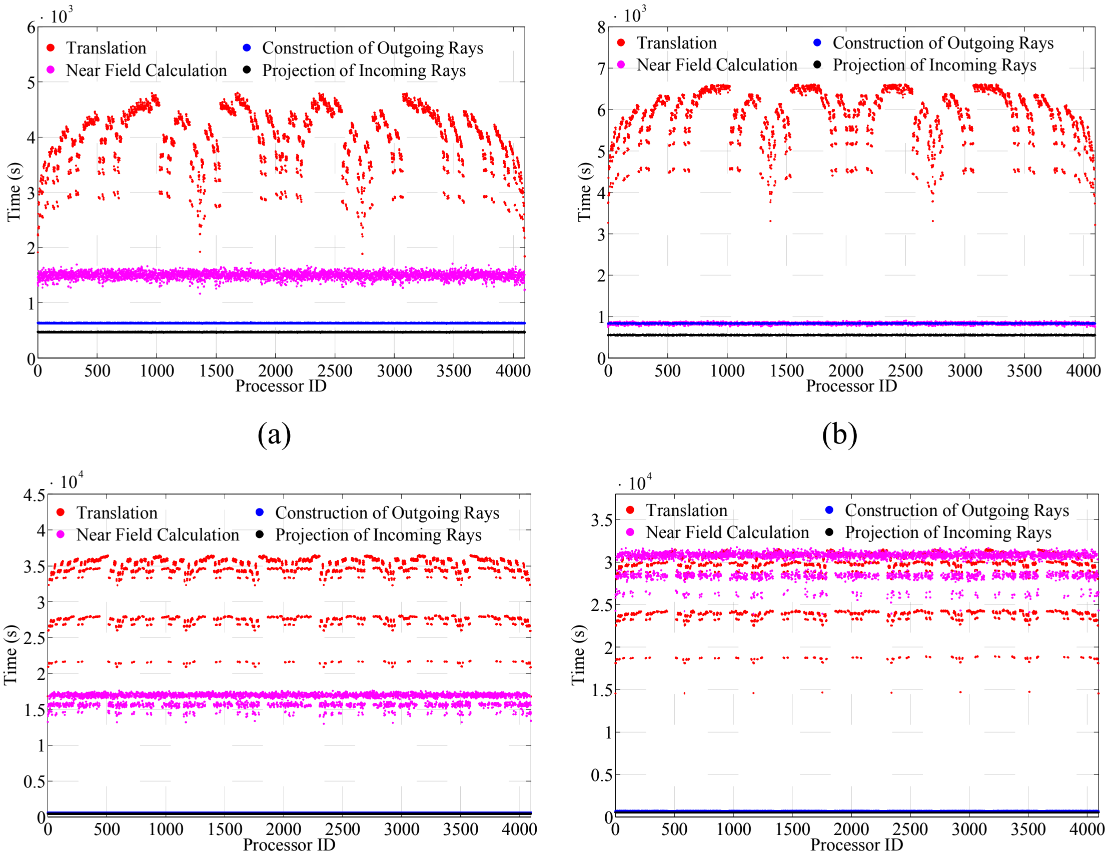


(a) (b)

Figure 2.5 (a) Total computation time and (b) memory requirement in scalar-surface, vector-surface, scalar-volume, and vector-volume tests when $N_p = 1024$ and N_s is changed from $80,000/128,000$ to $5,120,000/32,768,000$ in surface/volume tests.

Finally, the load balance achieved by the proposed parallel PWTD scheme is demonstrated when N_s is set to $10,240,000$, $20,480,000$, $64,000,000$, and $128,000,000$ for vector-surface, scalar-surface, vector-volume, and scalar-volume tests, respectively. The memory requirements, total simulation times, and specifications of tests are provided in Table 2.1. Furthermore, the computation time per processor for each

PWTD stage is plotted [Figure 2.6]. Clearly, excellent load balance is achieved in the near field calculation and ray construction/projection stages in all tests, while good load balance is observed for the translation stage. For the translation stage, the load balance in volume tests [Figure 2.6(c),(d)] appears to be better than that in surface tests [Figure 2.6(a),(b)]. This is because the overall load balance in volume tests ($d=3$) is dominated by those at lower levels of the PWTD tree as the computational cost of translations between level v box pairs scales as $O(v2^{(2-d)v})$ (when $N_p=1$ and N_s and N_t in (2.26) are fixed). The opposite holds in surface tests ($d=2$). And the translation operations can be better split among processors at lower levels of the PWTD tree.



(a)

(b)

(c)

(d)

Figure 2.6 Computation time for each PWTD stage in (a) vector-surface test ($N_s=10,240,000$), (b) scalar-surface test($N_s=20,480,000$), (c) vector-volume test

($N_s = 64,000,000$), and (d) scalar-volume test ($N_s = 128,000,000$).when $N_p = 4096$.

2.5 Chapter Conclusion

A highly scalable parallel PWTD algorithm for rapidly evaluating transient scalar/vector fields generated by large-scale source constellations was presented. The proposed scheme leverages a hierarchical parallelization strategy that divides the computation and memory loads among processors in spatial, angular, and temporal dimensions at all levels of the PWTD algorithm. The scheme employs a mix of non-blocked MPIs, and a queue-based, memory-efficient, and asynchronous communication technique to maintain scalable communications between processors. The scalability of the proposed scheme was validated using up to 4096 processors. The proposed scheme was applied to the analysis of transient fields generated by tens of millions of point sources/dipoles either bound to a surface or volumetrically distributed inside a cube. The proposed parallel PWTD scheme will greatly enhance the capabilities of various MOT-TDIE solvers.

CHAPTER 3

A Graphics Processing Unit Implementation of Multilevel PWTD Algorithm

3.1 Chapter Introduction

The computational complexity for directly evaluating transient fields produced by N_s point sources active for N_t time steps scale as $O(N_t N_s^2)$. This cost can be reduced to $O(N_t N_s \log^2 N_s)$ by the multilevel PWTD scheme [8, 13]. However, serial implementations of the PWTD schemes still impede their applicability to real-life problems. In addition to our work in Chapter 2 to parallelize the multilevel PWTD algorithm using CPU-clusters, this chapter studies the acceleration of the multilevel PWTD algorithm using GPUs.

Recently, GPUs have become popular in computational electromagnetics societies for their unprecedented computational power and unique memory hierarchy, rendering the hardware competitive for parallelizing various computational schemes [135-139]. That said, GPU acceleration of the multilevel PWTD algorithm is nontrivial due to PWTD's heterogeneous structure. Boosting the performance of the PWTD algorithm using GPUs calls for a complete implementation of all PWTD stages, as well as reorganization of the computational and memory management tasks.

Moreover, GPU-clusters, viz., CPU-clusters where each compute-node is augmented with at least one GPU device, have emerged as competitive computing platforms compared with traditional CPU-clusters. Specifically, they take advantages of the massive distributed memory capacity of CPU-clusters as well as provide significantly more computational power at each compute-node. Efficient implementations of the PWTD algorithm on GPU-clusters require scalable distributed parallelization strategy of the overall algorithm and judicious rearrangement of GPU-accelerated sub-tasks at each compute-node.

In this chapter, we first present a complete GPU-accelerated implementation of the multilevel PWTD scheme for evaluating transient vector fields from large-scale dipole constellations. The formulation of the vector-field PWTD algorithm is omitted here as it can be found in Section 2.2.2. GPU implementation strategies for all PWTD stages are detailed and improved. Note that the proposed GPU-acceleration can be applied, with minimal modifications, to the evaluation of scalar fields using the scalar-field PWTD algorithm described in Section 2.2.1. Numerical results show that substantial speedups and memory reduction have been achieved over serial CPU implementations. Next, we demonstrate the efficiency for the GPU-cluster-based implementation of the PWTD algorithm by combining the provably scalable parallelization strategy described in Chapter 2 and the single GPU acceleration technique.

3.2 Single GPU Acceleration

The PWTD scheme for computing dipole interactions consists of four stages: (i) calculation of near field interactions via direct methods; (ii) construction of outgoing rays, (iii) translation of outgoing rays into incoming rays, and (v) processing and projection of incoming rays. These stages are interleaved by (iv) global interpolation and filtering operations. A viable GPU implementation must comprehensively tackle all five computational components of the scheme, which complicates its development compared to other CEM schemes [138, 139]. Below, we delineate key ideas that guided the implementation of these five computational components before embarking on their detailed description.

A NVIDIA GPU consists of many streaming multiprocessors (SMs), each of which contains multiple cores. Under the Compute Unified Device Architecture (CUDA) framework, these SMs can execute a multi-threaded function, termed kernel. A group of 32 threads forms a basic execution unit, dubbed a warp that is dynamically scheduled to one SM. Warps are further combined into blocks such that all threads in one block perform similar tasks that can be synchronized. Moreover, the GPU provides several types of memory, including shared and global units. Shared memory, which is private to one block, has small capacity and low-latency while global memory, which is accessible by all blocks, has large capacity but higher-latency. To alleviate any performance degradation due to the use of high-latency global memory, the following overarching strategies drove the development of our GPU-PWTD implementation: 1) Coalescing memory access of all threads in a warp to contiguous memory addresses by careful arrangement of threads/blocks as well as input/output data layouts. 2) Hiding latency by scheduling more warps while a single warp in the SM is accessing the global memory. This is achieved by assigning sufficiently large number of warps (or equivalently threads and blocks). 3) Minimizing the memory usage by storing only necessary quantities (and calculating all others on-the-fly) and parallelizing loops that require the least global memory access.

3.2.1 Near field Calculation

The interactions between dipoles in near field boxes are computed by (2.15) for every time step via launching one GPU kernel. This operation can be performed on a GPU by parallelizing loops over source and observer boxes, as well as source dipoles and observer dipoles in each box. Note that multiple writing to one entry of $\overline{\mathbf{F}}_i$ (in (2.14)) is required at each iteration of the loops, which are over source boxes and source dipoles in one box. To this end, the loops that are over observer dipoles in a box and observer boxes are parallelized via a "one thread per observer dipole" and "one block per observer box" strategy, producing coalesced global memory access. This strategy can be illustrated by the example in Figure 3.1. Each grid in Figure 3.1 represents interaction between one source dipole and one observer dipole, each rectangular block of grids (boundary marked in bold) represents interactions between dipoles in a near field box pair. In this strategy, each thread (represent one observer dipole) in one block (representing one observer box) computes the interactions between the observer dipole (that the thread is responsible for) and source dipoles in all source boxes in the near field interaction list (NIL) of the observer box. To this end, threads collectively load $\mathbf{r}_n$ and $\hat{\mathbf{u}}_n$ of dipoles in each near field pair (stored in contiguous spaces in global memory) into their shared memory, calculate (2.15) on-the-fly, and update the pertinent entry of $\overline{\mathbf{F}}_i$. Note: The threads in one block are synchronized after computing the field produced by each source dipole to enable better coalesced global memory access patterns.

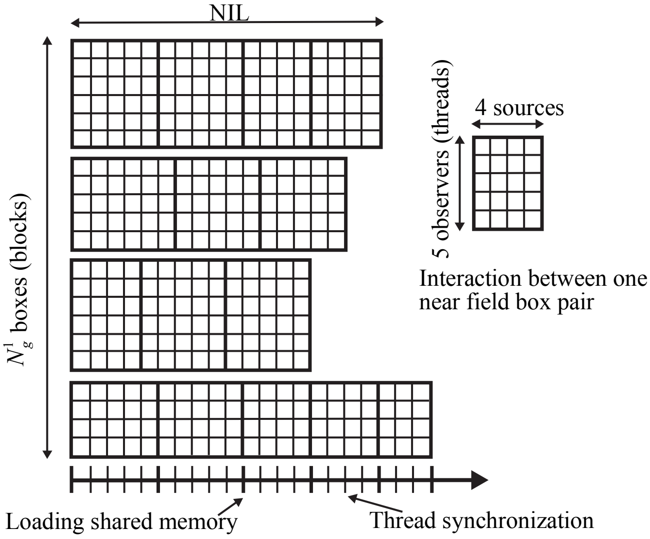


Figure 3.1 Single GPU implementation of the near field calculation.

3.2.2 Construction of Outgoing Rays

The outgoing rays of finest level boxes are computed by (2.17) for every M^1 time steps by launching one GPU kernel. This computation can be carried out on a GPU by parallelizing loops over temporal samples of a subsignal, dipoles of a box, directions, and boxes. Note that multiple accesses to the memory occupied by one outgoing ray is required in each iteration of the loops over temporal samples of subsignals and dipoles of a box. For this reason, the loops that are over directions and boxes are parallelized via a "one thread per direction" and "one block per box" strategy. In such strategy, the threads in each block collectively load $\mathbf{r}_n$ and $\hat{\mathbf{u}}_n$ of dipoles in one box into their shared memory, calculate the APS interpolants on-the-fly, use them to project $I_{n,j}$ onto one ray, and sum the projections of all dipoles (in the source box) to compute (2.19).

3.2.3 Translation of Outgoing Rays into Incoming Rays

Translation between far field box pair (α,α') at level v is performed by (2.18) for every ξM^v time steps via launching GPU kernels, where ξ is a constant that depends on $R_{c,\alpha\alpha'}$. This operation is executed on a GPU for each pair separately by the following steps [Figure 3.2]: Step 1: The l^{th} outgoing ray of box α, $\mathbf{G}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{pq},t)$, (depicted by a cuboid in Figure 3.2) is Fourier transformed to the frequency domain, $\hat{\mathbf{G}}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{pq},\omega)$. Step 2: The Fourier transform of the translator $\hat{\mathcal{T}}^{vec}(\hat{\mathbf{k}}^{v}_{pq},\omega,K^v)$ is directly computed in the frequency domain on approximately ξM^v samples (see [13] for analytical expressions) and then multiplied with the Fourier transform of the outgoing rays of box α, $\hat{\mathbf{G}}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{pq},\omega)$; both operations are parallelized via a "one thread per frequency sample" and "one block per direction" strategy thereby producing coalesced memory access. Step 3: The resulting data is inverse Fourier transformed into the time domain and the l^{th} incoming ray of box α', $\mathbf{G}^{-}_{l,\alpha'}(\hat{\mathbf{k}}^{v}_{pq},t)$, is updated. In steps 1 and 3, Fourier transforms are performed by the batched CUDA Fast Fourier Transform (CUFFT) library that allows simultaneous execution of $N^v_k = (K^v+1)(2K^v+1)$ FFTs; the transforms are accelerated by extending the sizes of frequency and temporal sequences to powers of two by zero padding.

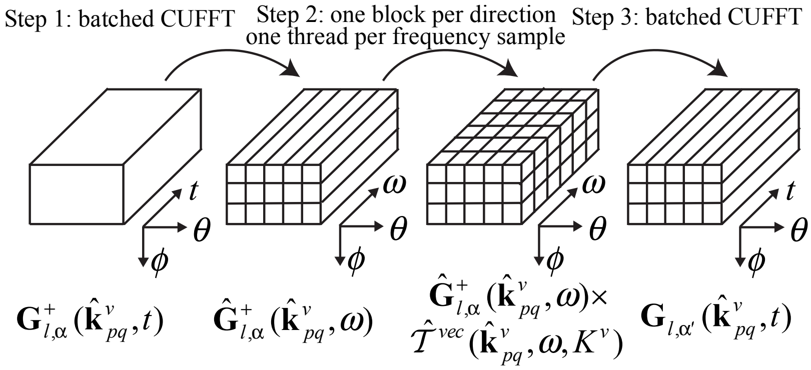

Step 1: batched CUFFT
Step 2: one block per direction
one thread per frequency sample
Step 3: batched CUFFT
t
θ
ϕ
ω
$\mathbf{G}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{pq},t)$
$\hat{\mathbf{G}}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{pq},\omega)$
$\hat{\mathbf{G}}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{pq},\omega)\times$
$\hat{\mathcal{T}}^{vec}(\hat{\mathbf{k}}^{v}_{pq},\omega,K^v)$
$\mathbf{G}_{l,\alpha'}(\hat{\mathbf{k}}^{v}_{pq},t)$

Figure 3.2 GPU implementation of one translation operation.

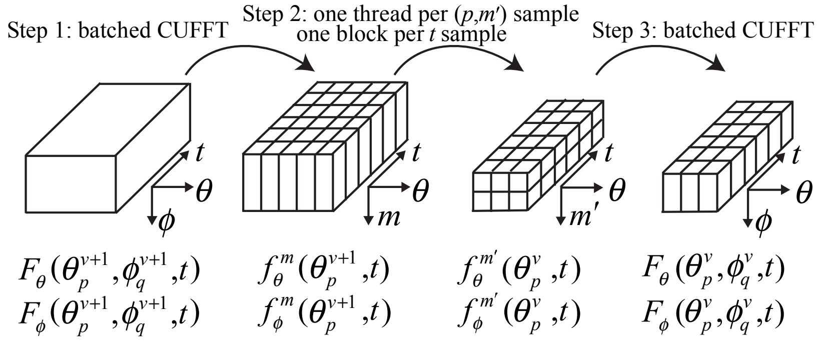
3.2.4 Processing and Projection of Incoming Rays

The incoming rays of finest level boxes are projected onto the dipoles by (2.19) for every time step by launching a GPU kernel. This projection can be performed on a GPU by parallelizing the loops that are over directions, dipoles of a box, and boxes. Since multiple accesses to the memory space of a dipole of a box is required at each iteration of the loop over directions, loops that are over dipoles of a box and loops over boxes are parallelized by a "one thread per dipole" and "one block per box" strategy. Each thread calculates the APS interpolants for all directions and uses them to update fields along the dipole.

3.2.5 Spherical Interpolation/Filtering

The outgoing/incoming rays of boxes at higher levels (i.e. $v>1$) are computed using the global vector spherical interpolation/filtering scheme of [13]. (Note: here only the GPU implementation of spherical filtering to obtain incoming rays is explained for the sake of brevity as that interpolate outgoing rays is very similar). The incoming rays of a box at level v consist of two transverse components, i.e., $\mathbf{G}^{-}_{l,\alpha'}(\hat{\mathbf{k}}^{v}_{pq},t)=F_{\theta}(\theta^{v}_{p},\phi^{v}_{q},t)\hat{\boldsymbol{\theta}}+F_{\phi}(\theta^{v}_{p},\phi^{v}_{q},t)\hat{\boldsymbol{\varphi}}$. Here, ϕ^{v}_{q} and θ^{v}_{p} are spherical coordinates defined as $\hat{\mathbf{k}}^{v}_{pq}=(\cos\phi^{v}_{q}\sin\theta^{v}_{p},\,\sin\phi^{v}_{q}\sin\theta^{v}_{p},\,\cos\theta^{v}_{p})$. These components are obtained by filtering the transverse components of the incoming rays of the parent box at level $v+1$, which are $F_{\theta}(\theta^{v+1}_{p},\phi^{v+1}_{q},t)$ and $F_{\phi}(\theta^{v+1}_{p},\phi^{v+1}_{q},t)$. This filtering operation is performed on a GPU by the following steps [Figure 3.3]: Step 1: The forward FFTs of $F_{\theta}(\theta^{v+1}_{p},\phi^{v+1}_{q},t)$ and $F_{\phi}(\theta^{v+1}_{p},\phi^{v+1}_{q},t)$ are computed along the ϕ-dimension. Step 2: The Fourier coefficients, $f^{m}_{\theta}(\theta^{v+1}_{p},t)$ and $f^{m}_{\phi}(\theta^{v+1}_{p},t)$, $m=1,\ldots,2K^{v+1}+1$, are truncated in the spectral domain via fast spectral truncation and correction [13] and the truncated Fourier coefficients, $f^{m'}_{\theta}(\theta^{v}_{p},t)$ and $f^{m'}_{\phi}(\theta^{v}_{p},t)$, $m'=1,\ldots,2K^{v}+1$, are obtained. The truncation and correction operations are parallelized via a "one thread per (p,m') sample" and "one block per t sample" strategy; each thread calculates $f^{m'}_{\theta}(\theta^{v}_{p},t)$ and $f^{m'}_{\phi}(\theta^{v}_{p},t)$ for one θ^{v}_{p} and one m'; again, this procedure yields coalesced memory access. Step 3: $F_{\theta}(\theta^{v}_{p},\phi^{v}_{q},t)$ and

$F_{\phi}(\theta_p^v,\phi_q^v,t)$, are obtained by inverse FFTing $f_{\theta}^{m'}(\theta_p^v,t)$ and $f_{\phi}^{m'}(\theta_p^v,t)$. Note that GPU kernels performing these steps are launched for each box separately. In steps 1 and 3, FFTs are performed by the batched CUFFT library, which allows simultaneous execution of $M^{v+1}(K^{v+1}+1)$ forward FFTs and $M^{v+1}(K^{v}+1)$ inverse FFTs.



Step 1: batched CUFFT
Step 2: one thread per (*p*,*m*′) sample
one block per *t* sample
Step 3: batched CUFFT
t
θ
ϕ
t
θ
m
t
θ
m'
t
θ
ϕ
$F_{\theta}(\theta_p^{v+1},\phi_q^{v+1},t)$
$F_{\phi}(\theta_p^{v+1},\phi_q^{v+1},t)$
$f_{\theta}^{m}(\theta_p^{v+1},t)$
$f_{\phi}^{m}(\theta_p^{v+1},t)$
$f_{\theta}^{m'}(\theta_p^{v},t)$
$f_{\phi}^{m'}(\theta_p^{v},t)$
$F_{\theta}(\theta_p^{v},\phi_q^{v},t)$
$F_{\phi}(\theta_p^{v},\phi_q^{v},t)$

Figure 3.3 GPU implementation of one spherical filtering operation.

3.3 Multi-GPU Acceleration

Although the above-described GPU implementation can achieve significant speedups compared to serial CPU implementations, the problem size that can be handled by single GPU implementation is still limited by the memory capacity of one GPU device. One possible remedy is to utilize the GPU-cluster as shown in Figure 3.4. A typical GPU-cluster consists of several compute-nodes connected with an InfiniBand network, each compute-node is equipped with one CPU (multi-core) and several GPUs that are connected with a PCI-E network. For simplicity, we assume that each GPU is associated with one CPU processor. In our implementation, the memory and computation loads of the PWTD algorithm is first distributed among N_p CPU processors using the provably scale CPU parallelization strategy described in Chapter 2. Then, the computational tasks of each processor are accelerated by one GPU.

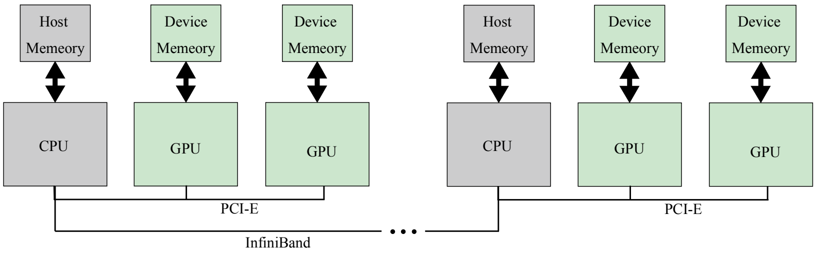


Figure 3.4 Architecture of a typical GPU-cluster

The multi-GPU implementation strategy greatly resembles the single GPU implementation except for the need of inter-node communication and reorganization of nonlocal computation tasks. Here the multi-GPU implementation for the PWTD stages is briefly summarized, with an emphasis on its difference compared with the single GPU implementation.

- *Near field calculation*. The GPU-cluster-based implementation still utilizes the “one block per observer box” and “one thread per observer” parallelization strategy [Figure 3.1]. However, the number of blocks associated with each CPU processor becomes approximately N_g^1 / N_p (recall that N_g^1 is the number of boxes at the finest level of the PWTD tree). Moreover, the NIL stored in the device memory becomes incomplete and hence the sources need to be communicated among processors before the GPU kernel is launched. Despite of these differences, good speedups can be achieved given that (i) N_g^1 is sufficiently large, and (ii) the communication cost is negligible in this stage.
- *Construction of outgoing rays from sources/projection of incoming rays onto observers*. In each CPU processor, these stages require no inter-node communication, therefore the parallelization strategy is still “one thread per outgoing ray, one block per source box” or “one block per observer box, one

thread per observer", except that the number of blocks becomes $N_g^1 \big/ N_p$ instead of N_g^1.

- *Translation*. Previously, the speedup using single GPU implementation of the translation stage is maximized via performing host-device memory transfers only at the *beginning* and *end* of the translation stage. However in GPU-cluster-based implementations, each arrived data packet (see Figure 2.3) needs to be transferred onto the GPU before the translation operation is performed. The overhead associated with this memory transfer may reduce the speedup of the translation stage when the number of CPU processors is large.
- *Spherical interpolation/filtering*. For case 1 and 2 (see Figure 2.2), the multi-GPU implementations of these stages remain unchanged except for need of redistribution of the filtered field samples via inter-node communication (for case 2). For case 3, however, the temporal dimension of rays is split by N_r^v CPU processors, the batch sizes of the GPU-based forward and backward FFTs become $M^{v+1}(K^{v+1}+1)/N_r^v$ and $M^{v+1}(K^v+1)/N_r^v$, and the number of blocks for the truncation and correction operations becomes approximately M^v / N_r^v.

3.4 Numerical Results

This section presents several numerical results that demonstrate the accuracy and efficiency of the proposed single and multiple GPU acceleration of the multilevel PWTD algorithm.

3.4.1 Fine-Tuning of the Parameters

First, it's worth noting that the speedups of GPU-PWTD strongly depend on the implementation parameters. This effect is investigated for two PWTD stages, viz., the near field calculation and the translation. The tests are performed on a Tesla C2050 device and an AMD Opteron 2220 SE. Figure 3.5 shows speedups for the near field calculation stage. Apparently, increasing the number of dipoles per box or the separation cutoff (equivalently, the size of NIL) results in higher speedups due to larger number of threads/blocks and more computational load for each thread. Similarly, Figure 3.6 shows speedups for the translation operation between one far field pair with varying K^{v} and translation length (proportional to $R_{c,\alpha\alpha'}$). Compared to the computation time of the translation operation performed using the Fastest Fourier Transform in the West (FFTW) library on a serial CPU, that of the GPU-accelerated implementation achieves significant speedup, especially when K^{v} is large (i.e., batch size is large). However, larger transform length will not always increase the speedup due to implementation differences between the CUFFT and FFTW libraries.

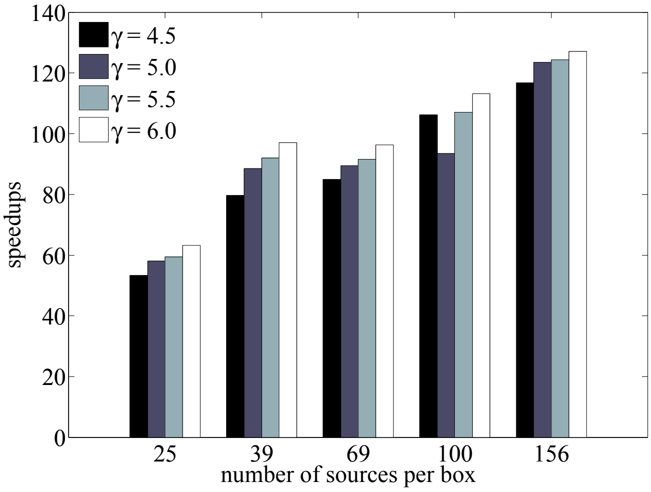


Figure 3.5 Speedups of GPU implementations over serial CPU implementations for the near field calculation stage

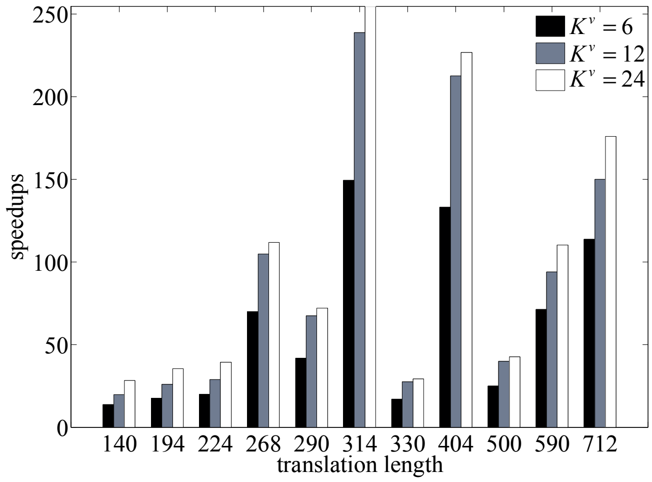


Figure 3.6 Speedups of GPU implementations over serial CPU implementations for the translation operation between a far field box pair.

3.4.2 Single GPU Acceleration

Next, the accuracy and efficiency of the single GPU-accelerated PWTD algorithm are demonstrated. All tests involve a set of dipoles that are randomly oriented and located on square plates with edge length ranging from 1.5 *m* $(N_s = 2500)$ to 9 *m* $(N_s = 40{,}000)$. The dipoles' temporal signature is $f_n(t) = m_n \cos(2\pi f_0 (t - 6\sigma)) \exp(-(t - 6\sigma)^2 / 2\sigma^2)$, where $\sigma = 4/(2\pi(f_{max} - f_0))$, $f_0 = 800\text{MHz}$, $f_{max} = 1\,\text{GHz}$, and m_n is a random real number between 0 and 1. The magnitudes of the time-derivatives of the electric fields along the dipoles, $|F_{i,m}|, m = 1,\ldots,N_s$, $i = 1,\ldots,N_t$, are computed for $N_t = 500$ time steps with $\Delta t = 6.25\times10^{-2}$ ns. In what follows, the GPU and CPU implementations of the direct scheme are termed GPU-Direct and CPU-Direct, respectively. GPU and serial CPU implementations (double precision) are executed on a Tesla C2050 device and an Intel Xeon E5-2670, respectively.

The field $|F_{i,m}|$ for an arbitrarily selected dipole, obtained by CPU-Direct, CPU-PWTD, and GPU-PWTD schemes are compared ($N_s = 10{,}000$) [Figure 3.7]. The L^2

norm error of the field computed by the GPU-PWTD algorithm is 1×10^{-4} (compared to the exact value obtained by the CPU-Direct scheme) while the relative difference between the field values obtained by the CPU-PWTD and GPU-PWTD schemes is around machine precision.

The computational time for each stage of CPU-PWTD and GPU-PWTD schemes is tabulated for $N_s = 2500$ and $N_s = 40{,}000$ [Table 3.1]. Note that the computational time for GPU-PWTD scheme includes the time of data transfer that is performed at the beginning and end of each stage. As N_s increases, the speedup achieved by GPU-PWTD scheme at each stage increases due to larger number of threads and blocks that are being leveraged.

The overall computational time and memory required by the CPU-Direct, GPU-Direct, CPU-PWTD, and GPU-PWTD schemes are compared for increasing N_s [Figure 3.8]. Here the parallelization strategy described in Section 3.2.1 was applied to the GPU-Direct scheme. The GPU-Direct scheme achieves 54.4X to 76.3X speedup while GPU-PWTD scheme achieves 30.4X to 53.3X speedup, and outperforms the other three schemes as N_s increases [Figure 3.8(a)]. As the GPU-Direct scheme computes matrix elements of $\bar{\bar{\mathbf{Z}}}_j^{vec}$ on-the-fly as opposed to the CPU-Direct scheme that pre-calculates them, it requires $O(N_s)$ global memory [Figure 3.8(b)]. (Note: Calculating $\bar{\bar{\mathbf{Z}}}_j^{vec}$ on-the-fly would result in dramatically higher computation time for the CPU-Direct scheme.) On the other hand, GPU-PWTD scheme achieves substantial memory reduction compared to CPU-PWTD scheme since it only stores ray data at one or two levels. The maximum number of sources for the GPU-PWTD scheme is $N_s = 40{,}000$ and limited by the 3 GB global memory. In addition, the performance of all schemes complies with the theoretical complexities.

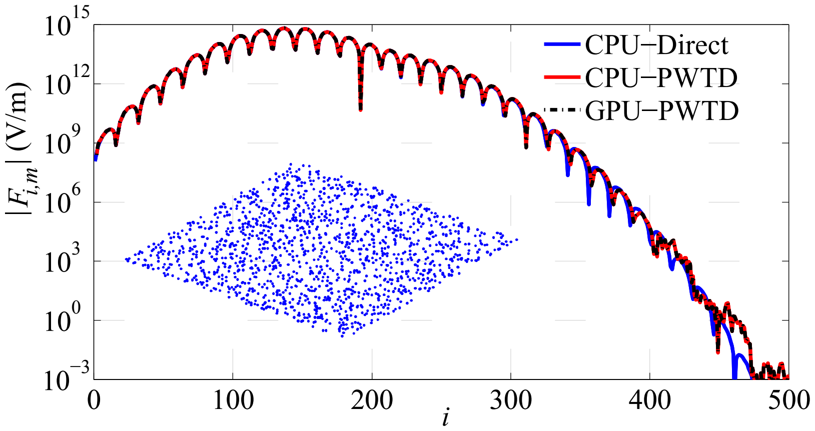


Figure 3.7 Comparison of $|F_{i,m}|$ obtained by CPU-direct, CPU-PWTD, and GPU-PWTD schemes.

	$N_s = 2500$			$N_s = 40{,}000$		
	CPU	GPU	Ratio	CPU	GPU	Ratio
Outgoing ray const.	11.5	2.1	5.5	201.6	30.8	6.7
Interpolation	2.3	0.2	11.5	332.1	23.1	14.3
Translation	194.6	6.8	28.6	17,883.1	380.4	47.1
Filtering	1.2	0.1	12.0	313.5	22.2	14.1
Incoming ray proj.	13.3	3.6	3.7	153.8	38.5	4.0
Near field calc.	783.4	20.2	38.8	20,275.9	239.7	84.8
Total	998.8	32.9	30.4	39,247.1	736.8	53.3

Table 3.1 Computation time for each stage of CPU-PWTD and GPU-PWTD schemes (in seconds) and the ratio between them

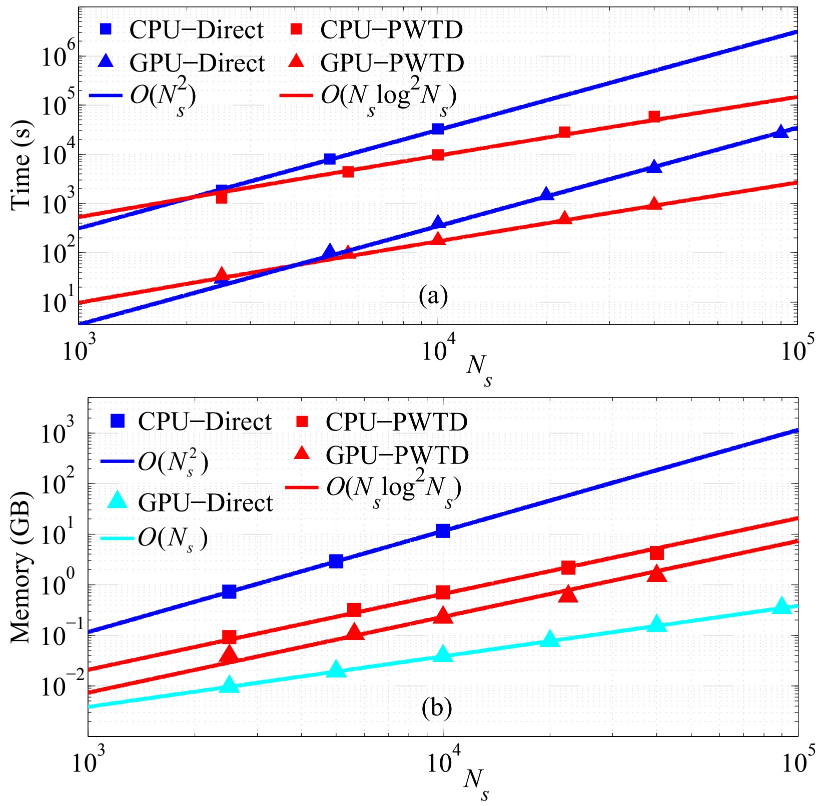


Figure 3.8 (a) Total computation time and (b) memory requirement of CPU-Direct, GPU-Direct, CPU-PWTD, and GPU-PWTD schemes.

3.4.3 Multi-GPU Acceleration

Finally, the efficiency of the multi-GPU-accelerated PWTD algorithm is demonstrated. All GPU-enabled simulations are carried out on two GPU compute-nodes connected with an InfiniBand network; each node consists of four Tesla C2050 and two Six-Core Intel Xeon X5650. For simplicity, each GPU device is assigned one MPI process (i.e., one CPU processor).

The speedups of the PWTD stages when the number N_p of GPUs changes from 1 to 8 [Figure 3.9]. Here, the number of dipoles is set to $N_s = 40{,}000$ and the number of time steps to $N_t = 500$. We first notice that for all PWTD stages, the eight GPU-based implementation does not achieve ideal speedups (i.e., 8X) compared to the single GPU implementation. This is because (i) although the computation is greatly accelerated by GPUs, the inter-node communication remain unchanged, which leads to reduced CCRs; (ii). the overhead associated with each host-device memory transfer reduces the speedups of the GPU implementations; this effect is significant especially for the translation stage that requires frequent host-to-device transfers for nonlocal data packets. Possible remedies to improve the speedups are (i) utilization of GPU-clusters with faster inter-node networks; (ii) CUDA-aware MPI solution that enables direct inter-node GPU-to-GPU memory transfers. That said, it's worth mentioning that more than 175X overall speedups have been achieved in this example [Figure 3.9].

Next, we illustrate the benefits of multiple GPUs to increase the solvable problem sizes. Figure 3.10(a) and (b) plot the computation time and memory requirement of single GPU and eight GPU-based implementations of the PWTD algorithm with $N_t = 500$ and different N_s. Apparently, the memory requirement is reduced by approximately eight times using eight GPUs compared with single GPU. As expected, the maximum problem size is increased from $N_s = 40{,}000$ to $N_s = 160{,}000$. However, as explained above, the speedups of the eight GPU-based implementation is not ideal.

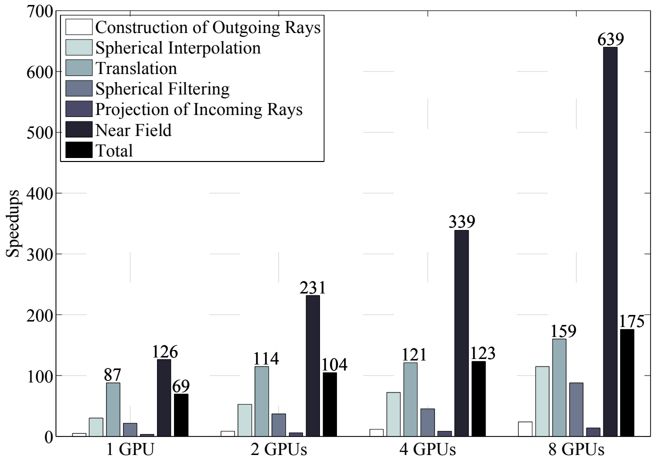


Figure 3.9 Speedups of multi-GPU-accelerated implementations the PWTD stages compared to serial CPU implementation.

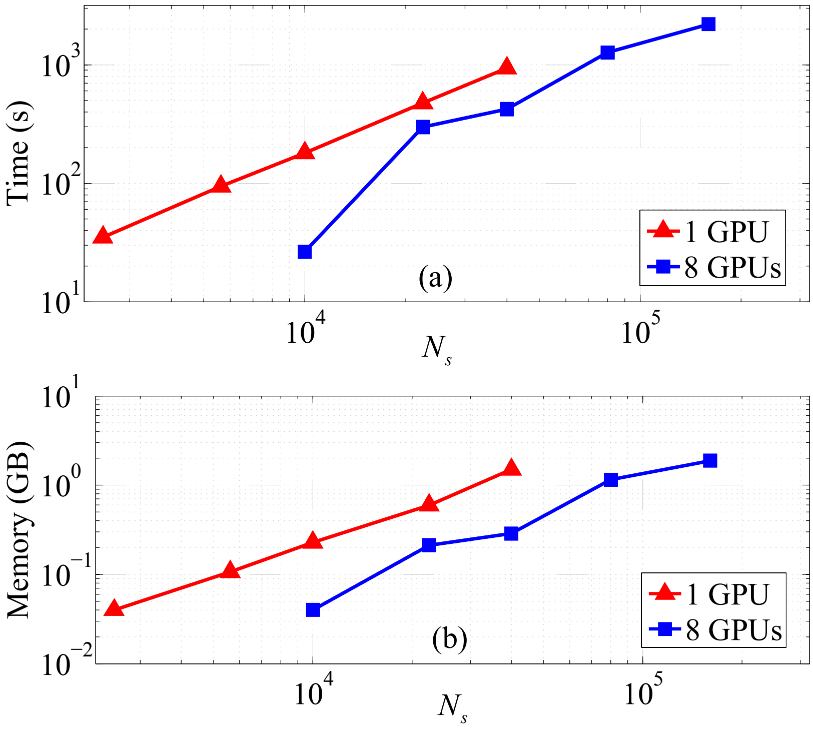


Figure 3.10 Scaling tests of the multilevel PWTD scheme implemented on the 1 GPU and 8 GPUs. (a) Computation time. (b) Memory requirement.

3.5 Chapter Conclusion

All-stage GPU-accelerated implementations of the multilevel PWTD schemes are presented. Multiple GPU-based implementations have been demonstrated. Substantial speedups and memory reductions have been achieved over serial CPU implementations,. The proposed single and multiple GPU acceleration of the multilevel PWTD algorithm can be used to enhance the classical MOT-TDIE solvers.

CHAPTER 4
A Parallel PWTD-Accelerated SIE Solver for Analyzing Transient Scattering from Electrically Large PEC Objects

4.1 Chapter Introduction

Transient electromagnetic scattering from electrically large objects involving PEC surfaces and piecewise homogeneous dielectric volumes can be analyzed using PWTD-accelerated MOT-TD-SIE solvers [13, 15]. These solvers can reduce the computational cost and memory requirement of unaccelerated MOT-TD-SIE solvers from $O(N_t N_s^2)$ and $O(N_s^2)$ to $O(N_t N_s \log^2 N_s)$ 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. Having said that, the applicability of PWTD-accelerated MOT-TD-SIE solver in analyzing transient scattering from electrically large objects can further be increased through effective parallelization schemes. In the past, problems involving half million spatial unknowns have been solved using parallel PWTD-accelerated MOT-TD-SIE solvers [115, 121]. However, these parallel solvers do not scale well on distributed memory clusters with large numbers of processors and are, therefore, incapable of solving real-life scattering problems that oftentimes involve millions of spatial unknowns.

In this chapter, the provable scalable parallel PWTD algorithm developed in Chapter 2 is incorporated into a MOT-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 hundred wavelengths and discretized with 10 million spatial unknowns.

4.2 Formulation

4.2.1 TD-SIEs and MOT Scheme

Consider the transient scattering problem described in Section 1.2.1.1. Here, the problem is formulated using the *differentiated* TD-EFIE/MFIE/CFIE as

$$-\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\dot{\mathbf{E}}^{i}(\mathbf{r},t)=\hat{\mathbf{n}}\times\eta_0\dot{\mathcal{L}}_e[\mathbf{J}](\mathbf{r},t)\quad\forall\mathbf{r}\in S,S^{+},S^{-} \tag{4.1}$$

$$\hat{\mathbf{n}}\times\dot{\mathbf{H}}^{i}(\mathbf{r},t)=\dot{\mathcal{L}}_h[\mathbf{J}](\mathbf{r},t)\quad\forall\mathbf{r}\in S^{-} \tag{4.2}$$

$$\hat{\mathbf{n}}\times\dot{\mathbf{H}}^{i}(\mathbf{r},t)-\beta/\eta_0\,\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\dot{\mathbf{E}}^{i}(\mathbf{r},t)=\dot{\mathcal{L}}_c[\mathbf{J}](\mathbf{r},t)\quad\forall\mathbf{r}\in S^{-}\,. \tag{4.3}$$

Here, $\dot{\mathcal{L}}_e$ and $\dot{\mathcal{L}}_h$ are differentiated TD-EFIE and TD-MFIE operators, and $\dot{\mathcal{L}}_c=\dot{\mathcal{L}}_h+\beta\hat{\mathbf{n}}\times\dot{\mathcal{L}}_e$ is the differentiated TD-CFIE operator [see Section 1.2.1.1]. Note: the TD-CFIE reduces to TD-EFIE and TD-MFIE when $\beta=\infty$ and $\beta=0$, respectively.

To numerically solve (4.3), the current density $\mathbf{J}(\mathbf{r},t)$ is discretized using N_s spatial basis functions and N_t temporal basis functions by (1.7). Substituting (1.7) into (4.3) and testing the resulting equation with $\mathbf{S}_m(\mathbf{r})$, $m=1,\ldots,N_s$, at $t=j\Delta t$ yields the set of linear equations

$$\bar{\bar{\mathbf{Z}}}_0\bar{\mathbf{I}}_j=\bar{\mathbf{F}}_j-\sum_{i=1}^{j-1}\bar{\bar{\mathbf{Z}}}_i\bar{\mathbf{I}}_{j-i}\,. \tag{4.4}$$

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

$$\{\bar{\mathbf{I}}_j\}_n=I_{n,j},\ n=1,\ldots,N_s \tag{4.5}$$

$$\begin{aligned}\{\bar{\mathbf{F}}_j\}_m=&\langle\mathbf{S}_m(\mathbf{r}),\hat{\mathbf{n}}\times\dot{\mathbf{H}}^{i}(\mathbf{r},t)\\&-\beta/\eta_0\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\dot{\mathbf{E}}^{i}(\mathbf{r},t)\rangle\Big|_{t=j\Delta t},\ m=1,\ldots,N_s\end{aligned} \tag{4.6}$$

$$\{\bar{\bar{\mathbf{Z}}}_i\}_{mn} = \left\langle \mathbf{S}_m(\mathbf{r}), \dot{\mathcal{L}}_c[\mathbf{S}_n T_{-i}](\mathbf{r},t) \right\rangle\Big|_{t=0} \quad m,n = 1,\ldots,N_s\,. \tag{4.7}$$

As mentioned in Section 1.2.5, the computationally most demanding operation is the evaluation of the sum on the RHS of (4.4). 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 [13] and is summarized next.

4.2.2 The 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 $v = 1,\ldots,N_L = O\left(\log(N_s^{0.5})\right)$. 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 $N_L - 2$, following similar procedures described in Section 2.2.1. 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 ($3 < \gamma < 6$) and their respective parent boxes do not constitute a far field pair. 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 (4.7) and their contributions are directly added to the RHS of (4.4). Interactions between spatial basis functions fully 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 $R_{c,\alpha\alpha'} = \left|\mathbf{R}_{c,\alpha\alpha'}\right| = \left|\mathbf{r}_o^c - \mathbf{r}_s^c\right|$ denote the distance between the source and observer box centers, $\mathbf{r}_s^c$ and $\mathbf{r}_o^c$ [Figure 4.1]. 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 $f_n(t)$ in (1.7) associated with $\mathbf{S}_n(\mathbf{r})$ is broken into N_l^v consecutive subsignals, $f_n^l(t)$, $l = 1,\ldots,N_l^v$, using APS interpolants [42] that is bandlimited to $\omega_s = \chi_t \omega_{max}$ and approximately time-limited to $-p_f \Delta t < t < p_f \Delta t$, $5 \le p_f \le 10$, as

$$f_n(t) = \sum_{l=1}^{N_l^v} f_n^l(t) = \sum_{l=1}^{N_l^v} \sum_{j=(l-1)M^v+1}^{lM^v} I_{n,j} T_j^{APS}(t) \tag{4.8}$$

where $T_j^{APS}(t) = T^{APS}(t - j\Delta t)$ and $N_l^v M^v = N_t$; M^v is chosen such that the duration of each subsignal, $T^v = (M^v + 2p_f)\Delta t$, is less than $(R_{c,\alpha\alpha'} - 2R^v)/c_0$. 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

$$\begin{aligned}
&\left\langle \mathbf{S}_m(\mathbf{r}), \dot{\mathcal{L}}_c[\mathbf{S}_n f_n^l](\mathbf{r},t) \right\rangle = \\
&\frac{1}{8\pi^2 c_0^2} \sum_{q=0}^{K^v} \sum_{p=-K^v}^{K^v} \omega_{qp}^v [-\beta \mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v) \\
&+ \mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{n}})]^\dagger * \mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t) * \\
&\mathbf{P}_n^+(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v) * f_n^l(t)
\end{aligned} \tag{4.9}$$

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, $\hat{\mathbf{k}}_{qp}^v$, $q = 0,\ldots,K^v$, $p = -K^v,\ldots,K^v$, represent directions of outgoing/incoming rays with a total of $N_k^v = (K^v + 1)(2K^v + 1)$ directions [8], † 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

$$\mathbf{P}_{\{m,n\}}^{\pm}(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{v}}) = \int_{S_{\{\mathrm{m,n}\}}} dS' \hat{\mathbf{v}} \times \mathbf{S}_{\{m,n\}}(\mathbf{r}') \delta(t \pm \hat{\mathbf{k}}_{qp}^v \cdot (\mathbf{r}' - \mathbf{r}_{\{o,s\}}^c) / c_0). \tag{4.10}$$

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

$$\mathcal{T}(\hat{\mathbf{k}}_{qp}^{v},t)=\frac{c_0\partial_t^3}{2R_{c,\alpha\alpha'}}\sum_{k=0}^{K^v}(2k+1)\Phi_k\left(\frac{c_0t}{R_{c,\alpha\alpha'}}\right)\Phi_k\left(\frac{\hat{\mathbf{k}}_{qp}^{v}\cdot\mathbf{R}_{c,\alpha\alpha'}}{R_{c,\alpha\alpha'}}\right) \tag{4.11}$$

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

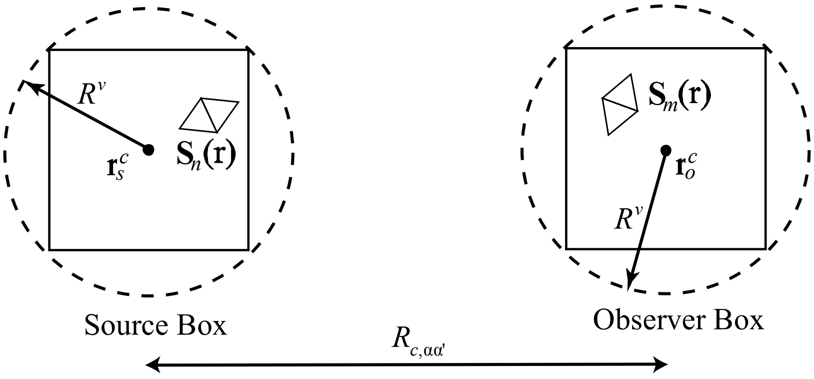


Figure 4.1 A far field box pair in the PWTD tree.

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} [8]. Note that only outgoing/incoming rays of the finest level boxes are constructed/projected directly from/onto basis functions using the projection function (4.10); those of higher level boxes are constructed/projected by an exact global vector spherical filtering technique described in [13]. The computational complexity analysis in [8] showed that the computational cost and memory requirements of a multilevel PWTD-accelerated MOT scheme applied to surface scatterers scale as $O(N_tN_s\log^2N_s)$ and $O(N_s^{1.5})$, respectively.

4.2.3 Parallelization of the PWTD-Accelerated TD-SIE Solver

The effective parallelization of the multilevel PWTD-accelerated TD-SIE solver calls for a uniform distribution of the near field MOT matrix elements (i.e. near field data) and the ray data and the pertinent workloads among processors. These loads are partitioned using the probably scalable parallelization strategy [developed in Chapter 2] that leverages hierarchical partitioning of the multilevel PWTD tree among processors and an asynchronous scheme for memory and cost efficient communications between processors. It's immediately clear from (2.22) that the memory load for each processor scales as $O(N_s^{1.5} / N_p)$ upon setting $d = 2$ and $\mu = 0$ in (2.22). In what follows, partitioning schemes for computation and communication loads in different stages of the PWTD-accelerated TD-SIE solver are briefly summarized, with an emphasis on their difference compared with that those developed in Chapter 2.

- *Construction/projection of outgoing/incoming rays.* At level $v = 1$, each processor constructs outgoing rays by convolving the projection function $\mathbf{P}_n^+(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v)$ in (4.10) with subsignals $f_n^l(t)$ associated with source spatial basis functions $\mathbf{S}_n(\mathbf{r})$ residing in its N_g^v / N_p source boxes; similarly, each processor projects the incoming rays using projection functions $\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v)$ and $\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{n}})$ onto test spatial basis functions $\mathbf{S}_m(\mathbf{r})$ in its N_g^v / N_p observer boxes. At level $v > 1$, the outgoing/incoming rays are constructed using the exact global vector spherical filtering technique; hence the parallelization strategy is the same as that used in parallelizing the PWTD algorithm for evaluating transient vector fields due to surface-bound sources [Chapter 2]. The computation load per processor for these two stages, CC_1 and CC_2, scale as $CC_1 \propto CC_2 = O(N_t N_s \log^2 N_s / N_p)$, and the communication load per processor for the stages, CC_1 and CC_2, scale as $CM_1 \propto CM_2 = O(N_t N_s \log N_s / N_p)$ [Note: these scaling estimates can be obtained via setting $\kappa = 2$ in (2.23) and (2.24)].

- *Translation.* The parallelization strategy is the same as that used in parallelizing the PWTD algorithm for evaluating transient vector fields due to surface-bound sources [Chapter 2]. As a result, the computation and communication loads per processor scale as $CC_3 = O(N_t N_s \log^2 N_s / N_p)$ and $CM_3 = O(N_t N_s \log N_s / N_p)$, respectively [Note: these scaling estimates can be obtained via setting $\kappa = 2$ in (2.26) and (2.27)].
- *Near field calculation.* Near field calculations include (i) matrix-vector multiplications on the left hand side (LHS) of (4.4) at each iteration of the solver executed for solving (4.4) at all time steps and (ii) partial matrix-vector multiplications on the RHS of (4.4). Here it is assumed that the solution of (4.4) 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 $CC_4 \propto CC_4 = O(N_t N_s / N_p)$.

By summing the computational and communication costs, CC_i and CM_i, $i = 1,2,3,4$, and noting that CC_4 is bounded by $CC_{\{1,2,3\}}$ and CM_4 is bounded by $CM_{\{1,2,3\}}$, the total computational cost and total communication cost of the PWTD-accelerated TD-SIE solver, CM and CC, scale

$$CC = O\left(\frac{N_t N_s \log^2 N_s}{N_p}\right),\ CM = O\left(\frac{N_t N_s \log N_s}{N_p}\right) \tag{4.12}$$

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.

4.3 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

$$\mathbf{E}^i(\mathbf{r},t) = \hat{\mathbf{p}} G(t - \mathbf{r}\cdot\hat{\mathbf{k}} / c_0) \tag{4.13}$$

where $G(t) = \cos[2\pi f_0 (t - t_0)]\exp[-(t - t_0)^2 / 2\sigma^2]$ 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 (4.4) at each time step. The GMRES iteration is terminated when the condition

$$\| \bar{\mathbf{V}}_j - \bar{\bar{\mathbf{Z}}}_0 \bar{\mathbf{I}}_j^{(n)} \| < \varepsilon \| \bar{\mathbf{V}}_j \| \tag{4.14}$$

is satisfied. Here, $\bar{\mathbf{I}}_j^{(n)}$ represents the vector of current coefficients in the n^{th} iteration, $\bar{\mathbf{V}}_j = \bar{\mathbf{F}}_j - \sum_{i=1}^{j-1} \bar{\bar{\mathbf{Z}}}_i \bar{\mathbf{I}}_{j-i}$ 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 is 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.

4.3.1 Canonical Examples

4.3.1.1 Sphere

The PWTD-accelerated TD-CFIE solver is applied to the analysis of scattering from a PEC sphere of radius 1 m centered about the origin. The sphere is illuminated by $\mathbf{E}^i(\mathbf{r},t)$ in (4.13) 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 $0.404\,\lambda$ and $\gamma = 4.5$. Table 4.1 presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires 7 TB of memory and 20 hours of CPU time when $N_p = 8192$. The solution at each time step is obtained in just one GMRES iteration, which yields $\varepsilon = 10^{-12}$.

The broadband RCS along the +*z* direction ($\theta = 0$) obtained using the proposed solver is compared with the exact Mie series solution in Figure 4.2; excellent agreement is observed. Snapshots of the current density on the sphere at times $380\ \Delta t$, $860\ \Delta t$, and $1400\ \Delta t$ show a surface wave fading away in the forward scattering direction [Figure 4.3]. Current densities induced at $(\theta = 180^{\circ}, \phi = 0)$ and $(\theta = 0, \phi = 0)$ computed using the proposed solver are in good agreement with the Mie series solution [Figure 4.4].

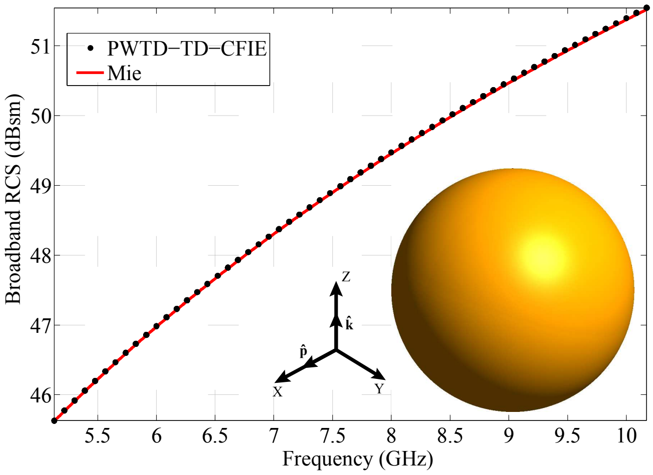
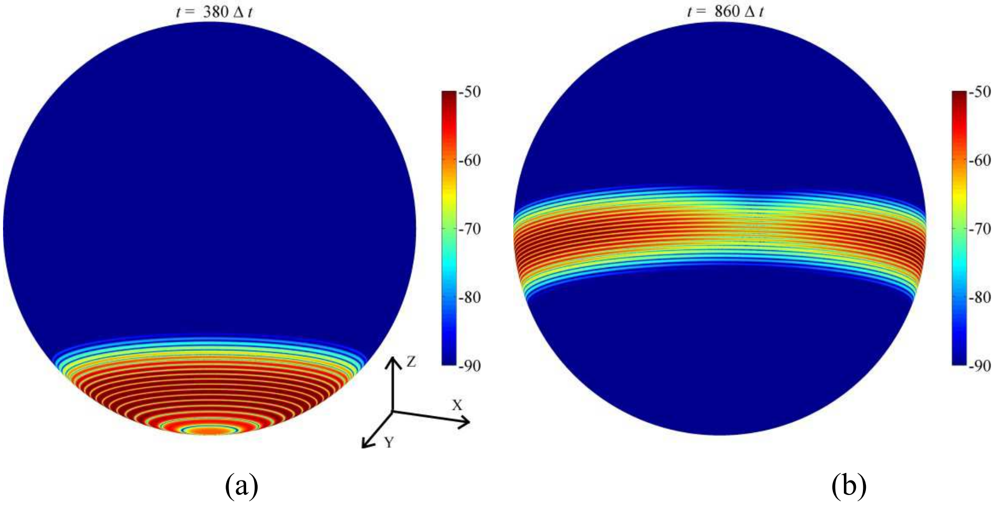
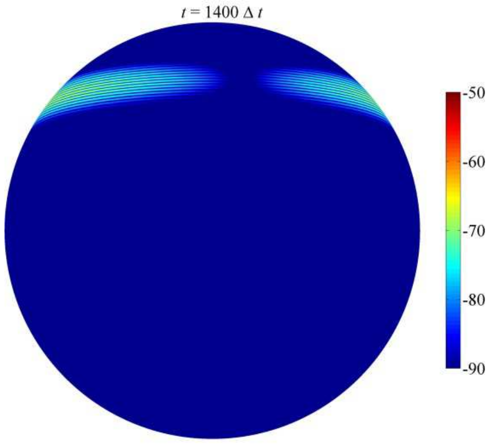


Figure 4.2 Broadband RCS of the sphere along the +*z* direction computed by the PWTD-accelerated TD-CFIE solver and Mie series solution.



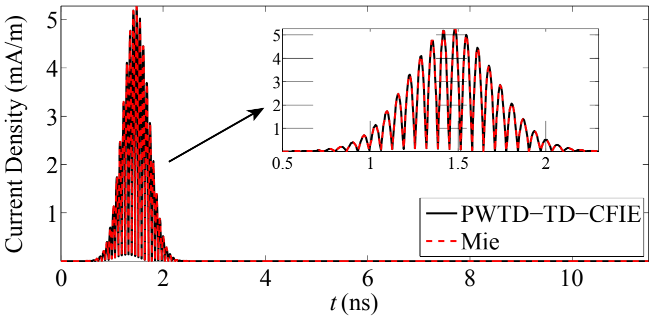
(a)

(b)

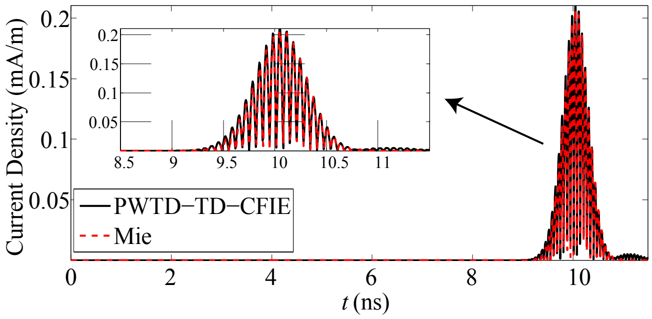


(c)

Figure 4.3 Snapshots of the current density (in dB) induced on the sphere computed by PWTD-accelerated TD-CFIE solver at (a) $t = 380\ \Delta t$, (b) $t = 860\ \Delta t$, (c) $t = 1400\ \Delta t$.



(a)



(b)

Figure 4.4 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).

	Sphere	Plate
Maximum dimension	2 m (70 λ)	120 m (160 λ)
Frequency band $[f_{\min}, f_{\max}]$	[5.12,10.24] GHz	[200, 400] MHz
Number of unknowns N_s	9,433,437	2,920,476
Time step size Δt	5 ps	125 ps
Number of processors N_t	2,300	1,000
Number of processors N_p	8,192	2,048
Memory for $\overline{\overline{\mathbf{Z}}}_l$/PWTD	3.28/3.78 TB	0.46/3.02 TB
Setup time	3 h	1.63 h
RHS time (near field)	1.38 h	628 s
RHS time (PWTD)	11.2 h	4 h
LHS time (GMRES)	3.8 h	108 s
Number of GMRES iterations	1	1
RCS calculation time	777 s	300 s

Table 4.1 Technical data for the setups and solutions of scattering problems involving canonical examples.

4.3.1.2 Plate

The PWTD-accelerated TD-EFIE solver is applied to the analysis of scattering from a PEC square plate with side length of $120\,\mathrm{m}$. The plate is centered at the origin and positioned parallel to the *xy* plane. The plate is illuminated by $\mathbf{E}^i(\mathbf{r},t)$ in (4.13) with $f_0 = 300\,\mathrm{MHz}$, $f_{bw} = 100\,\mathrm{MHz}$, $\hat{\mathbf{p}} = -\hat{\mathbf{y}}$, and $\hat{\mathbf{k}} = -\sin(5°)\hat{\mathbf{x}} - \cos(5°)\hat{\mathbf{z}}$. 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 $\Delta t = 125\,\mathrm{ps}$. A ten-level PWTD tree is constructed upon setting the side length of boxes at the finest level to $0.64\,\lambda$ and $\gamma = 3.5$. Table 4.1 presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires 3.5 TB of memory and seven hours 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 specular reflection) in Figure 4.5; good agreement is observed. Snapshots of the current density at times 440 Δt, 560 Δt, and 720 Δt reveal physical optics and edge diffracted currents [Figure 4.6].

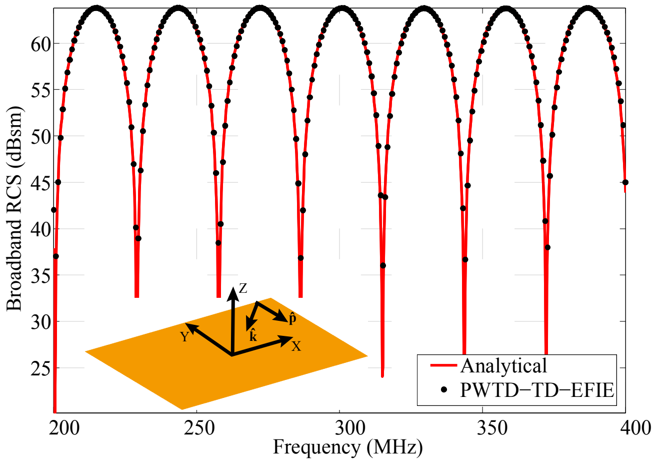
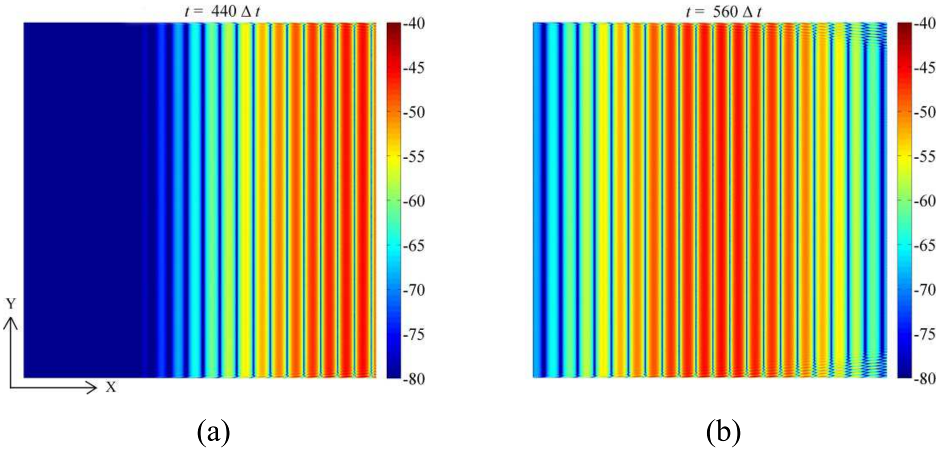
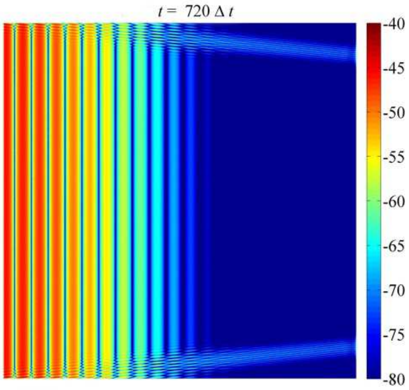


Figure 4.5 The broadband RCS of the PEC plate along +z direction computed by the PWTD-accelerated TD-EFIE solver and analytical formula.



(a)

(b)



(c)

Figure 4.6 Snapshots of the current density (in dB) induced on the plate obtained by the PWTD-accelerated TD-EFIE solver at (a) $t = 440\ \Delta t$, (b) $t = 560\ \Delta t$, (c) $t = 720\ \Delta t$.

4.3.2 Real-World Objects

4.3.2.1 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 (4.13) 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 $\Delta t = 31.25\,\text{ps}$. A ten-level PWTD tree is constructed upon setting the side length of boxes at the finest level to $0.48\,\lambda$ and $\gamma = 4$. Table 4.2 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 hours 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 bistatic RCS of the helicopter at 1.2 GHz is computed using the proposed solver and a frequency domain combined field integral equation (FD-CFIE) solver [140, 141] [Figure 4.7]; the results agree very well. In addition, the broadband RCS along the $+z$ direction ($\theta = 0$) is computed using both solvers [Figure 4.8]; again results are in good agreement. Moreover, the current densities induced at two points selected on the blade (labeled "front" in Figure 4.9) and the tail are computed [Figure 4.9]. Note the increase in the current density on the blade due to reflections from the cockpit. Finally, snapshots of the current induced on the helicopter at times $950\,\Delta t$, $1200\,\Delta t$, and $2300\,\Delta t$ are shown in Figure 4.10. The amplitudes of currents on edges and regions where multiple reflections exist are larger than those elsewhere.

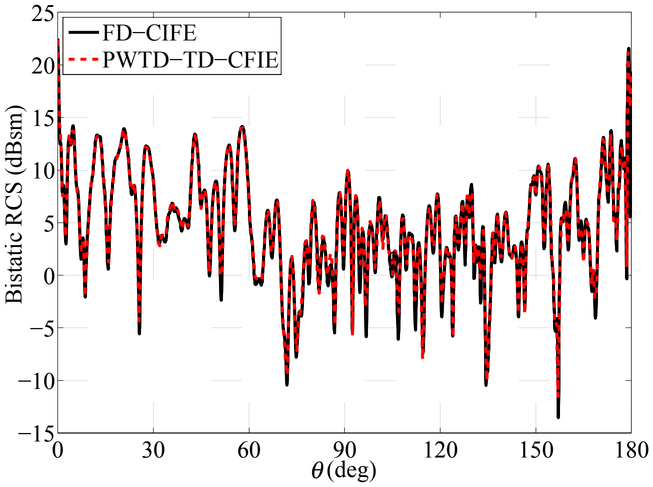


Figure 4.7 Bistatic RCS of the Rooivalk helicopter at 1.2 GHz computed at $\phi = 0$ and $\theta = [0,180^\circ]$ via the PWTD-accelerated TD-CFIE solver and FD-CIFE solver.

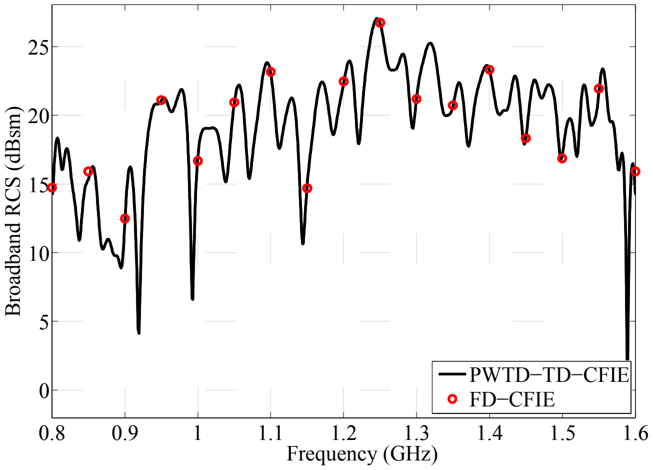


Figure 4.8 Broadband RCS of the Rooivalk helicopter along the +*z* direction computed by PWTD-accelerated TD-CFIE solver and FD-CFIE solver.

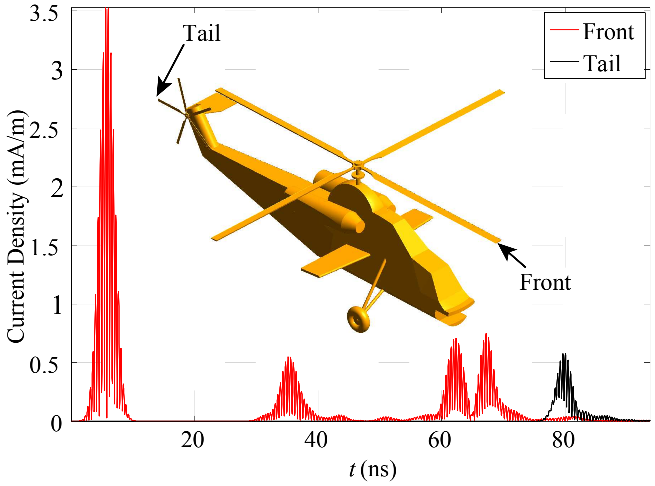
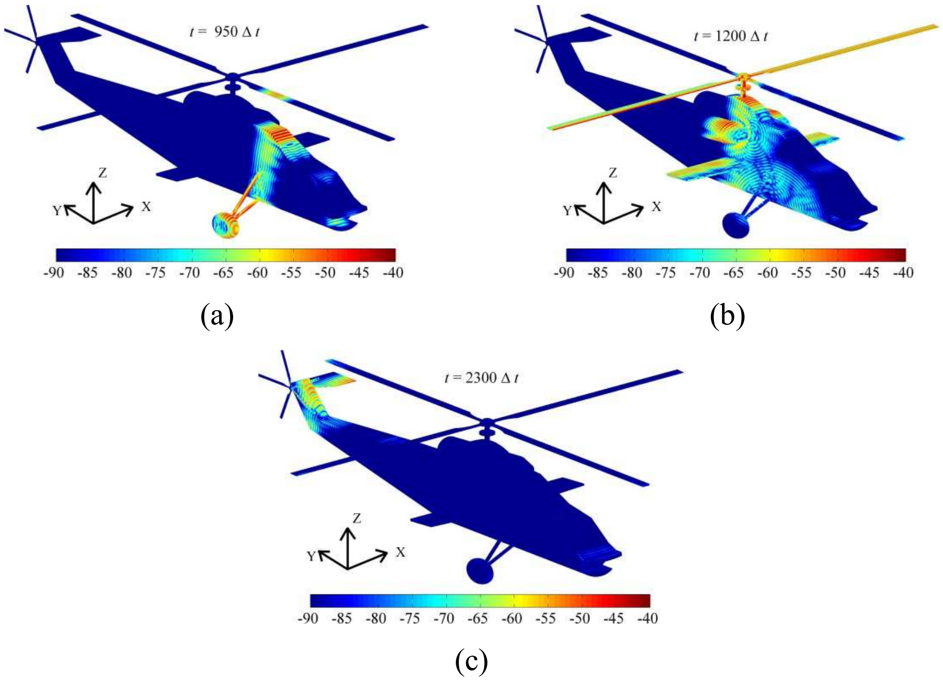


Figure 4.9 Magnitudes of the current density at the front and tail of the helicopter computed by the PWTD-accelerated TD-CFIE solver.



(a)

(b)

(c)

Figure 4.10 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) $t = 1200\ \Delta t$, (c) $t = 2300\ \Delta t$.

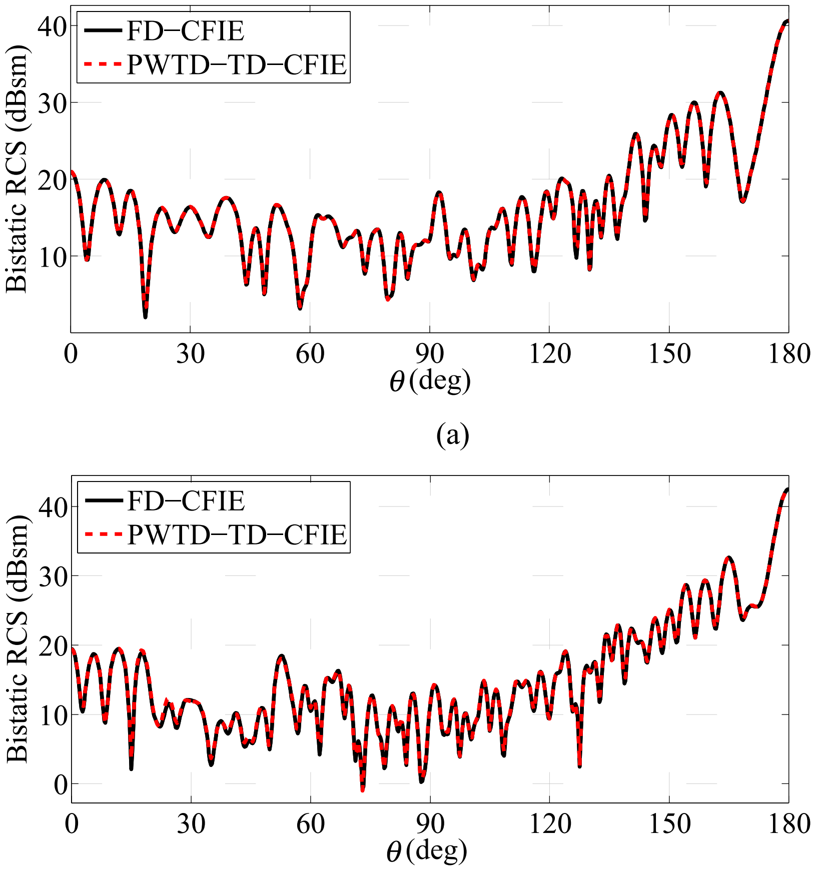
	Helicopter	Airbus A-320
Maximum dimension	22 m (117.3 λ)	37.5 m (33 λ)
Frequency band $[f_{\min}, f_{\max}]$	$[0.8, 1.6]$ GHz	$[133.5, 266.5]$ MHz
Number of unknowns N_s	2,436,813	1,020,069
Time step size Δt	31.25 ps	187 ps
Number of time steps N_t	3,000	1,600
Number of processors N_p	8,192	2,048
Memory for $\overline{\overline{\mathbf{Z}}}_i$/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

Table 4.2 The technical data for the setups and solutions of scattering problems involving real-life targets

4.3.2.2 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 (4.13) with $f_0 = 200\text{ MHz}$, $f_{bw} = 67.5\text{ MHz}$, $\hat{\mathbf{p}} = \hat{\mathbf{y}}$, 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 $\Delta t = 187\text{ ps}$. An eight-level PWTD tree is constructed upon setting the side length of boxes at the finest level to $0.38\ \lambda$ and $\gamma = 4.5$. Table 4.2 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 hours of CPU time when $N_p = 2048$. The solution at each time step is obtained in maximum three GMRES iterations, which yields $\varepsilon = 5\times 10^{-15}$.

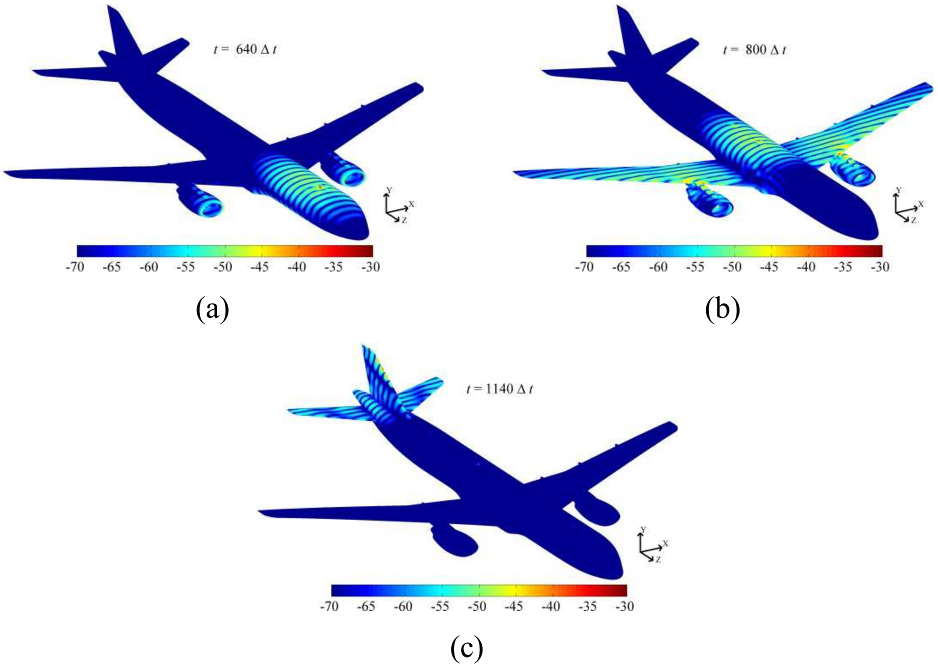
The bistatic RCS of the airplane is computed at 200 MHz and 250 MHz and compared to those obtained using a FD-CFIE solver [Figure 4.11]; results are in good agreement. In addition, snapshots of the current induced on the airplane at times 640 Δt, 800 Δt, and 1140 Δt are shown in Figure 4.12. The maximum current densities are induced on edges of antennas placed on top of the airplane and the engine intake.



(a)

(b)

Figure 4.11 Bistatic RCS of the Airbus-A320 airplane at (a) 200 MHz and (b) 250 MHz computed at $\phi = 0$ and $\theta = [0,180^\circ]$ via PWTD-accelerated TD-CFIE solver and FD-CFIE solver.



(a)

(b)

(c)

Figure 4.12 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) $t = 800\ \Delta t$, (c) $t = 1140\ \Delta t$.

4.4 Chapter 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. Indeed, numerical results validate the scalability and parallel efficiency of the proposed scheme up to 4096 processors. The resulting PWTD accelerated TD-SIE solver is successfully applied to analysis of transient scattering from objects measuring well over one hundred wavelengths in size and discretized using 10 million spatial unknowns.

CHAPTER 5
A Wavelet-Enhanced PWTD-Accelerated SIE Solver for Analysis of Transient Scattering from Electrically Large PEC Objects

5.1 Chapter Introduction

PWTD-accelerated MOT-TD-SIE solvers constitute a viable alternative to finite difference time domain solvers for analyzing complex transient scattering phenomena involving PEC [13] and homogenous dielectric [15] objects. These solvers are the time domain counterparts of MLFMA-accelerated FD-SIE solvers [142]. Their computational costs and memory requirements 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.

These attractive scaling laws aside, the applicability of the PWTD-TD-SIE solver to analyzing transient scattering from objects spanning many wavelengths was impeded for years due to the lack of an efficient parallelization scheme, for reasons relating to the heterogeneous structure of the PWTD algorithm. In Chapter 4, we present such a parallel PWTD-TD-SIE solver capable of solving electrically very large transient scattering problems involving 10 million spatial unknowns using thousands of CPUs. That being said, the overall memory cost of the PWTD-TD-SIE solver is oftentimes, to the our knowledge, one order of magnitude higher than that of the MLFMA-FD-SIE solver mainly due to the extra temporal dimension of the PWTD ray data.

In this chapter, a wavelet compression scheme is developed to alleviate the abovementioned computation bottleneck of PWTD-TD-SIE solver. In the past, wavelets (and wavelet packets) have been extensively studied and applied to solution of FD-SIEs for both low frequency [143] and high frequency [144-147] time-harmonic scattering problems, as well as the solution of TD-SIEs for transient scattering problems [148, 149]. These works employ mainly the wavelet representation of the surface current for the virtue of impedance matrix sparsification. In our work, in contrast, the wavelet is utilized to exploit the temporal sparsity of the PWTD ray data. The motivation of this work is as follows: in transient scattering problems, the object is oftentimes illuminated by plane waves with high frequency temporal pulses. These incident fields induce on the object a current density with temporal signatures resembling these incident pulses. As this spatial-temporal current is locally correlated, the PWTD ray data constructed from the current retains, in some extent, the high frequency pulse nature, and can be efficiently represented using wavelet-type bases. Specifically, we develop a wavelet-domain implementation of the PWTD algorithm by representing the PWTD ray data using local cosine wavelet bases (LCBs) and performing most of the PWTD operations in the wavelet-domain. The LCBs [150], compared to other types of wavelets, are known to be well suited for representing high frequency signals and integral kernels [147, 151], hence are chosen as the wavelet bases in this work. The resulting LCB-enhanced PWTD algorithm can significantly reduce the memory *and* computational costs of the conventional PWTD algorithm. Furthermore, the LCB-enhanced PWTD algorithm is incorporated into a TD-SIE solver to enable efficient and accurate analysis of transient scattering from electrically large PEC objects. The capability of this solver is demonstrated through its application to canonical and real-life electromagnetics problems involving objects spanning well over a hundred wavelengths.

5.2 MOT-based Solution of TD-SIEs and its PWTD Acceleration

5.2.1 TD-SIEs and MOT Scheme

Consider the transient scattering from a closed PEC surface S residing in free space, as described in Section 1.2.1.1 (and Section 4.2.1). Here, the TD-SIE formulation and its MOT-based solution are briefly summarized, more details can be found in Section 1.2.1.1 and Section 4.2.1. The differentiated TD-CFIE is

$$\hat{\mathbf{n}}\times\dot{\mathbf{H}}^i(\mathbf{r},t)-\beta/\eta_0\,\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\dot{\mathbf{E}}^i(\mathbf{r},t)=\dot{\mathcal{L}}_c[\mathbf{J}](\mathbf{r},t)\quad\forall\mathbf{r}\in S^-. \tag{5.1}$$

Here, $\dot{\mathbf{E}}^i(\mathbf{r},t)$ and $\dot{\mathbf{H}}^i(\mathbf{r},t)$ are the differentiated total electric and magnetic fields, $\mathbf{J}(\mathbf{r},t)$ is the current density induced on the surface, $\dot{\mathcal{L}}_c$ is the differentiated TD-CFIE operator, β is the CFIE combination constant, η_0 is the characteristic impedance of the free space, and S^- denotes the surface conformal to but just inside S.

Upon expansion of the current density $\mathbf{J}(\mathbf{r},t)$ by (1.7), substituting (1.7) into (5.1) and testing the resulting equation with $\mathbf{S}_m(\mathbf{r})$, $m=1,\ldots,N_s$, at $t=j\Delta t$, $j=1,\ldots,N_t$, a set of linear equations can be obtained as

$$\overline{\overline{\mathbf{Z}}}_0\overline{\mathbf{I}}_j=\overline{\mathbf{F}}_j-\sum_{i=1}^{j-1}\overline{\overline{\mathbf{Z}}}_i\overline{\mathbf{I}}_{j-i}. \tag{5.2}$$

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

$$\{\overline{\mathbf{I}}_j\}_n=I_{n,j},\ n=1,\ldots,N_s \tag{5.3}$$

$$\begin{aligned}\{\overline{\mathbf{F}}_j\}_m=&\langle\mathbf{S}_m(\mathbf{r}),\hat{\mathbf{n}}\times\dot{\mathbf{H}}^i(\mathbf{r},t)\\&-\beta/\eta_0\hat{\mathbf{n}}\times\hat{\mathbf{n}}\times\dot{\mathbf{E}}^i(\mathbf{r},t)\rangle\Big|_{t=j\Delta t},\ m=1,\ldots,N_s\end{aligned} \tag{5.4}$$

$$\{\overline{\overline{\mathbf{Z}}}_i\}_{mn}=\left\langle\mathbf{S}_m(\mathbf{r}),\dot{\mathcal{L}}_c[\mathbf{S}_nT_{-i}](\mathbf{r},t)\right\rangle\Big|_{t=0}\quad m,n=1,\ldots,N_s. \tag{5.5}$$

This MOT-based solution requires $O(N_tN_s^2)$ operations and $O(N_s^2)$ memory.

5.2.2 The Multilevel PWTD Algorithm

First, the PWTD tree can be constructed and far/near field box pairs can be identified following the same procedure described in Section 4.2.2. Next, interactions between spatial basis functions residing in near field box pairs are computed using (5.5); interactions between spatial basis functions fully contained in far field box pairs are evaluated by the PWTD scheme.

Consider interactions between all basis functions in the far field box pair α and α' shown in Figure 1.1. For all $n \in \alpha$, the temporal signature $f_n(t)$ associated with $\mathbf{S}_n(\mathbf{r})$ is broken into N_l^v consecutive subsignals $f_n^l(t)$ using the APS function $T^{APS}(t)$ via (4.8). Let $\mathbf{J}_\alpha^l(\mathbf{r},t) = \sum_{n\in\alpha} \mathbf{S}_n(\mathbf{r}) f_n^l(t)$ denote the current due to the l^{th} subsignal associated with all source basis functions in box α, fields produced by $\mathbf{J}_\alpha^l(\mathbf{r},t)$ (and tested by $\mathbf{S}_m(\mathbf{r})$ in box α') can be computed as follows: (i) construct a set of outgoing rays (of box α) in direction $\hat{\mathbf{k}}_{qp}^v$ by the convolving the projection function $\mathbf{P}_n^+(\hat{\mathbf{k}}_{pq}^v, t, \hat{\mathbf{k}}_{pq}^v)$ with subsignal $f_n^l(t)$ as

$$\mathbf{G}_{l,\alpha}^+(\hat{\mathbf{k}}_{qp}^v, t) = \frac{\partial_t^2}{16\pi^2 c_0^2} \sum_{n\in\alpha} \mathbf{P}_n^+(\hat{\mathbf{k}}_{qp}^v, t, \hat{\mathbf{k}}_{qp}^v) * f_n^l(t) \tag{5.6}$$

where $t \in [t_l^{+s}, t_l^{+e}]$ with starting point $t_l^{+s} = \left((l-1)M^v - p_f\right)\Delta t - R^v / c_0$ and ending point $t_l^{+e} = (lM^v + p_f)\Delta t + R^v / c_0$; (ii) translate the outgoing rays (of box α) into incoming rays (of box α') by convolving outgoing rays $\mathbf{G}_{l,\alpha}^+(\hat{\mathbf{k}}_{qp}^v, t)$ with the translation function $\mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t)$ as

$$\mathbf{G}_{l,\alpha'}^-(\hat{\mathbf{k}}_{qp}^v, t) = \mathcal{T}(\hat{\mathbf{k}}_{qp}^v, t) * \mathbf{G}_{l,\alpha}^+(\hat{\mathbf{k}}_{qp}^v, t)\ ; \tag{5.7}$$

here, $t \in [t_l^{-s}, t_l^{-e}]$ with the starting point $t_l^{-s} = \left((l-1)M^v - p_f\right)\Delta t - (R^v + R_{c,\alpha\alpha'}) / c_0$ and the ending point $t_l^{-e} = (lM^v + p_f)\Delta t + (R^v + R_{c,\alpha\alpha'}) / c_0$; (iii) project the incoming rays onto the test basis function $\mathbf{S}_m(\mathbf{r})$ in box α' by convolving the projection functions

$\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v,t,\hat{\mathbf{k}}_{qp}^v)$ and $\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v,t,\hat{\mathbf{n}})$ with the incoming rays $\mathbf{G}_{l,\alpha'}^-(\hat{\mathbf{k}}_{pq}^v,t)$ and summing over all directions with quadrature weights ω_{qp}^v as

$$\begin{aligned}\left\langle \mathbf{S}_m(\mathbf{r}),\dot{\mathcal{L}}_c[\mathbf{J}_\alpha^l](\mathbf{r},t)\right\rangle = \sum_{q=0}^{K^v}\sum_{p=-K^v}^{K^v}\omega_{qp}^v \\ [-\beta\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v,t,\hat{\mathbf{k}}_{qp}^v)+\mathbf{P}_m^-(\hat{\mathbf{k}}_{qp}^v,t,\hat{\mathbf{n}})]^\dagger * \mathbf{G}_{l,\alpha'}^-(\hat{\mathbf{k}}_{qp}^v,t)\ ,\end{aligned} \tag{5.8}$$

where the number of ray directions is $N_k^v=(K^v+1)(2K^v+1)$, $K^v=\left\lfloor 2\chi_s\omega_s R^v/c_0\right\rfloor+1$ with a spherical oversampling factor χ_s. In (5.6)-(5.8), the projection function is

$$\mathbf{P}_{\{m,n\}}^{\pm}(\hat{\mathbf{k}}_{qp}^v,t,\hat{\mathbf{v}})=\int_{S_{\{\mathrm{m,n}\}}} dS'\hat{\mathbf{v}}\times\mathbf{S}_{\{m,n\}}(\mathbf{r}')\delta(t\pm\hat{\mathbf{k}}_{qp}^v\cdot(\mathbf{r}'-\mathbf{r}_{\{o,s\}}^c)/c_0)\,, \tag{5.9}$$

here $S_{\{m,n\}}$ represents the support of $\mathbf{S}_{\{m,n\}}(\mathbf{r})$ and $\delta(\cdot)$ is the Dirac delta function. The translation function is

$$\mathcal{T}(\hat{\mathbf{k}}_{qp}^v,t)=\frac{c_0\partial_t}{R_{c,\alpha\alpha'}}\sum_{k=0}^{K^v}(2k+1)\Phi_k\left(\frac{c_0 t}{R_{c,\alpha\alpha'}}\right)\Phi_k\left(\cos\theta\right) \tag{5.10}$$

where $\Phi_k(\cdot)$ is the Legendre polynomial of degree k, $\cos\theta=\hat{\mathbf{k}}_{qp}^v\cdot\mathbf{R}_{c,\alpha\alpha'}/R_{c,\alpha\alpha'}$ and $|t|\le R_{c,\alpha\alpha'}/c_0$. Note that two out of the three differentiation operators ∂_t in (4.11) are transferred into (5.6) here. In practice, only outgoing/incoming rays of finest level boxes are constructed/projected directly from/onto spatial basis functions using (5.6)/(5.8); those of higher level boxes are computed via the global vector spherical interpolation/filtering [13, 132]. The computational complexity analyses in [8] showed that the computational cost and memory requirement of the multilevel PWTD-accelerated MOT scheme scale as $O(N_t N_s \log^2 N_s)$ and $O(N_s^{1.5})$ for surface scatterers, respectively. Note: in the rest of this paper, the computational cost of the PWTD-MOT scheme is rewritten as $O(N_s^{1.5}\log^2 N_s)$, as $N_t=O(N_s^{0.5})$ is a condition typically satisfied for smooth objects under high frequency excitations.

5.3 LCB-Enhanced Multilevel PWTD Algorithm

This section describes a wavelet-based compression scheme that further reduces the memory and computational costs associated with the PWTD algorithm by leveraging the local cosine wavelet compression among the temporal dimension of the ray data. In what follows, the memory compression scheme for the PWTD ray data using LCBs is first described in Section 5.3.1. Next, the wavelet-domain implementations of the translation and spherical interpolation/filtering stages, are expounded in Section 5.3.2 and Section 5.3.3. Finally, the memory requirement and computational complexity of the LCB-enhanced PWTD algorithm are analyzed in 5.3.4.

5.3.1 Representation of the Ray Data Using LCBs

In order to represent the PWTD ray data using LCBs, the temporal support of the outgoing/incoming ray $[t_l^{\pm s}, t_l^{\pm e}]$ is first partitioned into N intervals by an increasing sequence (a_r), $r = 1,...,N$ with $a_1 = t_l^s - \varepsilon$, $a_N = t_l^e + \varepsilon$ and $0 < \varepsilon \le (a_{r+1} - a_r)/2$ for $\forall r < N$ [Note: the symbol ± in the superscripts of $t_l^{\pm s}$ and $t_l^{\pm e}$ is dropped]. It's assumed that the length of the r^{th} interval is integer multiple of Δt, i.e., $a_{r+1} - a_r = M^r \Delta t$. On the r^{th} interval, the LCB functions $T_{ru}(t)$, $u \in \mathbb{N}$ are

$$T_{ru}(t) = B_r(t)\sqrt{\frac{2}{M^r \Delta t}}\cos\left((2u-1)\frac{t-a_r}{2M^r\Delta t}\pi\right) \tag{5.11}$$

where $B_r(t)$ is a smooth, compactly supported bell function defined as

$$B_r(t) = \begin{cases} b\left(\dfrac{t-a_r}{\varepsilon}\right) & a_r - \varepsilon \le t < a_r + \varepsilon \\ 1 & a_r + \varepsilon \le t < a_{r+1} - \varepsilon \\ b\left(\dfrac{a_{r+1}-t}{\varepsilon}\right) & a_{r+1} - \varepsilon \le t < a_{r+1} + \varepsilon \end{cases} . \tag{5.12}$$

Here, the cutoff function $b(t)$ is chosen as [152]

$$b(t)=\sin\left(\frac{\pi}{4}+\frac{\pi}{4}\sin\left(\frac{\pi}{2}\sin\frac{\pi t}{2}\right)\right),\ -1\le t\le 1\,. \tag{5.13}$$

With this choice of bell function (5.12) and cutoff function (5.13), these LCB functions $T_{ru}(t)$ are locally supported on the interval $[a_r-\varepsilon, a_{r+1}+\varepsilon]$ and they become orthonormal, i.e. $\langle T_{ru}(t), T_{sw}(t)\rangle = \delta_{rs}\delta_{uw},$, $\forall r,s<N,\ \forall u,w\in\mathbb{N}$, where δ_{rs} and δ_{uw} are Kronecker delta functions. Furthermore, the LCB functions are localized in frequency. Specifically, their Fourier transforms $\overline{T}_{ru}(\omega)$ are

$$\overline{T}_{ru}(\omega)=\sqrt{\frac{1}{2M^r\Delta t}}\left(e^{-ia_r\omega_u}\overline{B}_r(\omega-\omega_u)+e^{ia_r\omega_u}\overline{B}_r(\omega+\omega_u)\right) \tag{5.14}$$

where $\omega_u=\pi(u-1/2)/(M^r\Delta t)$ is the central frequency of $\overline{T}_{ru}(\omega)$ and $\overline{B}_r(\omega)$ is the Fourier transform of the bell function $B_r(t)$. Due to the smoothness (i.e., the quasi-bandlimitedness) of the bell function, the LCB functions are also quasi-bandlimited. For example, the LCB functions on a fixed interval with three different central frequencies are plotted [Figure 5.1(a)-(c)]. Clearly, these functions are localized in time and frequency.

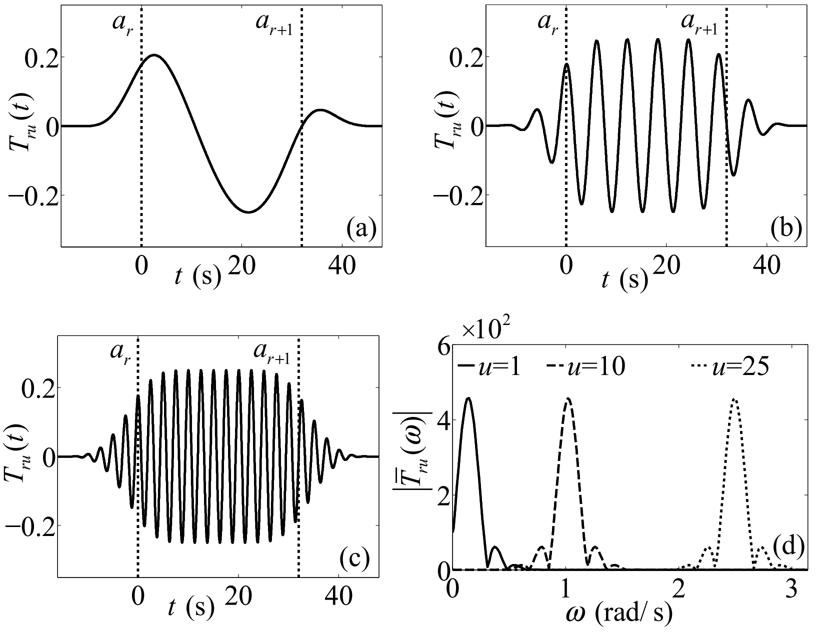


Figure 5.1 The LCB functions $T_{ru}(t)$ associated with interval $[\mathrm{a}_r, \mathrm{a}_{r+1}]$, $\mathrm{a}_r = 0$, $\Delta t = 1\,\mathrm{s}$, $(740, 245)\,\mathrm{MHz}$, $\varepsilon = 16$ with different central frequencies: (a) $u = 1$, (b) $u = 10$, (c) $u = 25$. (d) The Fourier transform $\bar{T}_{ru}(\omega)$ of (a), (b) and (c).

Next, the outgoing/incoming rays can be represented using the LCBs (5.11). The outgoing/incoming rays of a level v box in direction $\hat{\mathbf{k}}_{qp}^{v}$ consist of two transverse components, i.e., $\mathrm{G}_{l,\alpha}^{\pm}(\hat{\mathrm{k}}_{qp}^{v}, t) = G_\theta(\hat{\mathrm{k}}_{qp}^{v}, t)\hat{\boldsymbol{\theta}} + G_\varphi(\hat{\mathrm{k}}_{qp}^{v}, t)\hat{\boldsymbol{\varphi}}$, each transverse component is expressed using LCBs as

$$G_\psi(\hat{\mathrm{k}}_{qp}^{v}, t) = \sum_{r=1}^{N} \sum_{u=1}^{\infty} I_{ru}^{\psi}(\hat{\mathrm{k}}_{qp}^{v}) T_{ru}(t) . \tag{5.15}$$

Here, $\psi = \{\theta, \varphi\}$ and the LCB coefficients are $I_{ru}^{\psi}(\hat{\mathrm{k}}_{qp}^{v}) = \left\langle G_\psi(\hat{\mathrm{k}}_{qp}^{v}, t), T_{ru}(t) \right\rangle$ due to the orthonormality of the LCBs. In practice, these coefficients can be efficiently computed by first sampling the outgoing/incoming ray with time step size Δt and then applying to the resulting sequence the discrete local cosine transform (LCT) that requires $O(M^r \log M^r)$ operations for each interval [152]. The LCT results in a coefficient vector:

$$\bar{\mathbf{I}}_\psi(\hat{\mathbf{k}}_{qp}^{v}) = [I_{11}^{\psi}(\hat{\mathbf{k}}_{qp}^{v}), \ldots, I_{1M^1}^{\psi}(\hat{\mathbf{k}}_{qp}^{v}), \ldots, I_{N1}^{\psi}(\hat{\mathbf{k}}_{qp}^{v}), \ldots, I_{NM^N}^{\psi}(\hat{\mathbf{k}}_{qp}^{v})]^{\dagger}. \tag{5.16}$$

Note that the number of LCB coefficients associated with the r^{th} interval is M^r and the total number of LCB coefficients is $N_I = \sum_{r=1}^{N} M^r$. With a proper choice of the partition (a_r), the coefficient vector $\bar{\mathbf{I}}_\psi(\hat{\mathbf{k}}_{qp}^{v})$ exhibits fast decaying rate on each interval [153] and only those coefficients with magnitudes exceeding a prescribed threshold need to be stored. Based on the temporal feature of the ray data, two types of efficient partitioning mechanisms are considered in this work.

- *Single-resolution* bases: Partitioning the support with small intervals of same length, i.e., $M^r = M^{min}, \forall r$, and computing the LCB coefficients by $O(N_I \log M^{min})$ operations.

- *Multi-resolution* bases: Combining adjacent intervals of the single-resolution bases that (i) have similar maximum LCB coefficients and (ii) have fast decaying coefficients into larger intervals and computing the LCB coefficients on these new intervals by at most $O(N_I \log M^{max})$ operations where $M^{max}\Delta t$ is the maximum allowed interval length. In comparison, the conventional method to find the partitions that yield maximum sparsity requires $O(N_I \log^2 N_I)$ operations [152].

Note that in our implementation, the minimum and maximum interval lengths are chosen as constants, e.g. $M^{min} = 16$, and $M^{max} = 256$, hence computing the LCB coefficients with single- or multi-resolution bases requires $O(N_I)$ operations.

In general, the multi-resolution bases can achieve better sparsity in the LCB coefficients compared with the single-resolution bases, however, these bases (i.e., the optimal partitions) vary for each outgoing/incoming ray, which posts challenges to convert the PWTD operations into the wavelet domain. Therefore, in the proposed LCB-enhanced PWTD scheme, the computation of the outgoing/incoming rays is directly conducted in the wavelet domain using single-resolution bases; once computed, the ray data is stored as its LCB coefficients using multi-resolution bases. Next, the LCB-enhanced PWTD stages are explained in detail.

5.3.2 Translation in the Wavelet Domain

During the translation stage at level v, the outgoing/incoming rays $\mathrm{G}^{\pm}_{l,\alpha}(\hat{\mathrm{k}}^{v}_{qp},t)$ are represented using their LCB coefficient vectors $\overline{\mathbf{I}}^{\pm}_{\psi}(\hat{\mathbf{k}}^{v}_{qp})$ in the single-resolution bases with $M^{r} = M^{min}$. Specifically, consider a far field box pair α and α'. The coefficient vector of the incoming ray $\overline{\mathbf{I}}^{-}_{\psi}(\hat{\mathbf{k}}^{v}_{qp})$ (of length $N^{-}_{I} = M^{min}N^{-}$) in observer box α' can be computed by translating that of the outgoing ray $\overline{\mathbf{I}}^{+}_{\psi}(\hat{\mathbf{k}}^{v}_{qp})$ (of length $N^{+}_{I} = M^{min}N^{+}$) in source box α as

$$\bar{\mathbf{I}}_{\psi}^{-}(\hat{\mathbf{k}}_{qp}^{v}) = \bar{\bar{\mathbf{T}}}\bar{\mathbf{I}}_{\psi}^{+}(\hat{\mathbf{k}}_{qp}^{v}) \tag{5.17}$$

where the translation matrix $\bar{\bar{\mathbf{T}}}$ consists of matrix blocks $\bar{\bar{\mathbf{T}}}_{sr}$, $s=1,...,N^{-}$, $r=1,...,N^{+}$ of dimension $M^{min} \times M^{min}$. The entries of these matrix blocks $\bar{\bar{\mathbf{T}}}_{sr}$ are [154]

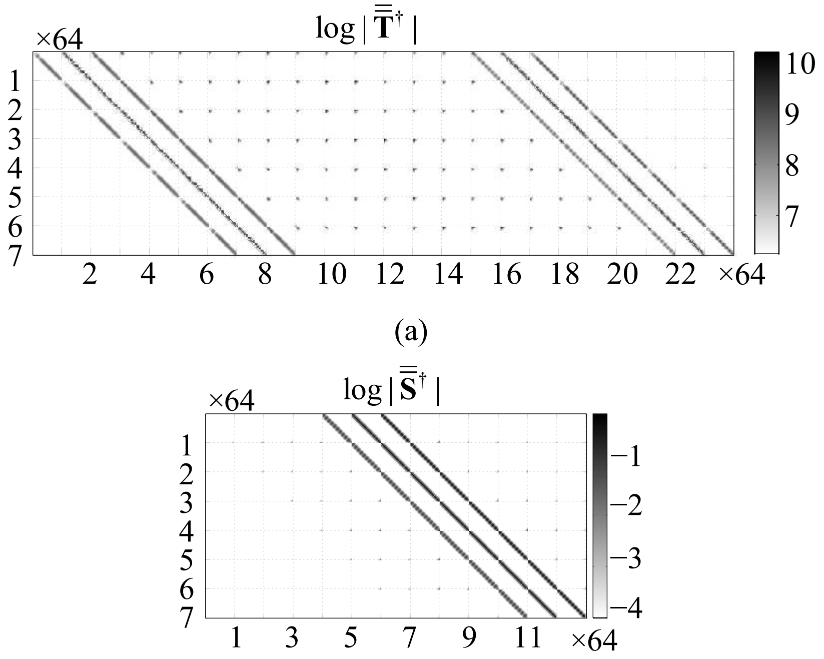
$$\begin{aligned} \{\bar{\bar{\mathbf{T}}}_{sr}\}_{wu} &= \left\langle T_{sw}^{-}(t), \mathcal{T}(\hat{\mathbf{k}}_{qp}^{v}, t) * T_{ru}^{+}(t) \right\rangle \\ &= \frac{1}{4\pi M^{min}\Delta t} \int_{-\infty}^{+\infty} d\omega [\bar{B}_1^{+}(\omega-\omega_u) + \bar{B}_1^{+}(\omega-\omega_u)] \\ &\times [\bar{B}_1^{-}(-\omega-\omega_w) + \bar{B}_1^{-}(-\omega-\omega_w)] e^{j\omega(a_s^{-}-a_r^{+})} \bar{\mathcal{T}}(\hat{\mathbf{k}}_{qp}^{v}, \omega). \end{aligned} \tag{5.18}$$

Here, $T_{ru}^{+}(t)$ and $T_{sw}^{-}(t)$, $u,w=1,...,M^{min}$ are the LCB functions associated with intervals staring with a_r^{+} and a_s^{-}, respectively. $\bar{B}_1^{\pm}(\omega)$ are the Fourier transformed bell functions associated with the first intervals of the outgoing/incoming ray. $\bar{\mathcal{T}}(\hat{\mathbf{k}}_{qp}^{v}, \omega)$ is the Fourier transform of the translation function $\mathcal{T}(\hat{\mathbf{k}}_{pq}^{v}, t)$ [8]. Note that $\bar{\bar{\mathbf{T}}}_{sr} = \bar{\bar{\mathbf{T}}}_{(s-r+1)1}$ if $s \geq r$ and $\bar{\bar{\mathbf{T}}}_{sr} = \bar{\bar{\mathbf{0}}}$ if $s < r$. Hence, only matrix blocks $\bar{\bar{\mathbf{T}}}_{s1}$, $s=1,...,N^{-}$ need to be computed. The computation of these block entries can be further facilitated by the following two observations: (i) $\{\bar{\bar{\mathbf{T}}}_{sr}\}_{wu} = 0$ if $|u-w|$ is large, due to the narrow band nature of the LCB functions; (ii) when u is large, the translated LCB functions become temporally localized (see Appendix):

$$\begin{aligned} \mathcal{T}(\hat{\mathbf{k}}_{qp}^{v}, t) * T_{ru}^{+}(t) \approx & \frac{c_0}{R_{c,\alpha\alpha'}} T_{ru}^{+}(t - R_{c,\alpha\alpha'} / c_0) f_1(\cos\theta) \\ & - \frac{c_0}{R_{c,\alpha\alpha'}} T_{ru}^{+}(t + R_{c,\alpha\alpha'} / c_0) f_2(\cos\theta) \end{aligned} \tag{5.19}$$

where $f_1(\cos\theta) = \sum_{k=0}^{K^v} (-1)^k (2k+1) \Phi_k(\cos\theta)$ and $f_2(\cos\theta) = \sum_{k=0}^{K^v} (2k+1) \Phi_k(\cos\theta)$, i.e., the translated LCB function is locally supported on the interval $[a_r^{+} - \varepsilon \pm R_{c,\alpha\alpha'} / c_0,\ a_{r+1}^{+} + \varepsilon \pm R_{c,\alpha\alpha'} / c_0]$. Therefore, the nontrivial entries in the matrix blocks can be efficiently identified and computed. For example, one translation matrix is plotted in Figure 5.2(a). Note that the translation matrices $\bar{\bar{\mathbf{T}}}$ depend on the direction $\hat{\mathbf{k}}_{qp}^{v}$

and the vector $\mathbf{R}_{c,\alpha\alpha'}$ connecting the box centers, hence can be reused for box pairs with the same $\mathbf{R}_{c,\alpha\alpha'}$. After the translation stage at level v is completed, the incoming rays are stored as their coefficient vectors in the multi-resolution bases.



(a)

(b)

Figure 5.2 (a) Magnitudes of a translation matrix $\overline{\overline{\mathbf{T}}}$ obtained by setting $\chi_t = 10$, $R^v = 3.5\lambda$, $\mathbf{R}_{c,\alpha\alpha'} = [24\lambda\ 0\ 0]^{\dagger}$, $\hat{\mathbf{k}}_{qp}^{v} = [0\ 0\ 1]^{\dagger}$, $M^{min} = 64$, $N^{+} = 7$, and $N^{-} = 24$. (b) Magnitudes of a shifting matrix $\overline{\overline{\mathbf{S}}}$ obtained by setting $\chi_t = 10$, $R^v = 7\lambda$, $R^{v-1} = 3.5\lambda$, $\mathbf{R}_{c,\alpha\alpha'} = [4\lambda\ 4\lambda\ 4\lambda]^{\dagger}$, $\hat{\mathbf{k}}_{qp}^{v} = [1/\sqrt{3}\ \ 1/\sqrt{3}\ \ 1/\sqrt{3}]^{\dagger}$, $M^{min} = 64$, $N^{+} = 7$, and $N^{+\prime} = 13$. The accuracy is set to 1×10^{-4}.

5.3.3 Spherical Interpolation/Filtering in the Wavelet Domain

The outgoing/incoming rays in $v > 1$ boxes are constructed/projected by spherical interpolation/filtering. [Note that those in $v = 1$ boxes are constructed/projected by (5.6) /(5.9) followed by/following the compression/reconstruction of the ray data using (5.16).]

In this subsection, only the spherical interpolation used to construct the outgoing rays in the wavelet domain is expounded, as the spherical filtering for projection of the incoming rays can be carried out similarly.

Consider construction of outgoing rays $\mathrm{G}^{+}_{l',\alpha'}(\hat{\mathrm{k}}^{v}_{qp},t)$, $q=0,...,K^{v}$, $p=-K^{v},...,K^{v}$ in level v box α' from the outgoing rays $\mathrm{G}^{+}_{l,\alpha}(\hat{\mathrm{k}}^{v-1}_{qp},t)$, $q=0,...,K^{v-1}$, $p=-K^{v-1},...,K^{v-1}$ in its child box α . Let $\overline{\mathbf{I}}^{+\prime}_{\psi}(\hat{\mathbf{k}}^{v}_{qp})$ and $\overline{\mathbf{I}}^{+}_{\psi}(\hat{\mathbf{k}}^{v-1}_{qp})$ denote the LCB coefficient vectors that represent $\mathrm{G}^{+}_{l',\alpha'}(\hat{\mathrm{k}}^{v}_{qp},t)$ and $\mathrm{G}^{+}_{l,\alpha}(\hat{\mathrm{k}}^{v-1}_{qp},t)$ using single-resolution bases, respectively.

First, the coefficient vectors $\overline{\mathbf{I}}^{+}_{\theta}(\hat{\mathbf{k}}^{v}_{qp})$ and $\overline{\mathbf{I}}^{+}_{\varphi}(\hat{\mathbf{k}}^{v}_{qp})$ whose entries form the quantities $I^{\theta}_{ru}(\hat{\mathrm{k}}^{v}_{qp})\hat{\boldsymbol{\theta}}+I^{\varphi}_{ru}(\hat{\mathrm{k}}^{v}_{qp})\hat{\boldsymbol{\varphi}}$ are obtained by directly applying the global vector spherical interpolation scheme [13] to the quantities $I^{\theta}_{ru}(\hat{\mathrm{k}}^{v-1}_{qp})\hat{\boldsymbol{\theta}}+I^{\varphi}_{ru}(\hat{\mathrm{k}}^{v-1}_{qp})\hat{\boldsymbol{\varphi}}$ obtained from $\overline{\mathbf{I}}^{+}_{\theta}(\hat{\mathbf{k}}^{v-1}_{qp})$ and $\overline{\mathbf{I}}^{+}_{\theta}(\hat{\mathbf{k}}^{v-1}_{qp})$.

Next, the interpolated LCB coefficients are shifted from the center of box α to that of box α'. Note that, in time domain, the outgoing ray $\mathrm{G}^{+}_{l',\alpha'}(\hat{\mathrm{k}}^{v}_{qp},t)$ in box α' is obtained by shifting the interpolated outgoing ray $\mathrm{G}^{+}_{l,\alpha}(\hat{\mathrm{k}}^{v}_{qp},t)$ in box α as

$$\begin{aligned}\mathbf{G}^{+}_{l',\alpha'}(\hat{\mathbf{k}}^{v}_{qp},t) &= \mathcal{S}(\hat{\mathbf{k}}^{v}_{qp},t)*\mathbf{G}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{qp},t)\\ &=\mathbf{G}^{+}_{l,\alpha}(\hat{\mathbf{k}}^{v}_{qp},t-\hat{\mathbf{k}}^{v}_{qp}\cdot\mathbf{R}_{c,\alpha\alpha'})\ .\end{aligned} \tag{5.20}$$

In contrast, in the wavelet domain, the coefficient vector $\overline{\mathbf{I}}^{+\prime}_{\psi}(\hat{\mathbf{k}}^{v}_{qp})$ can be computed by

$$\overline{\mathbf{I}}^{+\prime}_{\psi}(\hat{\mathbf{k}}^{v}_{qp})=\overline{\overline{\mathbf{S}}}\,\overline{\mathbf{I}}^{+}_{\psi}(\hat{\mathbf{k}}^{v}_{qp}). \tag{5.21}$$

Here $\overline{\overline{\mathbf{S}}}$ is the shifting matrix similar to the translation matrix $\overline{\overline{\mathbf{T}}}$, the entries of matrix block $\overline{\overline{\mathbf{S}}}_{sr}$, $r=1,...,N^{+}$, $s=1,...,N^{+\prime}$ where $N^{+\prime}$ denotes the number of intervals in the support of $\mathbf{G}^{+}_{l',\alpha'}(\hat{\mathbf{k}}^{v}_{qp},t)$, are expressed as

$$\{\overline{\overline{\mathbf{S}}}_{sr}\}_{wu}=\left\langle T^{+\prime}_{sw}(t),\mathcal{S}(\hat{\mathbf{k}}^{v}_{qp},t)*T^{+}_{ru}(t)\right\rangle. \tag{5.22}$$

Here, $u, w = 1, ..., M^{min}$, $T_{sw}^{+\prime}(t)$ are the LCB functions associated with the s^{th} interval in the support of $\mathbf{G}_{l',\alpha'}^{+}(\hat{\mathbf{k}}_{qp}^{v}, t)$. Just like the translation matrix, the shifting matrix is highly sparse [see the example in Figure 5.2(b)] and can be reused for boxes α and α' that have the same $\mathbf{R}_{c,\alpha\alpha'}$. After the contributions from all child boxes have been accounted for, the LCB coefficient vector $\overline{\mathbf{I}}_{\psi}^{+\prime}(\hat{\mathbf{k}}_{qp}^{v})$ of box α is stored using multi-resolution bases.

5.3.4 Computational Complexity

In this section, the memory requirement and computational complexity of the proposed LCB-enhanced PWTD scheme are discussed for smooth quasi-planar structures under plane wave incidence. Note that these scaling estimations hinge on the compression performance of the outgoing/incoming ray data using the LCBs. For a general structure, the ray data consists of locally smooth parts that resemble the incident pulses, as well as the locally non-smooth parts due to the APS function $T^{APS}(t)$ used to section the temporal signature $f_n(t)$. These non-smooth parts result in locally very poor sparsity when represented using the LCBs and can degrade the overall compression performance. However, for quasi-planar structures under certain types of excitation, the current density induced on the surface propagates along certain predominant direction, which generates specific sparsity patterns in the outgoing/incoming ray data. These sparsity patterns greatly alleviate the abovementioned performance degradation and can help reduce the memory and computational costs of the LCB-enchend PWTD scheme. In what follows, the memory requirement and computational complexity of the proposed LCB-enhanced PWTD scheme are discussed for two types of plane wave incidence.

5.3.4.1 Near-normal incidence

In this case, the magnitudes of the non-smooth parts of the ray data become vanishingly small when the box size is electrically large [Note: as the overall memory and computational costs are dominated by those at higher levels of the PWTD tree, its assumed in this section that the box sizes are electrically large]. As a result, the memory requirement for storing one outgoing/incoming ray in the proposed scheme scales nearly as $O(1)$, as opposed to $O(M^v)$ in the classical PWTD scheme. As there is ray data in N_k^v directions associated with N_g^v boxes, the overall memory requirement scales as $\sum_{v=1}^{N_L} N_g^v N_k^v O(1) = O(N_s \log N_s)$.

The computational costs of the translation stage are dominated by the wavelet-domain translation operation in (5.17). Although the translation matrix $\bar{\bar{\mathbf{T}}}$ is highly sparse, its first few columns (corresponding to source LCB functions $T_{ru}^{+}(t)$ with small r) have $O(N_I^-) = O(M^v)$ nonzero entries; the rest columns have $O(1)$ nonzero entries due to the localization property shown in (5.19). Since each LCB coefficient vector $\bar{\mathbf{I}}_\psi^+(\hat{\mathbf{k}}_{qp}^v)$ has $O(1)$ nonzero entries, the computational cost of each translation operation scales at most as $O(M^v)$. As there are $O(1)$ nontrivial outgoing rays at each direction in one source box, the overall computational complexity of the translation stage scales as $\sum_{v=1}^{N_L} N_g^v N_k^v O(M^v) O(1) = O(N_s^{1.5})$.

The computational costs of the spherical interpolation stage are dominated by two operations: the wavelet-domain spherical interpolation and shifting. The spherical interpolation of one entry of the coefficient vectors $\bar{\mathbf{I}}_\psi^+(\hat{\mathbf{k}}_{qp}^{v-1})$ requires $O(N_k^v \log K^v)$ operations [132]; in contrast, shifting one entry of the interpolated coefficient vectors $\bar{\mathbf{I}}_\psi^+(\hat{\mathbf{k}}_{qp}^v)$ by (5.21) in all directions requires $O(N_k^v)$ operations as there are $O(1)$ nonzero entries in each column of the shifting matrix $\bar{\bar{\mathbf{S}}}$. Since there are $O(1)$ nontrivial outgoing coefficients vectors each having $O(1)$ nonzero entries for $O(N_g^v)$ source boxes, the computational complexity of the spherical interpolation stage scales as $\sum_{v=1}^{N_L} N_g^v O(1) O(1) O(N_k^v \log K^v) = O(N_s \log^2 N_s)$. Note: similar analysis can be performed for the spherical filtering stage, however, as there are $O(N_t / M^v)$ nontrivial incoming coefficient vectors in each box as opposed to $O(1)$ outgoing ones, the complexity reduction of the spherical filtering stage is not significant compared to the spherical interpolation and translation stages.

5.3.4.2 Very-oblique incidence

If non-smooth parts of the ray data exist, it's assumed that the compression performance of the ray data is dominated by that of the non-smooth part. The compression performance can be analyzed, without loss of generality, by an example [Figure 5.3]. In Figure 5.3, a set of Hertzian dipoles, with locations $\mathbf{r}_n = (x_n, y_n, z_n)$ and orientations $\hat{\mathbf{u}}_n$, are randomly scattered across a smooth quasi-planar surface S (approximately) parallel to the *x-y* plane. Here, $|z_n| <= \varepsilon_z$ for some constant $\varepsilon_z << \lambda$. The temporal signature of the n^{th} dipole is $f_n(t) = F(t - \mathbf{r}_n \cdot \hat{\mathbf{k}} / c_0)$. Here, $F(t)$ is an incidence pulse with time width t_{bw}, and $\hat{\mathbf{k}} = (\sin\theta^i, 0, -\cos\theta^i)$, $\theta^i \le 90^{\mathrm{o}}$ and sufficiently large, is the incidence direction. Let α be a level v box centered at origin with edge length $L^v = 2R^v / \sqrt{3}$. The outgoing ray $\mathbf{G}^+_{l,\alpha}(\hat{\mathbf{k}}^v_{qp}, t)$, $\hat{\mathbf{k}}^v_{qp} = (\cos\phi^v_p \sin\theta^v_q, \sin\phi^v_p \sin\theta^v_q, \cos\theta^v_q)$, is constructed by (5.6) with $\mathbf{S}_n(\mathbf{r}) = \hat{\mathbf{u}}_n \delta(\mathbf{r} - \mathbf{r}_n)$. It's assumed that the each subsignal $f^l_n(t)$ is locally non-smooth near $t = lM^v \Delta t$ (or $t = (l-1)M^v \Delta t$) due to the presence of the APS function. Moreover, it's only locally non-smooth for those subsignals associated with dipoles confined in a rectangular box of dimension $(c_0 t_{bw} / \sin\theta^i) \times L^v \times 2\varepsilon_z$ [See Figure 5.3]. From (5.6) and (5.9), the outgoing ray $\mathbf{G}^+_{l,\alpha}(\hat{\mathbf{k}}^v_{qp}, t)$ becomes locally non-smooth near time

$$\begin{aligned} t &= lM^v \Delta t + \hat{\mathbf{k}}^v_{qp} \cdot (\mathbf{r}_n - \mathbf{r}^c_s) / c_0 \\ &= lM^v \Delta t + (\cos\phi^v_p \sin\theta^v_q x_n + \sin\phi^v_p \sin\theta^v_q y_n + \cos\theta^v_q z_n) / c_0. \end{aligned} \tag{5.23}$$

The support length of the non-smooth part of the outgoing ray is therefore at most $t_{bw} |\cos\phi^v_p \sin\theta^v_q| / \sin\theta^i + L^v |\sin\phi^v_p \sin\theta^v_q| / c_0 + 2\varepsilon_z |\cos\theta^v_q| / c_0$. Apparently, when ϕ^v_p is small (i.e., ray direction close to the plane of incidence), the memory cost for storing one outgoing ray using LCBs scales nearly as $O(1)$; when ϕ^v_p is close to 90^{o} (i.e., ray direction away from the plane of incidence), the memory cost for storing one outgoing ray using LCBs scales as $O(M^v)$. Note: similar analysis and results can be obtained for the incoming rays.

As the memory reduction is not significant for ray data in directions that correspond to ϕ_p^v near 90°, the scaling estimates of the memory and computation of the LCB-enhanced PWTD scheme remain the same as those of the conventional PWTD scheme. That said, these costs can still be significantly reduced (i.e., with smaller leading constants) by the proposed LCB-enhanced PWTD scheme.

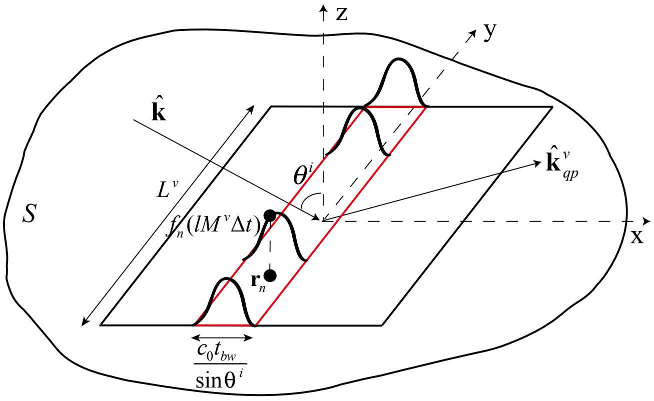


Figure 5.3 Non-smooth part of the outgoing rays from one box with very oblique incidence.

5.4 Numerical Results

This section presents numerical results that demonstrate the efficiency, accuracy, and applicability of the proposed LCB-enhanced PWTD-TD-SIE solver. In all examples considered here, the temporal basis function $T(t)$ is the fourth-order piecewise Lagrange polynomial. Unless otherwise stated, the structures are illuminated by a plane wave with electric field as

$$\mathbf{E}^i(\mathbf{r},t) = \hat{\mathbf{p}}F(t - \mathbf{r}\cdot\hat{\mathbf{k}}/c_0). \tag{5.24}$$

In (5.24), $F(t) = \cos[2\pi f_0(t-t_0)]e^{-(t-t_0)^2/2\sigma^2}$ is a modulated and quasi-bandlimited Gaussian pulse, f_0 is the modulation frequency, $t_0 = 6\sigma$ is the delay of the pulse, $\sigma = 3/(2\pi f_{bw})$ is a measure of pulse duration, the parameter $2f_{bw}$ represents the essential bandwidth. Note that the maximum angular frequency is $\omega_{max} = 2\pi(f_0 + f_{bw})$. $\hat{\mathbf{p}}$ and $\hat{\mathbf{k}}$ denote the polarization and propagation direction of the plane wave. All frequency domain (i.e., time harmonic) data presented in this section is obtained by dividing the Fourier transform of the time domain data by that of $F(t)$. A GMRES algorithm and a diagonal preconditioner are used to iteratively solve (5.2) at each time step. The GMRES iteration is terminated when the condition

$$\| \bar{\mathbf{V}}_j - \bar{\bar{\mathbf{Z}}}_0 \bar{\mathbf{I}}_j^{(n)} \| < \delta \| \bar{\mathbf{V}}_j \| \tag{5.25}$$

is satisfied. Here, $\bar{\mathbf{I}}_j^{(n)}$ represents the vector of current coefficients in the n^{th} iteration, $\bar{\mathbf{V}}_j = \bar{\mathbf{F}}_j - \sum_{i=1}^{j-1} \bar{\bar{\mathbf{Z}}}_i \bar{\mathbf{I}}_{j-i}$ is total RHS at time step j, and δ is the desired residual error. Here δ is set to 10^{-12}.

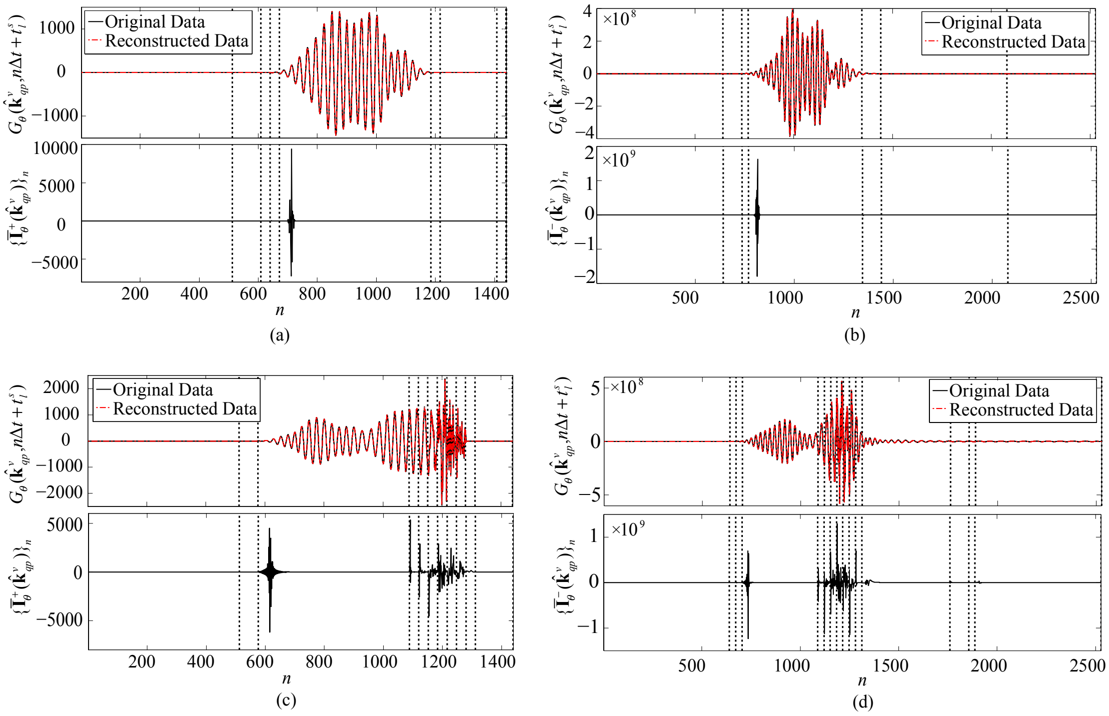
The simulations are performed on two computing platforms: (i) A Sandy Bridge compute-node that has four Eight-Core 2.40 GHz Intel Xeon E5-4640 processors and 1 TB memory. (ii) A Sandy Bridge cluster where each node has two Eight-Core 2.60 GHz Intel Xeon E5-2670 processors and 64 GB memory. The proposed solver is parallelized by the scalable parallelization scheme described in Chapter 4. One MPI process is launched per compute-node and OpenMP utilizes all cores on each node.

5.4.1 Compression of One Ray

First, the compression performance of representing one outgoing/incoming ray using the LCBs is investigated. To this end, consider a far field box pair α and α'. Suppose the source box α encloses a square plate centered at origin and located in the *x*-*y* plane. A set of Hertzian dipoles are randomly scattered across the plate with orientations $\hat{\mathbf{u}}_n$ and positions $\mathbf{r}_n$, $n \in \alpha$. The temporal signature of the n^{th} dipole is $f_n(t) = F(t - \mathbf{r}_n \cdot \hat{\mathbf{k}} / c_0)$ with $f_0 = 7.68\text{ GHz}$, $f_{bw} = 2.56\text{ GHz}$ and time step size $\Delta t = 4.88\text{ ps}$. The outgoing ray is constructed by (5.6) with $\mathbf{S}_n(\mathbf{r}) = \hat{\mathbf{u}}_n \delta(\mathbf{r} - \mathbf{r}_n)$ and the incoming ray is computed by the translation operation in (5.7). Such ray data is compressed and reconstructed using its multi-resolution LCB coefficients with accuracy set to 1×10^{-3}. For simplicity, only the θ component of the ray data is investigated as similar compression performance regarding the φ component can be achieved.

The entries of the coefficient vector $\{\overline{\mathbf{I}}_{\theta}^{\pm}(\hat{\mathbf{k}}_{qp}^{v})\}_{n}$ and samples of the ray data $G_{\theta}(\hat{\mathbf{k}}_{qp}, n\Delta t + t_{l}^{s})$, $n = 1,\ldots,N_{I}^{\pm}$ obtained with $R^{v} = 14\lambda$ are plotted for $\hat{\mathbf{k}} = \hat{\mathbf{z}},\ \hat{\mathbf{k}}_{qp}^{v} = -\hat{\mathbf{x}}$ [Figure 5.4(a),(b)] and $\hat{\mathbf{k}} = \hat{\mathbf{x}},\ \hat{\mathbf{k}}_{qp}^{v} = -\hat{\mathbf{x}}$ [Figure 5.4(c),(d)], respectively. Note that the vertical dashed lines correspond to the ending points $a_{r}^{\pm}$, $r = 2,\ldots,N^{\pm}$ of the intervals. When $\hat{\mathbf{k}} = \hat{\mathbf{z}}$, the temporal signature given by $F(t - \mathbf{r}_{n} \cdot \hat{\mathbf{k}} / c_{0})$ resembles that associated with the current induced by normal incidence, hence the outgoing/incoming ray is smooth (when the box is electrically large) and can be well compressed by the LCBs. In contrast, when $\hat{\mathbf{k}} = \hat{\mathbf{x}}$, the temporal signature resembles that associated with the current induced by incidence at a very oblique angle. In this case, the outgoing/incoming ray has locally non-smooth parts due to the presence of the APS function. These non-smooth parts can be automatically identified by the LCBs [Figure 5.4(c),(d)].

The compression ratio κ is defined as the length of the coefficient vector $N_{I}^{\pm}$ divided by the number of nontrivial entries in the vector. The compression ratios of the outgoing [Figure 5.5(a)] and incoming [Figure 5.5(b)] ray with $\hat{\mathbf{k}} = \hat{\mathbf{z}},\ \hat{\mathbf{k}}_{qp}^{v} = \hat{\mathbf{y}}$, $\hat{\mathbf{k}} = \hat{\mathbf{z}},\ \hat{\mathbf{k}}_{qp}^{v} = \hat{\mathbf{z}}$, $\hat{\mathbf{k}} = \hat{\mathbf{x}},\ \hat{\mathbf{k}}_{qp}^{v} = \hat{\mathbf{y}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{x}},\ \hat{\mathbf{k}}_{qp}^{v} = \hat{\mathbf{z}}$ obtained from boxes with different radius are plotted. For $\hat{\mathbf{k}} = \hat{\mathbf{z}}$, the compression ratio improves as the box size increases, which justifies our assumption about the $O(1)$ memory requirement to store one ray in Section 5.3.4.1. However, when $\hat{\mathbf{k}} = \hat{\mathbf{x}}$ (i.e., $\theta^{i} = 90^{\circ}$), the compression ratio in direction $\hat{\mathbf{k}}_{qp}^{v} = \hat{\mathbf{y}}$ (i.e., $\phi_{p}^{v} = 90^{\circ}$) remains the same as the box size increases. That said, significant memory reduction can be achieved when storing the ray data in all directions using LCBs.



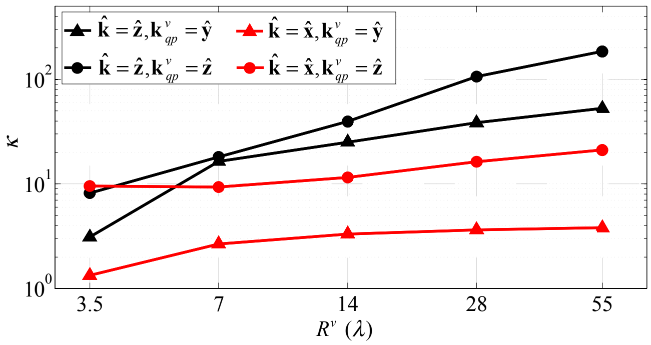
(a)

(b)

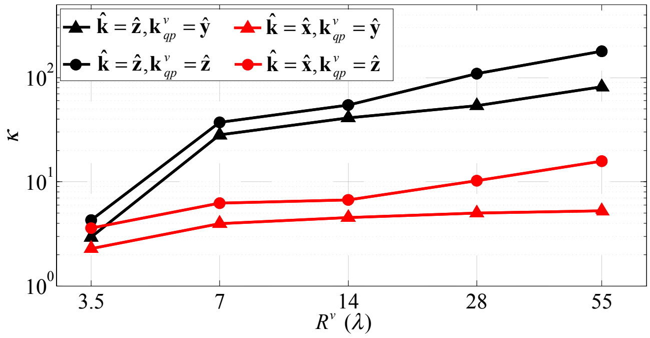
(c)

(d)

Figure 5.4 LCB representation of one ray (θ component) in direction $\hat{\mathbf{k}}_{qp}^{v} = -\hat{\mathbf{x}}$ of a box of radius $R^{v} = 14\lambda$. (a) Outgoing ray, $\hat{\mathbf{k}} = \hat{\mathbf{z}}$. (b) Incoming ray, $\hat{\mathbf{k}} = \hat{\mathbf{z}}$. (c) Outgoing ray, $\hat{\mathbf{k}} = \hat{\mathbf{x}}$. (a) Incoming ray, $\hat{\mathbf{k}} = \hat{\mathbf{x}}$.



(a)

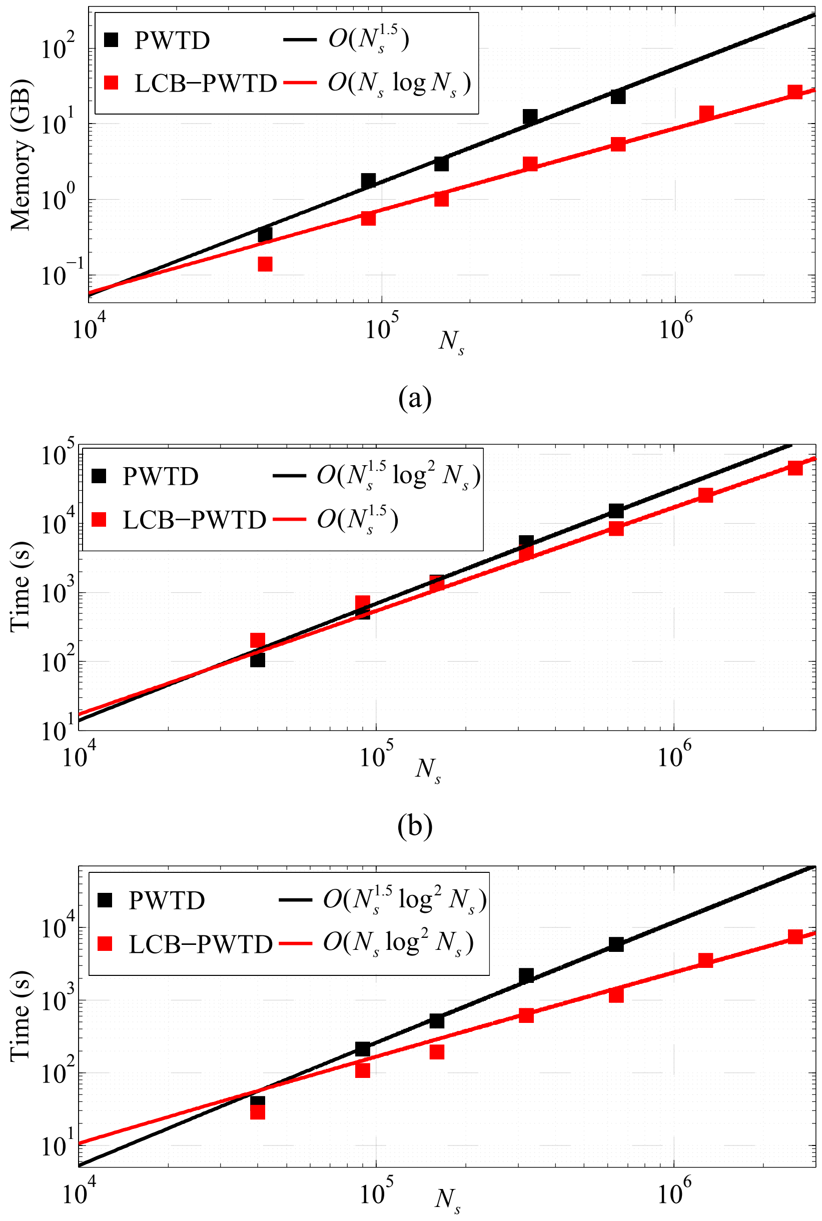


(b)

Figure 5.5 Compression ratio κ of one ray in a box with radius R^v varying from 3.5λ to 55λ. (a) Outgoing ray. (b) Incoming ray.

5.4.2 Complexity Validation

Next, the memory requirement and computational complexity of the proposed LCB-enhanced PWTD scheme are validated. To this end, a set of N_s Hertzian dipoles are randomly scattered across a square. The orientation and position of the n^{th} dipole are $\hat{\mathbf{u}}_n$ and $\mathbf{r}_n$, $n=1,\ldots,N_s$. The temporal signature of all dipoles is $f_n(t)=F(t)$ with $f_0=768$ MHz, $f_{bw}=256$ MHz and discretized using step size $\Delta t=62.5$ ps. The number of time steps is chosen as $N_t=3.44N_s^{0.5}$. The ray data is compressed using LCBs with $M^{min}=16$ and accuracy set to 10^{-4}. In the test, N_s is increased from $40{,}000$ to $2{,}560{,}000$ while the edge length of the plate is changed from 20 λ to 160 λ and edge length of the boxes at the finest level of PWTD tree is set to 1.25λ. The test is performed on the Sandy Bridge cluster with 8 processors. The memory costs on one processor for the classical PWTD scheme and the LCB-enhanced PWTD schemes are plotted in Figure 5.6(a). Those costs comply with the theoretical estimates of $O(N_s^{1.5})$ and $O(N_s\log N_s)$, respectively. Apparently, the LCB-enhanced PWTD scheme achieves significant memory reduction compared with the classical PWTD scheme. The computation times for the translation and spherical interpolation stage of the classical PWTD and LCB-enhanced PWTD scheme are plotted [Figure 5.6(b),(c)], those of the LCB-enhanced PWTD scheme are in good agreement with the theoretical predictions of $O(N_s^{1.5})$ and $O(N_s \log^2 N_s)$, as opposed to the $O(N_s^{1.5} \log^2 N_s)$ scaling of the classical PWTD scheme.



(a)

(b)

(c)

Figure 5.6 (a) Memory cost of the ray data on one processor, and computation time of the (b) translation and (c) spherical interpolation stages of the PWTD and LCB-

PWTD schemes using the Sandy Bridge cluster. N_s is changed from $40,000$ to $2,560,000$.

5.4.3 Canonical Example

First, the proposed LCB-enhanced PWTD-TD-CFIE solver is applied to the analysis of transient scattering from a NASA almond that can be enclosed by a hypothetical box with dimensions $25\text{ cm}\times 9.7\text{ cm}\times 3.2\text{ cm}$. The almond is illuminated by a modulated Gaussian plane wave with $f_0 = 42$ GHz and $f_{bw} = 15$ GHz. The current density induced on the almond is discretized using $N_s = 507{,}156$ spatial basis functions. The PWTD ray data is compressed using LCBs with $M^{min} = 16$ and accuracy set to 10^{-3}. The simulation is performed $N_t = 1300$ time steps with $\Delta t = 1\,\text{ps}$ on the Sandy Bridge compute-node.

Three incident fields with different polarizations and directions are tested, i.e., $(\hat{\mathbf{k}} = \hat{\mathbf{x}}, \hat{\mathbf{p}} = \hat{\mathbf{z}})$, $(\hat{\mathbf{k}} = \hat{\mathbf{x}}, \hat{\mathbf{p}} = \hat{\mathbf{y}})$ and $(\hat{\mathbf{k}} = \hat{\mathbf{z}}, \hat{\mathbf{p}} = \hat{\mathbf{y}})$. The snapshot of the current density on the almond due to each incidence field is plotted [Figure 5.6]. Table 5.1 presents the memory costs of the ray data and computational costs of the translation and spherical interpolation stages. These results are compared with the reference data obtained by the classical PWTD-TD-CFIE solver with $(\hat{\mathbf{k}} = \hat{\mathbf{x}}, \hat{\mathbf{p}} = \hat{\mathbf{z}})$. In this example, up to 4.4-fold memory reduction and 1.5-fold speedup can be achieved.

Next, the almond is illuminated by $\mathbf{E}^i(\mathbf{r},t)$ in (5.24) with $f_0 = 160\,\text{GHz}$, $f_{bw} = 60\,\text{GHz}$, $\hat{\mathbf{p}} = \hat{\mathbf{y}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{z}}$. The current density induced on the almond is discretized using $N_s = 5{,}371{,}092$ spatial basis functions. The PWTD ray data is compressed using LCBs with $M^{min} = 16$ and accuracy set to 10^{-3}. The simulation is performed $N_t = 960$ time steps with $\Delta t = 0.25\,\text{ps}$ and run on the Sandy Bridge cluster. An eleven-level PWTD tree is constructed upon seting the edge length of box at the finest level to $0.488\,\lambda$ and $\gamma = 4$.

Table 5.2 presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires around 988 GB of memory and 11 days of CPU time when 32 processors are used. Note that the memory cost for the ray data is reduced from 1.95 TB to 454 GB by leveraging the LCB-based compression. The bistatic RCS of the almond is computed at 140 GHz, 160 GHz and 180 GHz and compared to those obtained using a frequency domain combined field integral equation (FD-CFIE) solver [Figure 5.8]; results are in good agreement. In addition, snapshots of the current induced on the almond at times 300 Δt, 440 Δt, 460 Δt, and 560 Δt reveal physical optics-induced current on the light side of the almond and edge diffracted current on the shadow side [Figure 5.9].

	Ray data (GB)	Translation (h)	Interpolation (h)
Reference	36.8	50.5	12.6
$\hat{\mathbf{k}}=\hat{\mathbf{x}},\hat{\mathbf{p}}=\hat{\mathbf{z}}$	8.42	26.1	15.5
$\hat{\mathbf{k}}=\hat{\mathbf{x}},\hat{\mathbf{p}}=\hat{\mathbf{y}}$	10.6	32.1	18.4
$\hat{\mathbf{k}}=\hat{\mathbf{z}},\hat{\mathbf{p}}=\hat{\mathbf{y}}$	12.1	32.8	20.7

Table 5.1 Memory cost of the ray data and computational costs of the translation and spherical interpolation stages for the problem involving the NASA almond with $N_s = 507{,}156$ spatial basis functions

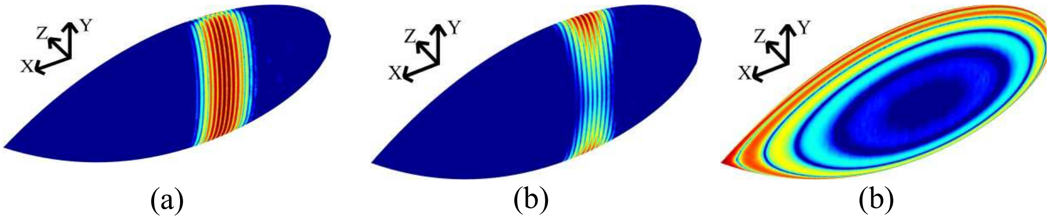


Figure 5.7 Snapshots of the current density induced on the NASA almond with $N_s = 507{,}156$ spatial basis functions obtained by the LCB-enhanced PWTD-TD-CFIE solver with (a) $\hat{\mathbf{k}}=\hat{\mathbf{x}},\hat{\mathbf{p}}=\hat{\mathbf{z}}$ at $t = 520\ \Delta t$, (b) $\hat{\mathbf{k}}=\hat{\mathbf{x}},\hat{\mathbf{p}}=\hat{\mathbf{y}}$ at $t = 520\ \Delta t$, (c) $\hat{\mathbf{k}}=\hat{\mathbf{z}},\hat{\mathbf{p}}=\hat{\mathbf{y}}$ at $t = 600\ \Delta t$.

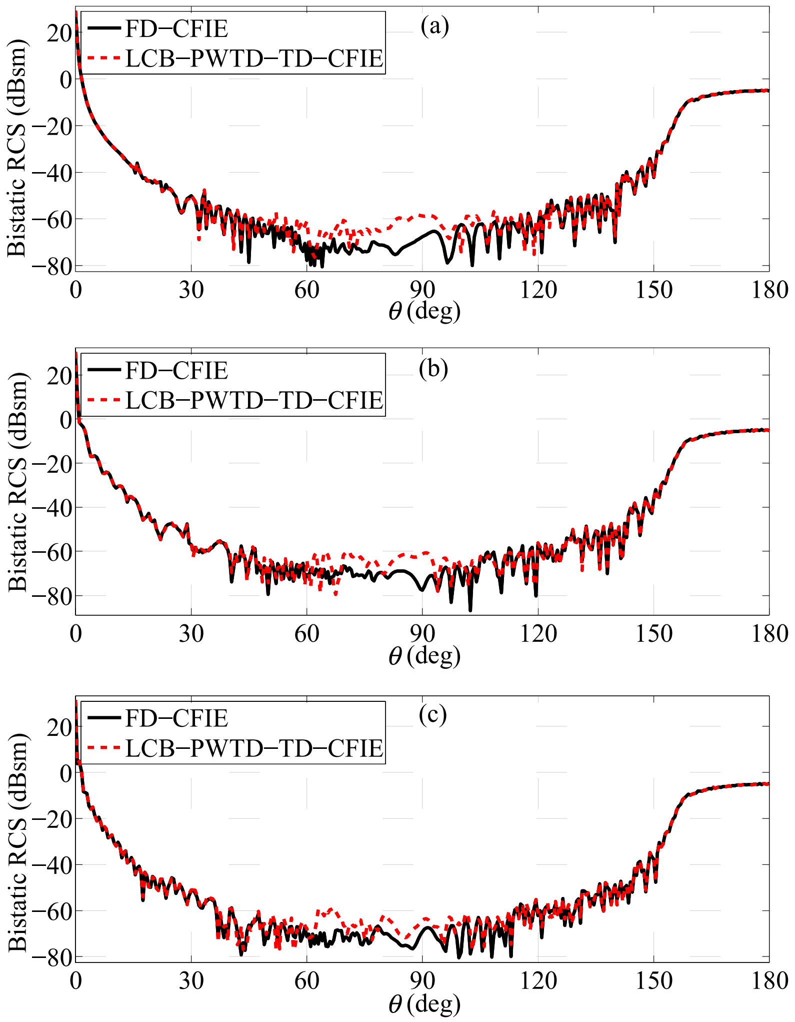
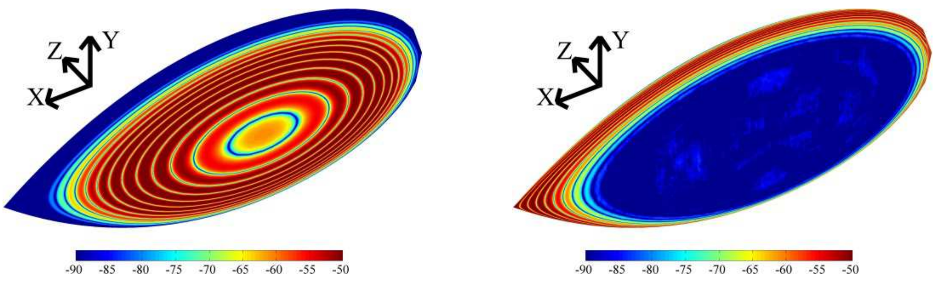
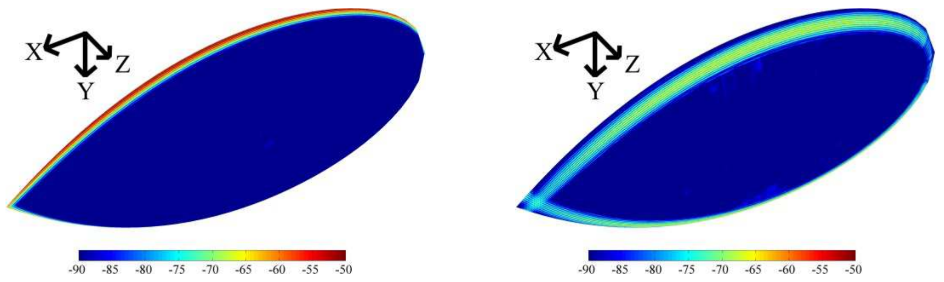


Figure 5.8 Bistatic RCS of the NASA almond discretized with $N_s = 5{,}371{,}092$ spatial basis functions at (a) 140 GHz (b) 160 GHz and (c) 180 GHz computed at $\phi = 0^\circ$ and $\theta = [0,180]^\circ$ by LCB-enhanced PWTD-TD-CFIE solver and FD-CFIE solver.



(a) (b)



(c) (d)

Figure 5.9 Snapshots of the current density (in dB) induced on the NASA almond with $N_s = 5{,}371{,}092$ spatial basis functions obtained by the LCB-enhanced PWTD-TD-CFIE solver at (a) $t = 300\ \Delta t$, (b) $t = 440\ \Delta t$, (c) $t = 460\ \Delta t$, (d) $t = 560\ \Delta t$

	NASA almond	Airbus A-320
Maximum dimension	0.25 m (185.26 λ)	37.5 m (123.46 λ)
Frequency (f_0, f_{bw})	$(160, 60)$ GHz	$(740, 245)$ MHz
Number of unknowns N_s	5,371,092	4,086,129
Time step size Δt	0.25 ps	50 ps
Number of time steps N_t	960	1,040
Number of processors N_p	32	40
Memory (near field)	534 GB	775.7 GB
Memory (PWTD)	$1951 \rightarrow 454$ GB	$601 \rightarrow 158$ GB
Setup time	3.2 h	8.7 h
RHS time (near field)	2 h	2.7 h
RHS time (PWTD)	10.5 days	41 h
LHS time (GMRES)	4.6 h	42 min
RCS calculation time	18.8 h	768.5 s

Table 5.2 The technical data for the setups and solutions of scattering problems involving real-life targets.

5.4.4 Real-Life Target

Finally, the LCB-enhanced PWTD-TD-CFIE solver is applied to the analysis of transient scattering from an Airbus-A320 model, which can be enclosed by a fictitious 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 (5.24) with $f_0 = 740$ MHz, $f_{bw} = 245$ MHz, $\hat{\mathbf{p}} = \hat{\mathbf{z}}$, and $\hat{\mathbf{k}} = \hat{\mathbf{y}}$. The current induced on the airplane is discretized using $N_s = 4{,}086{,}129$ spatial basis functions. The PWTD ray data is represented using LCBs with $M^{min} = 16$ and accuracy set to 10^{-3}. The simulation is executed for $N_t = 1040$ time steps with $\Delta t = 50$ ps and run on the Sandy Bridge cluster. A ten-level PWTD tree is constructed upon setting the side length of boxes at the finest level to $0.467\ \lambda$ and $\gamma = 4$.

Table 5.2 presents the solver parameters and CPU and memory requirements of the solver's different stages. The solver requires around 933.7 GB of memory and 53.4 hours of CPU time when 40 processors are used. Note that the memory cost for the ray data is reduced from 601 GB to 158 GB by leveraging the LCB-based compression. The broadband RCS of the airplane along the +*z* direction ($\theta = 0$) is computed using the LCB-enhanced PWTD-TD-CFIE solver and the FD-CFIE solver [Figure 5.10]; the results agree well. In addition, the current densities induced at two points selected on the engine intake and the tail are computed [Figure 5.11]. Finally, snapshots of the current induced on the airplane at times $380\ \Delta t$, $480\ \Delta t$, and $640\ \Delta t$ are shown in Fig. 12.

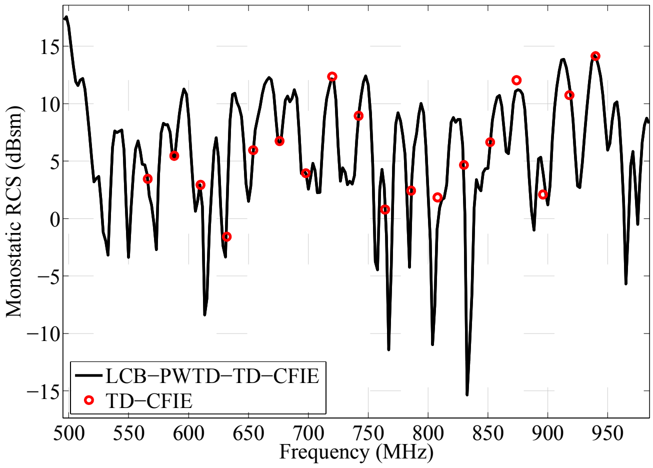


Figure 5.10 Broadband RCS of the Airbus A-320 model along the +*z* direction computed by LCB-enhanced PWTD-TD-CFIE solver and FD-CFIE solver.

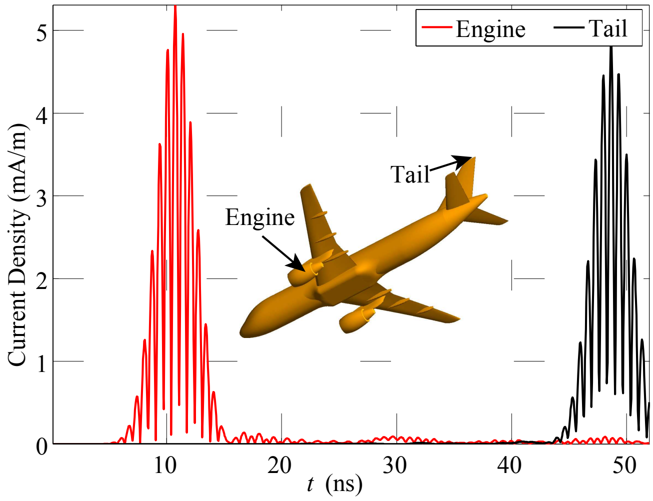
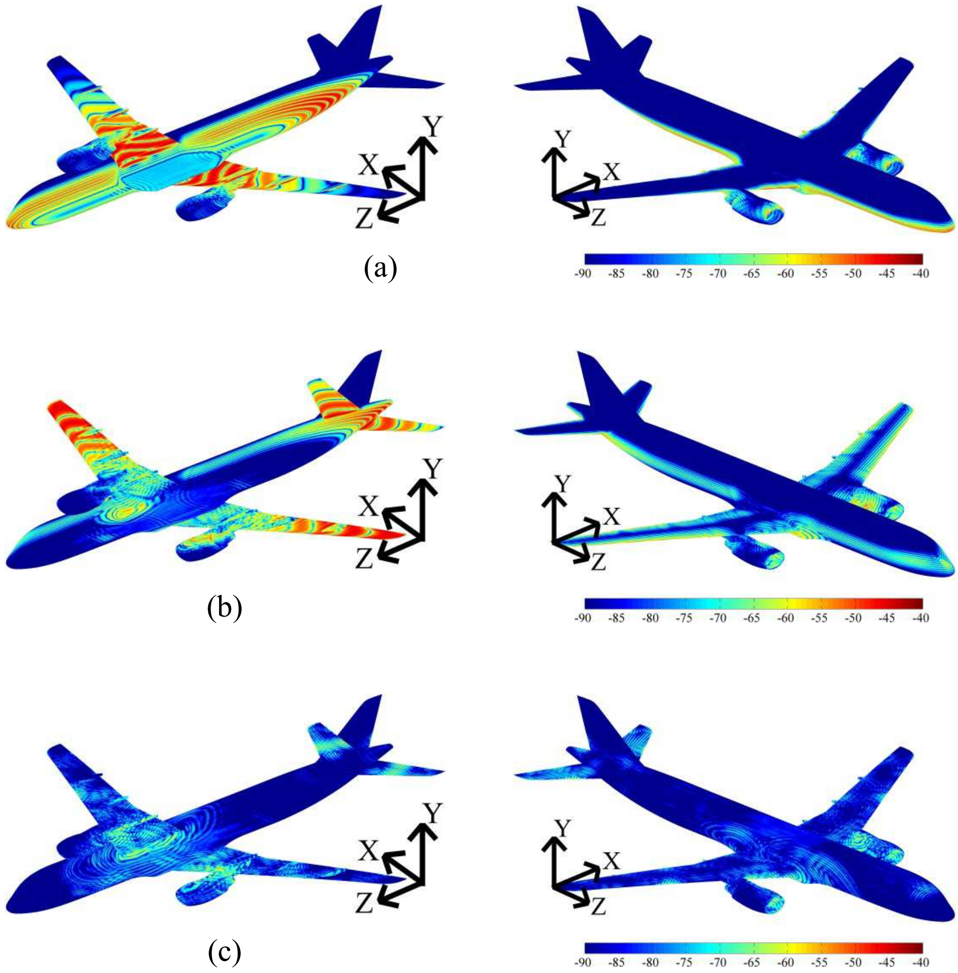


Figure 5.11 Magnitudes of the current density at the engine intake and tail of the Airbus A-320 model computed by the LCB-enhanced PWTD-TD-CFIE solver



(a)

(b)

(c)

Figure 5.12 Snapshots of the current density (in dB) induced on the Airbus A-320 model obtained by the LCB-enhanced PWTD-TD-CFIE solver at (a) $t = 380\ \Delta t$, (b) $t = 480\ \Delta t$, (c) $t = 640\ \Delta t$.

5.5 Chapter Conclusion

This chapter presents a local cosine wavelet-based compression scheme to reduce the memory and computational costs of the multilevel PWTD-accelerated MOT-based TD-SIE solver. The proposed scheme compresses the PWTD ray data along the temporal dimension using LCBs and carries out the PWTD operations including translation, shifting, and spherical interpolation/filtering in the wavelet domain. The resulting LCB-enhanced PWTD-TD-SIE solver yields reduced memory and computational costs compared to the conventional PWTD-TD-SIE solver. Indeed, when applied to the analysis of transient scattering from smooth quasi-planner objects under high frequency excitations with near-normal incidence angle, the memory cost of the solver scales as $O(N_s \log N_s)$ and the computational cost scales nearly as $O(N_s^{1.5})$. These scaling estimates are theoretically proved and validated by numerical examples. This LCB-enhanced PWTD-TD-SIE solver is successfully applied to the analysis of transient scattering from canonical and real-life objects measuring well over one hundred wavelengths in size.

CHAPTER 6
An Explicit and PWTD-Accelerated TD Electric Field VIE Solver for Analyzing Transient Scattering from Electrically Large Dielectric Objects

6.1 Chapter Introduction

Transient analysis of electromagnetic scattering from large inhomogeneous dielectric objects has various engineering and scientific applications ranging from the design of optoelectronic devices and broadband antenna radomes to the study of blood cell aggregations. Among simulators capable of electromagnetic characterization of such objects, MOT-based time domain electric field volume integral equation (TD-EFVIE) solvers are rapidly gaining ground [19]. The TD-EFVIE is constructed by enforcing that the total electric field is equal to the incident electric field plus the scattered electric field due to electric flux density induced throughout the scatter. To numerically solve the TD-EFVIE, the unknown electric flux density is expanded using spatio-temporal basis functions. Inserting this expansion into the TD-EFVIE and testing the resulting equation in space and time yield a set of linear systems that can be solved by the MOT scheme.

The MOT scheme can be implicit or explicit, depending on the types of the spatio-temporal basis functions. The implicit MOT, oftentimes using Schaubert-Wilton-Glisson spatial basis functions [155] and piecewise polynomial temporal basis functions [12], requires implicitly solving the linear system every time step [19-21, 156]. In contrast, the explicit MOT scheme, usually leverages pulse spatial basis functions and low order temporal basis functions. These simple basis functions render the explicit MOT scheme computationally more efficient, yet less stable than their implicit counterpart [51, 55, 56, 61]. Recently, a stable explicit MOT-TD-EFVIE solver that updates the flux density using a predictor-corrector (PC) scheme is proposed [157]. Moreover, CPU parallelized [158] and graphic processing unit accelerated [159] implementations are developed to further advance the capability of the solver. That said, the real-life applicability of this solver is still limited by its computational complexity, i.e., $O(N_s^2 N_t)$ where N_s is the number of spatial basis functions and N_t is the number of simulation time steps. To overcome this computational bottleneck, fast algorithms such as multilevel PWTD algorithm [8] and TD-AIM [9] must be considered.

In this work, a PWTD-accelerated PC-based MOT-TD-EFVIE solver is developed. In the past, PWTD has been successfully applied to the acceleration of MOT-based TD-SIE and TD-VIE solvers [13, 15, 20, 21]. When used in tandem with TD-EFVIE solvers, it reduces the abovementioned computational complexity to $O(N_s^2 N_t)$ [20, 21]. Previously, various PWTD-accelerated implicit MOT-TD-EFVIE solvers were developed wherein the PWTD scheme permits fast computation of electric fields due to past flux density [20, 21]. However, in the proposed explicit solver, the electric fields are computed from vector potentials using finite difference. Hence the PWTD scheme is used to accelerate computation of the vector potentials due to the flux density. Specifically, in the predictor step, the interactions between far field box pairs are evaluated by decomposing vector potentials into their three Cartesian components that are propagated independently using a scalar field PWTD scheme; in the corrector step, the electric fields are updated using a time-dependent averaging factor that improves the accuracy while maintaining the stability as opposed to the constant averaging factor used in [157]. The parallelization strategy developed in Chapter 2 is also integrated into this solver. The efficiency and accuracy of this parallel PWTD-PC-EFVIE solver are demonstrated via its application to transient scattering from canonical objects involving 25 million spatial unknowns. Furthermore, the solver is applied to characterization of light interaction with red blood cells involving 12 million spatial unknowns.

The rest of this Chapter is organized as follows. Section 6.2.1 explains the PC-based MOT-TD-EFVIE solution. Section 6.2.2 delineates the PWTD scheme used to accelerate the MOT solution. Section 6.2.3 describes the parallelization of the PWTD-

PC-EFVIE solver. Numerical examples that demonstrate the efficiency, accuracy and applicability of the proposed solver are presented in Section 6.3, followed by the concluding remarks in Section 6.4.

6.2 Formulation

6.2.1 PC-Based TD-EFVIE Solution

Let V denote the support of inhomogeneous dielectric scatterers that reside in an unbounded background medium with permittivity ε_0. It's assumed that the scatterers are isotropic, non-magnetic, non-dispersive, lossless and have permittivity $\varepsilon(\mathbf{r})$. Let $\mathbf{E}^i(\mathbf{r},t)$ denote an incident electric field that is bandlimited to maximum frequency f_{max}; it's assumed that $\mathbf{E}^i(\mathbf{r},t)=0,\ \forall\mathbf{r}\in V$ for $t<0$. This incident field induces, in the scatterers, an electric flux density $\varepsilon(\mathbf{r})\mathbf{E}(\mathbf{r},t)$ which in turn generates a scattered electric field $\mathbf{E}^s(\mathbf{r},t)$. Here $\mathbf{E}(\mathbf{r},t)$ is the total electric field. A TD-EFVIE can be formulated by decomposing the total field $\mathbf{E}(\mathbf{r},t)$ into the incident field $\mathbf{E}^i(\mathbf{r},t)$ and the scattered field $\mathbf{E}^s(\mathbf{r},t)$ as

$$\begin{aligned}\mathbf{E}(\mathbf{r},t) &= \mathbf{E}^i(\mathbf{r},t)+\mathbf{E}^s(\mathbf{r},t)\\ &= \mathbf{E}^i(\mathbf{r},t)+\nabla\nabla\cdot\mathbf{P}(\mathbf{r},t)-\frac{\partial_t^2}{c_0^2}\mathbf{P}(\mathbf{r},t).\end{aligned} \tag{6.1}$$

Here, c_0 is the speed of light in the background medium, ∂_t denotes time derivative and $\mathbf{P}(\mathbf{r},t)$ represents the *modified* magnetic vector potential due to the induced electric flux density that is expressed as

$$\mathbf{P}(\mathbf{r},t)=\int_V d\mathbf{r}'\frac{\left(\varepsilon(\mathbf{r}')-\varepsilon_0\right)\mathbf{E}(\mathbf{r}',t-R/c_0)}{4\pi\varepsilon_0 R}. \tag{6.2}$$

Here $R=|\mathbf{r}-\mathbf{r}'|$ is the distance between source point $\mathbf{r}'$ and observer point $\mathbf{r}$.

To numerically solve (6.1), the computation domain is discretized using cubic elements with edge length Δs. Here Δs needs to resolve the minimum wavelength inside the scatterers. It's assumed that the permittivity inside the n^{th} element is constant, i.e., $\varepsilon(\mathbf{r}) = \varepsilon(\mathbf{r}_n)$ where $\mathbf{r}_n$ is the center of the n^{th} element. Next, the total electric field in the scatterers, $\mathbf{E}(\mathbf{r},t)$, is discretized using spatial basis functions $S_n(\mathbf{r})$, $n = 1,...,N_s$ and temporal basis functions $T_j(t)$, $j = 1,...,N_t$, as

$$\mathbf{E}(\mathbf{r},t) = \sum_{n=1}^{N_s} \sum_{j=1}^{N_t} \mathbf{E}_{n,j} S_n(\mathbf{r}) T_j(t). \tag{6.3}$$

Here, $\mathbf{E}_{n,j}$ is the vector (with three Cartesian components) expansion coefficient associated with spatio-temporal basis function $S_n(\mathbf{r})T_j(t)$. The spatial basis function $S_n(\mathbf{r})$ is the pulse basis function defined in the n^{th} cubic element, i.e., $S_n(\mathbf{r}) = 1,\ |\mathbf{r} - \mathbf{r}_n|_\infty \le \Delta s / 2$, where $|\cdot|_\infty$ denotes the Linfinity norm. The temporal basis function $T_j(t) = T(t - j\Delta t)$ is the shifted Lagrange polynomial [12]. Here, the time step size Δt is chosen by a Courant-Friedrichs-Lewy (CFL)-type condition, as $c_0 \Delta t \le \Delta s \le 2c_0 \Delta t$. With such choice of spatio-temporal basis function, the expansion coefficient $\mathbf{E}_{n,j}$ simply becomes the electric field sampled at the n^{th} element center and time step j, i.e., $\mathbf{E}_{n,j} = \mathbf{E}(\mathbf{r}_n, t_j)$. To compute $\mathbf{E}_{n,j}$ via explicit MOT, equation (6.1) is enforced at space-time $(\mathbf{r}_m, t_i)$, $m = 1,...,N_s$, $i = 1,...,N_t$, and the spatial operator $\nabla\nabla\cdot$ and temporal operator ∂_t^2 / c_0^2 in (6.1) are approximated by finite difference. Note that the finite difference schemes require computation of vector potential $\mathbf{P}(\mathbf{r},t)$ at times t_{i-1}, t_i, t_{i+1}, and at element centers $\mathbf{r}_m$, $m = 1,...,N_s'$. Here, $N_s' > N_s$ denotes the number of elements that have adjacent elements (including themselves) residing in the scatterers. Note: the element indices are sorted such that the first N_s elements correspond to those in the scatterers and the following $N_s' - N_s$ elements correspond to those in the background medium. Upon substituting (6.3) into (6.2), the vector potential $\mathbf{P}(\mathbf{r}_m, t_i)$, $m = 1,...,N_s'$ is computed as

$$\mathbf{P}(\mathbf{r}_m, t_i) = \mathbf{P}_s(\mathbf{r}_m, t_i) + \mathbf{P}_c(\mathbf{r}_m, t_i) + \mathbf{P}_{nc}(\mathbf{r}_m, t_i)\ . \tag{6.4}$$

Here, $\mathbf{P}_s(\mathbf{r}_m,t_i)=(\varepsilon(\mathbf{r}_m)-\varepsilon_b)\beta^s\mathbf{E}(\mathbf{r}_m,t_i)$ is dubbed the "self-term" contribution, $\beta^s=\int_{V_m} d\mathbf{r}S_m(\mathbf{r})/(4\pi\varepsilon_0\,|\mathbf{r}-\mathbf{r}_m|)$ and V_m denotes the support of $S_m(\mathbf{r})$. Note: β^s is a constant independent of m and can be computed analytically [160]. The second term $\mathbf{P}_c(\mathbf{r}_m,t_i)$ and third term $\mathbf{P}_{nc}(\mathbf{r}_m,t_i)$ on the RHS of (6.4) are computed by

$$\mathbf{P}_r(\mathbf{r}_m,t_i)=\sum_{n\in r_m}\sum_{j\in I_{imn}}\frac{w_n\mathbf{E}_{n,j}T_{j-i}(-R_{mn}/c_0)}{R_{mn}} \tag{6.5}$$

Here, $n\neq m$, $r\in\{c,nc\}$, $w_n=\Delta s^3(\varepsilon(\mathbf{r}_n)-\varepsilon_0)/(4\pi\varepsilon_0)$, $R_{mn}=|\mathbf{r}_m-\mathbf{r}_n|$, $I_{imn}=\{i-\lfloor R_{mn}/c_0\Delta t\rfloor,\ldots,i-\lfloor R_{mn}/c_0\Delta t\rfloor-p\}$ and p denotes the order of the temporal basis function. Note: in order to derive (6.5), the single point quadrature rule is used for the integration in (6.2). In (6.5), c_m and nc_m denote the "causal" and "non-causal" set of source points for observation point $\mathbf{r}_m$. Specifically, $c_m=\{n:|\mathbf{r}_m-\mathbf{r}_n|\geq 2c_0\Delta t\}$ and the computation of $\mathbf{P}_c(\mathbf{r}_m,t_{i+1})$ does not require the field coefficients $\mathbf{E}_{n,j}$ at time steps $j\geq i$. In contrast, $nc_m=\{n:|\mathbf{r}_m-\mathbf{r}_n|<2c_0\Delta t, m\neq n\}$ and the computation of $\mathbf{P}_{nc}(\mathbf{r}_m,t_{i+1})$ requires the knowledge of $\mathbf{E}_{n,i}$, which is not yet available. For this reason, $\mathbf{P}_c$ and $\mathbf{P}_{nc}$ represent the "causal" and "non-causal" contributions, respectively. As an example, a H-shaped structure is discretized by cubic elements with their element centers represented by small dots [Figure 6.1]. Consider an observer element m, m resides inside or outside the scatterer if $m<N_s$ or $N_s<m<N_s'$. The self element and causal/non-causal set of source elements are plotted in Figure 6.1. Under the CFL condition, the size of the non-causal sets is relatively small.

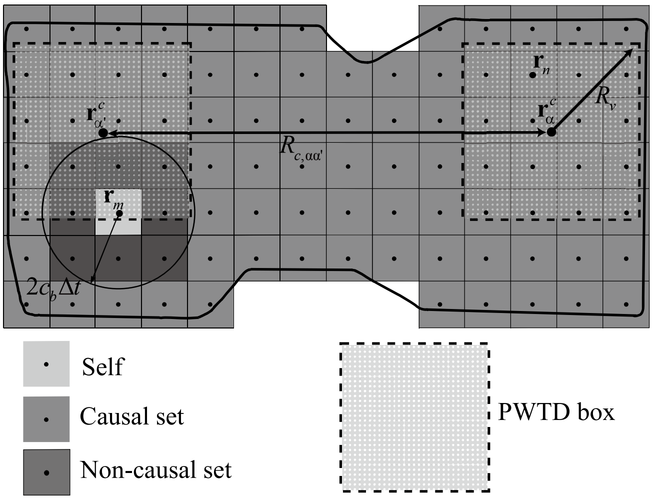


Figure 6.1 A H-shaped structure discretized using source cubic elements. One far field box pair is shown in the figure.

Next, the proposed explicit MOT scheme for solving (6.1) is elucidated. At each time step i, the electric field sampled at the m^{th} element center, $\mathbf{E}(\mathbf{r}_m,t_i)$, is computed by a predictor-corrector scheme. The predictor step estimates a total electric field throughout the scatterers, $\mathbf{E}^{pre}(\mathbf{r}_m,t_i)$, without knowledge of the unknown quantity $\mathbf{P}_{nc}(\mathbf{r}_m,t_{i+1})$, by

$$\begin{aligned}\mathbf{E}^{pre}(\mathbf{r}_m,t_i) = \beta_m\{&\mathbf{E}^{i}(\mathbf{r}_m,t_i) - \tilde{\delta}^{b}[\mathbf{P}_s](\mathbf{r}_m,t_i)\\ &-\delta^{c}[\mathbf{P}_c](\mathbf{r}_m,t_i) - \delta^{b}[\mathbf{P}_{nc}](\mathbf{r}_m,t_i)\\ &+\mathcal{G}[\mathbf{P}_c+\mathbf{P}_{nc}](\mathbf{r}_m,t_i) + \mathcal{G}[\mathbf{P}_s](\mathbf{r}_m,t_{i-1})\}\end{aligned} \tag{6.6}$$

where $\beta_m = 1/(1+2\beta^{s}(\varepsilon(\mathbf{r}_m)-\varepsilon_0)/(c_0^2\Delta t^2))$, $\delta^{c}[\cdot]$ and $\delta^{b}[\cdot]$ are the second order central and backward difference approximations of the temporal operator ∂_t^2/c_0^2. Specifically, let $f(t_i)=\mathbf{P}_r(\mathbf{r}_m,t_i)$, $r\in\{s,\ c,\ nc\}$, these finite difference formulas are

$$\begin{aligned}&\delta^{c}[f](t_i)=(f(t_{i+1})-2f(t_i)+f(t_{i-1}))/(c_0^2\Delta t^2)\\ &\delta^{b}[f](t_i)=(2f(t_i)-5f(t_{i-1})+4f(t_{i-2})-f(t_{i-3}))/(c_0^2\Delta t^2),\end{aligned} \tag{6.7}$$

and $\tilde{\delta}^{b}[f](t_i)=\delta^{b}[f](t_i)-2f(t_i)/(c_0^2\Delta t^2)$. In (6.6), $\mathcal{G}[\cdot]$ is the first order central difference approximation of the spatial operator $\nabla\nabla\cdot$, which can be found in Appendix of [157].

The corrector step uses $\mathbf{E}^{pre}(\mathbf{r}_m,t_i)$ to update the self-term and non-causal potential contributions that, in turn, are used to correct the total electric field. Specifically, the self-term contribution can be computed as $\mathbf{P}_s^{pre}(\mathbf{r}_m,t_i)=\beta^s(\varepsilon(\mathbf{r}_m)-\varepsilon_0)\mathbf{E}^{pre}(\mathbf{r}_m,t_i)$; the non-causal contribution at time step $i+1$, $\mathbf{P}_{nc}^{pre}(\mathbf{r}_m,t_{i+1})$, is computed using $\mathbf{E}^{pre}(\mathbf{r}_m,t_i)$ via (6.5). The (corrected) electric field, $\mathbf{E}(\mathbf{r}_m,t_i)$, can be computed as

$$\begin{aligned}\mathbf{E}(\mathbf{r}_m,t_i)=\beta_m\{&\mathbf{E}^i(\mathbf{r}_m,t_i)-\tilde{\delta}^b[\mathbf{P}_s^{pre}](\mathbf{r}_m,t_i)\\&-\omega_{mi}\delta^c[\mathbf{P}_c+\mathbf{P}_{nc}^{pre}](\mathbf{r}_m,t_i)\\&-(1-\omega_{mi})\delta^c[\mathbf{P}_c+\mathbf{P}_{nc}](\mathbf{r}_m,t_{i-1})\\&+\omega_{mi}\mathcal{G}[\mathbf{P}_s^{pre}+\mathbf{P}_c+\mathbf{P}_{nc}^{pre}](\mathbf{r}_m,t_i)\\&+(1-\omega_{mi})\mathcal{G}[\mathbf{P}_s+\mathbf{P}_c+\mathbf{P}_{nc}](\mathbf{r}_m,t_{i-1})\}.\end{aligned}\tag{6.8}$$

Note that the central difference $\delta^c[\cdot]$ is now applied to both the causal and non-causal contributions. In (6.8), ω_{mi} denotes an averaging factor that stabilizes the corrector step. Throughout this paper, it's assumed that the incident electric field is a plane wave propagating in direction $\hat{\mathbf{k}}$. In this case, ω_{mi} can be chosen as

$$\omega_{mi}=1+\frac{1}{2}\cos\left(\frac{\pi(t_i-\hat{\mathbf{k}}\cdot\mathbf{r}_m/c_0-\tau_1)}{2(\tau_2-\tau_1)}\right),\quad \frac{\hat{\mathbf{k}}\cdot\mathbf{r}_m}{c_0}+\tau_1\le t_i\le\frac{\hat{\mathbf{k}}\cdot\mathbf{r}_m}{c_0}+\tau_2 \tag{6.9}$$

and $\omega_{mi}=1$ for $t_i<\hat{\mathbf{k}}\cdot\mathbf{r}_m/c_0+\tau_1$, $\omega_{mi}=0.5$ for $t_i>\hat{\mathbf{k}}\cdot\mathbf{r}_m/c_0+\tau_2$. Here, $\tau_2>\tau_1>0$ are parameters related to the bandwidth of the plane wave. Note: larger ω_{mi} leads to better accuracy, yet worse stability, and $\omega_{mi}=1$ indicates that no averaging is applied. Apparently, the averaging factor varies smoothly as a function of m and i, and stabilizes the MOT solution in the "late time". Once $\mathbf{E}(\mathbf{r}_m,t_i)$ is computed, the vector potential contributions $\mathbf{P}_s(\mathbf{r}_m,t_i)$ and $\mathbf{P}_{nc}(\mathbf{r}_m,t_{i+1})$ can be updated, and the MOT scheme moves to the next time step $i+1$.

In the above-described PC-based MOT scheme, the memory and computational costs depend on those associated with computation of the electric fields via finite difference in (6.6)/(6.8) and computation of the vector potentials by (6.5). The former requires only space-time localized operations and is computationally inexpensive. The latter, in contrast, involves interactions between N_s' observer points and N_s source points for N_t time steps, and requires storage of $\mathbf{E}_{n,j}$ at N_s source points for $O(D_{\max}/c_0\Delta t) = O(N_s^{1/3})$ consecutive time steps. Here $D_{\max}$ is the maximum distance between source and observer points. As a result, the computational and memory costs of the latter (and the solver) scale as $O(N_t N_s^2)$ and $O(N_s^{4/3})$, respectively. These costs are prohibitively high when the solver is applied to the analysis of transient phenomena involving electrically large objects. Next, a PWTD-accelerated PC-based MOT scheme that requires only $O(N_t N_s)$ CPU and $O(N_s \log N_s)$ memory resources is explained.

6.2.2 PWTD Acceleration

The PWTD algorithm described in the section permits fast evaluation of the vector potential $\mathbf{P}_r(\mathbf{r}_m, t_{i+1})$ in (6.5), rather than evaluation of the electric fields due to the flux density as in the PWTD-accelerated implicit MOT-based TD-EFVIE solvers [20, 21]. In what follows, the PWTD algorithm is briefly summarized while only those details pertinent to the aforementioned differences are expounded.

First, a fictitious box enclosing the N_s' elements is recursively subdivided into eight smaller boxes until the edge length of the smallest boxes reaches a prescribed portion of the minimum wavelength at the maximum frequency, $\lambda = c_0 / f_{max}$. An element n is said to reside in a box if its center $\mathbf{r}_n$ locates inside that box; empty boxes are immediately discarded. This procedure constructs a hierarchical tree structure of $N_L = O\left(\log(N_s^{1/3})\right)$ levels. At each level v, $v = 1, \ldots, N_L$, there exist $N_g^v \approx 8^{N_L - v}$ nonempty boxes. The radius of a sphere enclosing a level v box is $R^v = 2^{(v-1)} R^1$ with $R^1 = O(1)$.

Next, starting from the coarsest level N_L, two boxes α and α' centered at $\mathbf{r}_\alpha^c$ and $\mathbf{r}_{\alpha'}^c$ are termed a far field pair if (i) their center distance $R_{c,\alpha\alpha'} = \left|\mathbf{R}_{c,\alpha\alpha'}\right| = \left|\mathbf{r}_{\alpha'}^c - \mathbf{r}_\alpha^c\right|$ satisfies the condition $R_{c,\alpha\alpha'} > \gamma R^v$, $(4 \le \gamma \le 6)$, and (ii) their parent boxes do not constitute a far field pair. Those box pairs at the finest level $v = 1$, which do not constitute far field pairs, are termed near field pairs; also each box at the finest level forms a near field pair with itself. The interactions between elements in near field pairs (self interaction excluded) –henceforth called near field calculation– are directly evaluated by (6.5), while those between far field pairs are handled by the PWTD scheme. When the finest level box size is properly chosen, all non-causal contribution $\mathbf{P}_{nc}(\mathbf{r}_m, t_{i+1})$ and partial causal contribution $\mathbf{P}_c(\mathbf{r}_m, t_{i+1})$ are handled by the near field calculation; the rest causal contribution $\mathbf{P}_c(\mathbf{r}_m, t_{i+1})$ is accounted for by PWTD. Consider a far field pair α and α' [Figure 6.1], the total electric field at the n^{th} source element in box α, $\mathbf{E}(\mathbf{r}_n, t)$, is first represented using an approximate prolate spheroidal (APS) function $T^{APS}(t)$ that is bandlimited to $f_s = \chi_t f_{max}$, where $\chi_t = 1/(2 f_{max} \Delta t)$ is the temporal oversampling factor, and approximately time-limited to $-p_f \Delta t < t < p_f \Delta t$, $5 \le p_f \le 10$ (see [42] regarding more details about the APS function). This allows splitting $\mathbf{E}(\mathbf{r}_n, t)$ into N_l^v consecutive bandlimited "sub-fields", $\mathbf{E}^l(\mathbf{r}_n, t)$, as

$$\mathbf{E}(\mathbf{r}_n, t) = \sum_{l}^{N_l^v} \mathbf{E}^l(\mathbf{r}_n, t) = \sum_{l}^{N_l^v} \sum_{j=(l-1)M^v+1}^{lM^v} \mathbf{E}(\mathbf{r}_n, t_j) T_j^{APS}(t). \tag{6.10}$$

Here, $N_l^v M^v = N_t$, M^v is chosen such that the duration of each sub-field, $T^v = (M^v + 2p_f)\Delta t$, satisfies that condition $T^v < (R_{c,\alpha\alpha'} - 2R^v)/c_0$, and $T_j^{APS}(t) = T^{APS}(t - j\Delta t)$ is the shifted APS function. Next, the potential (causal contribution) at element center $\mathbf{r}_m$ due to the l^{th} sub-fields of all source elements that reside in box α, $\mathbf{P}_c^l(\mathbf{r}_m, t)$, is computed in three stages. (i) A set of outgoing rays associated with box α is constructed by projecting the sub-field $\mathbf{E}^l(\mathbf{r}_n, t)$, $n \in \alpha$ in directions $\hat{\mathbf{k}}_{pq}^v$, as

$$\mathbf{G}_{l,\alpha}^{+}(\hat{\mathbf{k}}_{pq}^v, t) = \sum_{n \in \alpha} \delta[t + \hat{\mathbf{k}}_{pq}^v \cdot (\mathbf{r}_n - \mathbf{r}_\alpha^c) / c_0] * w_n \mathbf{E}^l(\mathbf{r}_n, t). \tag{6.11}$$

Here $*$ denotes temporal convolution, $\delta[\cdot]$ is the Dirac function, $p = 0,\ldots,K^v$, $q = -K^v,\ldots,K^v$, $K^v = \lfloor 4\pi\chi_s f_s R^v / c_0 \rfloor + 1$ is the number of spherical harmonics with χ_s being the spherical oversampling factor, and $N_k^v = (K^v+1)(2K^v+1)$ is total number of directions [8]. Note that the outgoing ray $\mathbf{G}_{l,\alpha}^{+}(\hat{\mathbf{k}}_{pq}^{v},t)$ has three Cartesian components that can be independently constructed from the Cartesian components of $\mathbf{E}^{l}(\mathbf{r}_n,t)$. (ii) The outgoing rays are translated into incoming rays associated with box α', $\mathbf{G}_{l,\alpha'}^{-}(\hat{\mathbf{k}}_{pq}^{v},t)$, by convolving $\mathbf{G}_{l,\alpha}^{+}(\hat{\mathbf{k}}_{pq}^{v},t)$ with the translation function $\mathcal{T}(\hat{\mathbf{k}}_{pq}^{v},t)$, as

$$\mathbf{G}_{l,\alpha'}^{-}(\hat{\mathbf{k}}_{pq}^{v},t) = \mathcal{T}(\hat{\mathbf{k}}_{pq}^{v},t) * \mathbf{G}_{l,\alpha}^{+}(\hat{\mathbf{k}}_{pq}^{v},t). \tag{6.12}$$

Here, the translation function is

$$\mathcal{T}(\hat{\mathbf{k}}_{pq}^{v},t) = \frac{-\partial_t}{4\pi R_{c,\alpha\alpha'}} \sum_{k=0}^{K^v} (2k+1)\Phi_k\left(\frac{c_0 t}{R_{c,\alpha\alpha'}}\right)\Phi_k\left(\frac{\hat{\mathbf{k}}_{pq}^{v}\cdot\mathbf{R}_{c,\alpha\alpha'}}{R_{c,\alpha\alpha'}}\right) \tag{6.13}$$

where $\Phi_k(\cdot)$ is the Legendre polynomial of degree k and $|t| \le R_{c,\alpha\alpha'}/c_0$. (iii) The causal contribution $\mathbf{P}_c^{l}(\mathbf{r}_m,t)$ are computed by projecting the incoming rays $\mathbf{G}_{l,\alpha'}^{-}(\hat{\mathbf{k}}_{pq}^{v},t)$ onto the observer element m and summing over all directions with weights ω_{pq} [8]

$$\mathbf{P}_c^{l}(\mathbf{r}_m,t) = \sum_{p=0}^{K^v}\sum_{q=-K^v}^{K^v} \omega_{pq}\delta[t - \hat{\mathbf{k}}_{pq}^{v}\cdot(\mathbf{r}_m - \mathbf{r}_{\alpha'}^{c})/c_0] * \mathbf{G}_{l,\alpha'}^{-}(\hat{\mathbf{k}}_{pq}^{v},t). \tag{6.14}$$

Note that in (6.11), (6.12) and (6.14), $\mathbf{P}_c^{l}(\mathbf{r}_m,t)$ is computed by decoupling its three Cartesian components and evaluating them independently be the scalar-field PWTD scheme. In practice, only outgoing/incoming rays of boxes at the finest level are constructed/projected directly from/onto the elements using (6.11)/(6.14), those at higher levels are computed by a scalar spherical interpolation/filtering scheme described in [132].

The computational and memory costs of the abovementioned PWTD-accelerated PC-based TD-EFVIE solver is briefly summarized here. Note that the analysis in [8] showed that computational costs of spherical interpolation/filtering and translation operations for one ray in one box along all directions scale as $O(T^v N_k^v \log T^v)$. As there exist N_l^v rays and N_g^v boxes at each level, the overall computational costs of the spherical interpolation/filtering and translation scale as $\sum_{v=1}^{N_L} N_k^v N_l^v N_g^v T^v \log T^v = O(N_s N_t)$. Meanwhile, the computational costs of the construction/projection of outgoing/incoming rays at the finest level, near field calculation and the evaluation of the electric field via finite difference, all scale as $O(N_s N_t)$. Therefore, the computational cost of the proposed PWTD-PC-EFVIE solver scales as $O(N_s N_t)$.

The memory cost of storing one outgoing/incoming ray in one box along all directions scales as $O(T^v N_k^v)$. As there are $O(1)$ rays in N_g^v boxes at each level that need to be stored, the memory cost for storing the ray data scales as $\sum_{v=1}^{N_L} N_g^v T^v N_k^v O(1) = O(N_s \log N_s)$. On the other hand, storing the electric fields $\mathbf{E}_{n,j}$ for evaluation of the potentials requires only $O(N_s)$ memory. Hence, the memory cost of the PWTD-PC-EFVIE solver scales as $O(N_s \log N_s)$.

6.2.3 Parallelization of the PWTD-PC-EFVIE Solver

Despite its attractive computational and memory costs estimates, the above-described PWTD-PC-EFVIE solver, if implemented using a serial CPU, still has limited applicability to electrically large transient scattering problems that oftentimes involve millions of spatial unknowns. Here, a highly scalable parallel implementation of the solver using distributed-memory CPU clusters is described. Note that the proposed solver has two computation phases in the predictor/corrector step, viz., the computation of vector potentials by the PWTD algorithm and the computation of electric fields by finite difference. In what follows, the parallelization of these phases are expounded.

First, efficient parallelization of the potential computation phase requires uniform partitioning of the memory and computation workloads associated with outgoing/incoming rays (i.e., ray data) in the PWTD algorithm. However, this task is nontrivial due to the PWTD algorithm's heterogeneous tree structure, as at each level the ray data is computed for $N_g^v = O(N_s / 8^v)$ boxes in spatial dimension, $N_k^v = O(4^v)$ samples in angular dimension, and $T^v = O(2^v)$ samples in temporal dimension; partitioning along a single dimension leads to poor load balance at either higher or lower PWTD tree levels [115, 121].

The proposed parallelization scheme leverage the partitioning strategy developed in Chapter 2 that distributes the loads hierarchically in more than one dimension depending on the number of boxes N_g^v at each level and the number of processors N_p. Let v_b denote the highest possible level at which $N_g^v \geq N_p$. At level $v \leq v_b$, each processor stores the ray data for all angular and temporal samples in approximately N_g^v / N_p boxes; at level $v > v_b$, each processor stores $N_k^v N_g^v / N_p$ angular samples and all temporal samples for one box. This parallelization strategy results in computation and memory load balancing and produces scalable communication patterns among processors at all levels of the PWTD tree [see Chapter 2]. The computation loads of the PWTD stages are partitioned as follows:

- *Construction/projection of outgoing/incoming rays at the finest level.* Each processor constructs outgoing rays by (6.11) for the source elements in its N_g^1 / N_p boxes; similarly, each processor computes the partial causal potential contribution (due to PWTD) by projecting the incoming rays in (6.14) onto the observer elements in its N_g^1 / N_p boxes.
- *Construction/projection of outgoing/incoming rays via spherical interpolation/filtering.* At level $v \leq v_b$, each processor spherically interpolates/filters the complete ray data for its N_g^v / N_p boxes; at level $v > v_b$, each processor spherically interpolates/filters the ray data for $O(T^v N_g^v / N_p)$ temporal samples and all angular samples of the ray data of one box. In other

words, the main computation workloads are split along the spatial and temporal dimensions.

- *Translation.* At level $v \leq v_b$, each processor performs translation in all directions for the N_g^v / N_p observations boxes it is responsible for; at level $v > v_b$, each processor carries out translation at $N_k^v N_g^v / N_p$ directions for the observer box it is in charge of. In other words, the computation workloads are distributed along the spatial and angular dimensions.
- *Near field calculation.* Each processor computes the noncausal potential contribution $\mathbf{P}_{nc}(\mathbf{r}_m, t_{i+1})$ and partial causal potential contribution $\mathbf{P}_c(\mathbf{r}_m, t_{i+1})$ for the observer elements in its N_g^1 / N_p boxes.

Next, the parallelization of the second phase is described. Note that the computation of electric field $\mathbf{E}(\mathbf{r}_m, t_i)$ at the m^{th} observer element in box α by the finite difference schemes in (6.6)/(6.8) requires the vector potential $\mathbf{P}_r(\mathbf{r}, t_i)$ and $\mathbf{P}_r(\mathbf{r}, t_{i-1})$ at its neighboring elements, which resides in box α and its adjacent boxes [Figure 6.1]. Therefore, the second computation phase can be easily parallelized using a partitioning strategy similar to that used in the near field calculation stage of the PWTD algorithm: each processor computes the electric fields at the observer elements in its N_g^v / N_p boxes.

6.3 Numerical Results

This section presents several numerical examples that demonstrate the efficiency, accuracy, and capability of the parallel PWTD-PC-EFVIE solver. In all examples considered here, the scatterers are excited by a plane wave with electric field given by

$$\mathbf{E}^i(\mathbf{r}, t) = \hat{\mathbf{p}} \cos[2\pi f_0 (t - \mathbf{r} \cdot \hat{\mathbf{k}} / c_0)] e^{-(t - \mathbf{r} \cdot \hat{\mathbf{k}} / c_0 - t_0)^2 / 2\sigma^2} \tag{6.15}$$

where f_0 is the central frequency, $\sigma = 3/(2\pi(f_{\max} - f_0))$, and $t_0 = 8\sigma$. The polarization direction $\hat{\mathbf{p}}$ and propagation direction $\hat{\mathbf{k}}$ of the plane wave are chosen as $\hat{\mathbf{p}} = \hat{\mathbf{x}}$ and $\hat{\mathbf{k}} = \hat{\mathbf{z}}$, respectively. All scatterers, unless otherwise stated, are centered at origin and located in free space (i.e., $\varepsilon_0 = 1$). 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 solver leverages a hybrid MPI and OpenMP parallelization strategy; one MPI process is launched per CPU and OpenMP uses four cores of each CPU.

6.3.1 Parallel efficiency

In this subsection, the parallel efficiencies of different stages of the PWTD-PC-EFVIE solver are investigated. Here, the parallel efficiency κ is measured as $\kappa = N_{ref} T_{N_{ref}} / N_p T_{N_p}$, where $T_{N_{ref}}$ is the reference execution time using N_{ref} processors and T_{N_p} is the execution time using N_p processors. Note: N_{ref} is chosen as the minimum number of processors required to perform a given simulation.

6.3.1.1 Cube

First, the PWTD-PC-EFVIE solver is applied to the analysis of transient scattering from a dielectric cube of edge length $4.7\ \mu\text{m}$ and permittivity $\varepsilon(\mathbf{r}) = 1.5$. The cube is illuminated by an incident electric field in (6.15) with $f_0 = 400\ \text{THz}$ and $f_{max} = 600\ \text{THz}$. The cube is discretized by $N_s = 804{,}357$ source elements and $N_s = 804{,}357$ observer elements upon setting $\Delta s = 0.05\ \mu\text{m}$. The simulation is performed $N_t = 240$ time steps using $\Delta t = 0.167\ \text{fs}$. A six-level PWTD tree is constructed upon setting the edge length of boxes at the finest level to $0.294\,\lambda$ and $\gamma = 6$.

The parallel efficiencies κ and computational costs of the predictor and corrector steps of the PWTD-PC-EFVIE solver are listed using $N_{ref} = 64$ and $N_p = 1024$ [Table Table 6.1]. As discussed in Section 6.2.2, the computational costs and parallel efficiencies of the predictor step dominates those of the solver as the computationally most demanding part, i.e., calculation of causal potential contribution $\mathbf{P}_c(\mathbf{r}_m, t_{i+1})$, is performed in this step. In this example, an efficiency of over 80% is achieved for the proposed solver. The parallel efficiencies κ of different PWTD stages for computation of $\mathbf{P}_{nc}(\mathbf{r}_m, t_{i+1})$ and $\mathbf{P}_c(\mathbf{r}_m, t_{i+1})$ in the predictor/corrector step are plotted with N_p varied from 64 to 1024 [Figure 6.2(a)]. Apparently, the parallel efficiency of the PWTD scheme is dominated by those of the translation and near field calculation stages.

6.3.1.2 Sphere

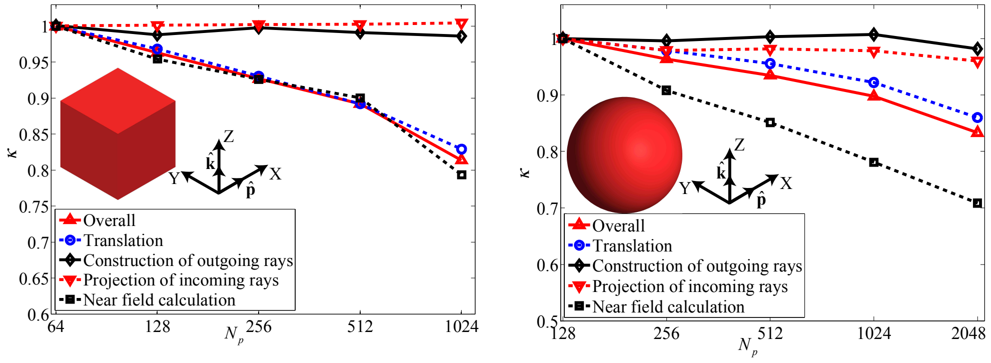
Next, the proposed solver is applied to the analysis of transient scattering from a dielectric sphere of radius 3 μm and permittivity $\varepsilon(\mathbf{r}) = 1.2$. The sphere is excited by an electric field in (6.15) with $f_0 = 400$ THz and $f_{max} = 600$ THz. The sphere is discretized with $N_s = 904{,}089$ source elements and $N_s' = 973{,}283$ observer elements upon setting $\Delta s = 0.025$ μm. The test is run $N_t = 100$ time steps with $\Delta t = 0.084$ fs. A seven-level PWTD tree is constructed upon setting the edge length of boxes at the finest level to $0.2\,\lambda$ and $\gamma = 6$.

The parallel efficiencies κ and computational costs of the predictor and corrector steps of the PWTD-PC-EFVIE solver are listed using $N_{ref} = 128$ and $N_p = 2048$ [Table Table 6.1]. Again, an efficiency of over 80% is achieved for the solver. The parallel efficiencies κ of the PWTD stages in the predictor/corrector step with N_p varied from 128 to 2048 are plotted in Figure 6.2(b).

Cube			
	Predictor	Corrector	Overall
$N_p = 64$	17.87 h	38.6 min	18.76 h
$N_p = 1024$	1.35 h	3.7 min	1.44 h
κ	82.7%	65.7%	81.7%
Sphere			
	Predictor	Corrector	Overall
$N_p = 128$	17.4 h	1.3 h	19 h
$N_p = 2048$	1.3 h	8.3 min	1.74 h

κ	83.6%	60%	80.5%

Table 6.1 Computational Costs and Parallel Efficiencies of the Predictor and Corrector Steps



(a)

(b)

Figure 6.2 Parallel efficiencies κ of different PWTD stages used in the predictor/corrector step for the examples involving (a) a dielectric cube and (b) a dielectric sphere.

6.3.2 Canonical Examples

In this subsection, the accuracy of the PWTD-PC-EFVIE solver is demonstrated by two canonical examples: a shell and a two-layer sphere. In both examples, the radar cross section (RCS) of the scatterers is computed using the discrete Fourier transform of the time domain data generated by the solver. In addition, the *x* component of scattered field, $E_x^s(\mathbf{r},t)$, is computed at several positions outside the scatterers. These results are compared with the Mie series solutions.

6.3.2.1 Shell

First, the proposed solver is first applied to the analysis of scattering from a dielectric shell with inner radius $2.55\ \mu\text{m}$, outer radius $2.7\ \mu\text{m}$ and permittivity $\varepsilon(\mathbf{r}) = 1.21$. The shell is excited by an electric field in (6.15) with $f_0 = 0$ THz and $f_{max} = 600$ THz. The sphere is discretized with $N_s = 481{,}906$ source elements and $N_s' = 770{,}828$ observer elements upon setting $\Delta s = 0.03\ \mu\text{m}$. The test is run $N_t = 800$ time steps with $\Delta t = 0.1$ fs. A six-level PWTD tree is constructed upon setting the edge length of boxes at the finest level to 0.341 λ and $\gamma = 5.5$. The averaging factor ω_{mi} is chosen by setting $\tau_1 = 3.2\ t_0$ and $\tau_2 = 3.5\ t_0$ in (6.9). The simulation requires 4.5 hours of CPU time using $N_p = 1024$.

The bistatic RCS of the shell at 299.4 THz obtained using the proposed solver is compared with the exact Mie series solution in Figure 6.3; excellent agreement is observed. Apparently, the maximum value of the RCS is achieved along the forward direction (i.e., $\theta = 0$). Moreover, the scattered fields $E_x^s(\mathbf{r},t)$ at Cartesian coordinates (0, $3\ \mu\text{m}$, 0), (0, 0, $3\ \mu\text{m}$) and (0, 0, $-3\ \mu\text{m}$) are computed [Figure 6.4]. The results agree well with the Mie series solutions. Note that these fields consist of contributions from wave refraction, reflection and diffractions.

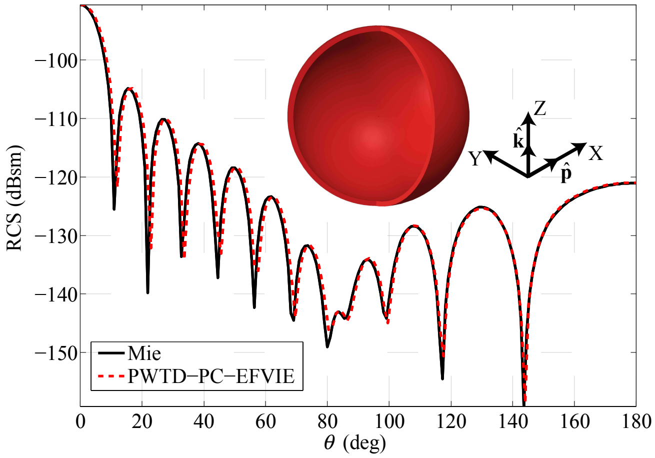
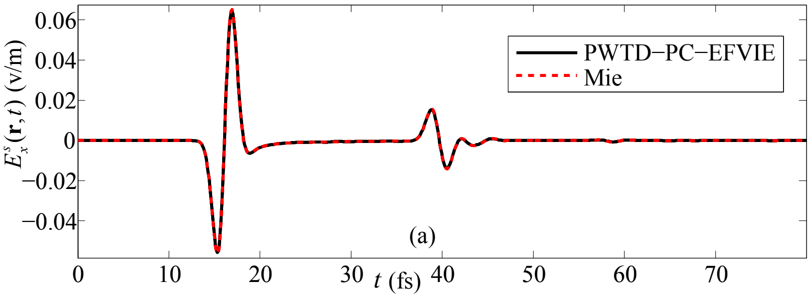


Figure 6.3 Bistatic RCS of the shell at 299.4 THz computed at $\phi = 0^\circ$ and $\theta = [0,180]^\circ$ via the PWTD-PC-EFVIE solver and Mie series solution.



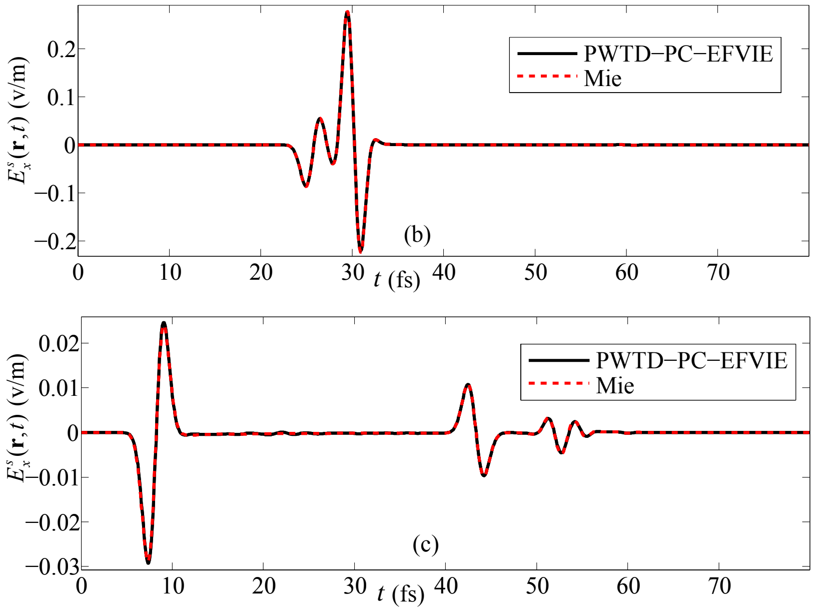


Figure 6.4 Scattered electric fields (x component) of the shell at positions (a) (0, $3\ \mu\text{m}$, 0), (b) (0, 0, $3\ \mu\text{m}$) and (c) (0, 0, $-3\ \mu\text{m}$) obtained from the PWTD-PC-EFVIE solver and Mie series solution.

6.3.2.2 Two-layer sphere

Next, the proposed solver is applied to the analysis of scattering from a two-layer sphere with inner radius $5.25\ \mu\text{m}$ and outer radius $5.4\ \mu\text{m}$. The permittivities of the inner and outer layers are $\varepsilon(\mathbf{r}) = 1.02$ and $\varepsilon(\mathbf{r}) = 1.08$, respectively. An electric field in (6.15) with $f_0 = 0$ THz and $f_{max} = 600$ THz is used to excite the object. The object is discretized with $N_s = 24{,}427{,}317$ source elements and $N_s' = 25{,}042{,}247$ observer elements upon setting $\Delta s = 0.03\ \mu\text{m}$. The simulation is performed $N_t = 1000$ time steps with $\Delta t = 0.1$ fs. A seven-level PWTD tree is constructed upon setting the edge length of boxes at the finest level to $0.34\ \lambda$ and $\gamma = 5$. The averaging factor ω_{mi} is chosen by setting $\tau_1 = 3\ t_0$ and $\tau_2 = 3.5\ t_0$ in (6.9). The simulation requires 15.5 hours of CPU time using $N_p = 4096$. To the best of authors' knowledge, this is the largest problem ever solved using TD-VIE solvers.

The bistatic RCS of the two-layer sphere at 199.7 THz obtained by the solver is compared with the Mie series solution [Figure 6.5]. The results show good agreement.

Figure 6.6 plots the scattered fields $E_x^s(\mathbf{r},t)$ at Cartesian coordinates ($5.7\ \mu\text{m}$, 0, 0), (0, 0, $5.7\ \mu\text{m}$) and (0, 0, $-5.7\ \mu\text{m}$) obtained from the solver and the Mie series solution. Again, good agreement demonstrates the accuracy of the proposed solver.

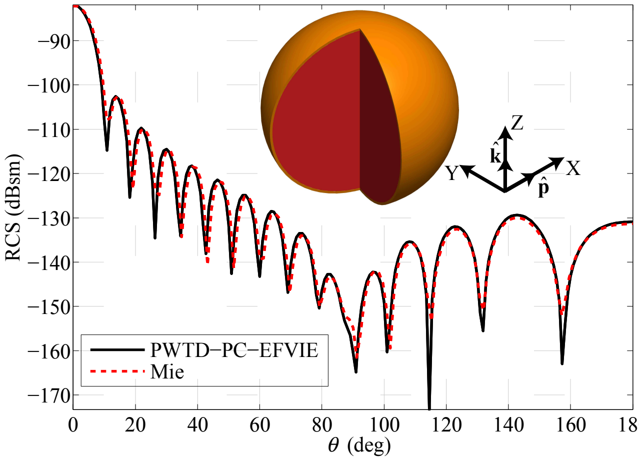
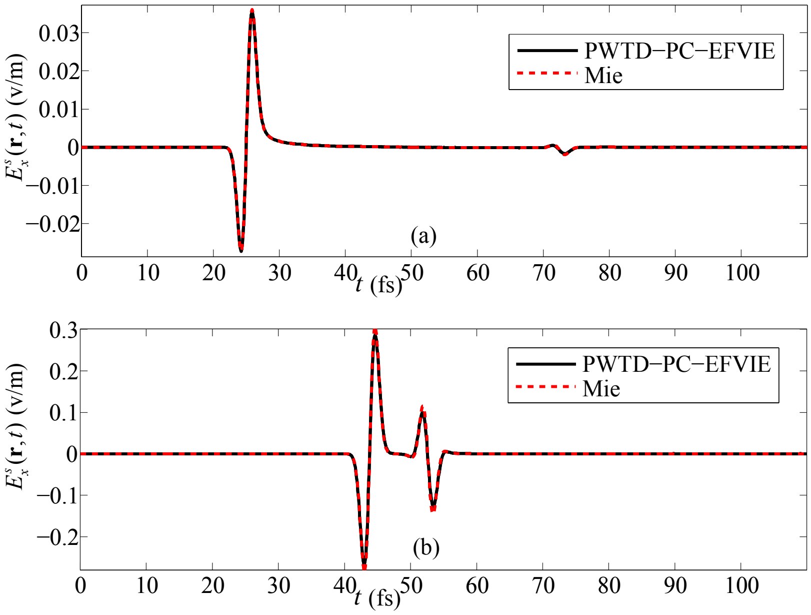


Figure 6.5 Bistatic RCS of the shell at 199.7 THz computed at $\phi = 0^\circ$ and $\theta = [0,180]^\circ$ via the PWTD-PC-EFVIE solver and Mie series solution.



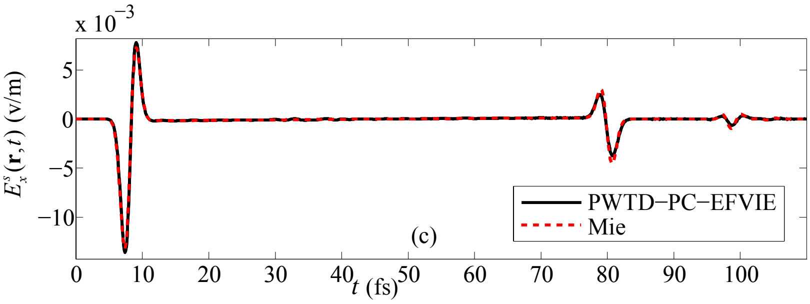


Figure 6.6 Scattered electric fields (x component) of the two-layer sphere at positions (a) ($5.7\ \mu\text{m}$, 0, 0), (b) (0, 0, $5.7\ \mu\text{m}$) and (c) (0, 0, $-5.7\ \mu\text{m}$) obtained from the PWTD-PC-EFVIE solver and Mie series solution.

6.3.3 Real-Life Example

Finally, the applicability of the PWTD-PC-EFVIE solver is demonstrated through its application to the transient analysis of light interaction with a red blood cell aggregation. The aggregation consists of eight red blood cells, which fit in a fictitious box with dimensions $18.3\ \mu\text{m} \times 8.7\ \mu\text{m} \times 14.5\ \mu\text{m}$ [Figure 6.7]. The geometrical details of each cell are described in [11]. The permittivities of the scatterers and the background medium are $\varepsilon(\mathbf{r}) = 1.97$ and $\varepsilon_0 = 1.81$, respectively. The scatterers are illuminated by an electric field in (6.15) with $f_0 = 400$ THz and $f_{max} = 600$ THz. The scatterers are discretized with $N_s = 11{,}746{,}563$ source elements and $N_s' = 13{,}775{,}837$ observer elements upon setting $\Delta s = 0.03\ \mu\text{m}$. The simulation is performed $N_t = 700$ time steps with $\Delta t = 0.134$ fs. An eight-level PWTD tree is constructed upon setting the edge length of boxes at the finest level to $0.375\ \lambda$ and $\gamma = 6$. The averaging factor ω_{mi} is chosen by setting $\tau_1 = 1.3\ t_0$ and $\tau_2 = 1.6\ t_0$ in (6.9). The simulation requires 23.5 hours of CPU time using $N_p = 4096$.

The scattered fields $E_x^s(\mathbf{r},t)$ at Cartesian coordinates (19.3 μm, 4 μm, 7.5 μm) and (10 μm, 4 μm, 0) are computed [Figure 6.7]. In addition, Figure 6.8 plots snapshots of the total electric fields $\mathbf{E}(\mathbf{r},t)$ inside the cells on the plane $y = 4\ \mu\text{m}$ at times $220\ \Delta t$, $340\ \Delta t$ and $500\ \Delta t$. These results help better understand the diffraction and refraction phenomena for light interacting with red blood cells.

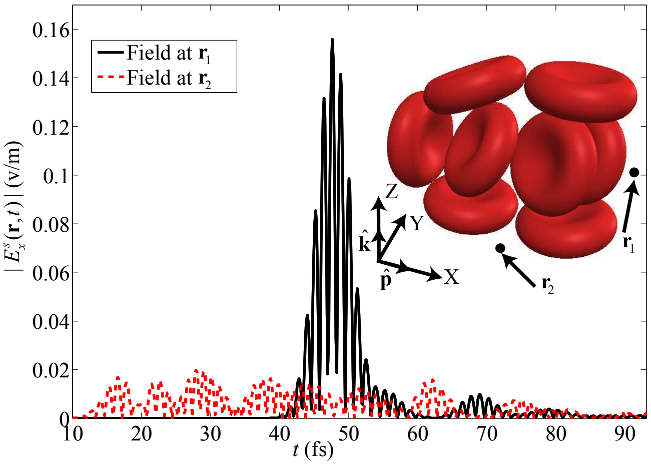


Figure 6.7 Scattered fields (magnitudes of x components) of the cell cluster at positions $\mathbf{r}_1$=(19.3 μm, 4 μm, 7.5 μm) and $\mathbf{r}_2$=(10 μm, 4 μm, 0 μm) obtained from the PWTD-PC-EFVIE solver.

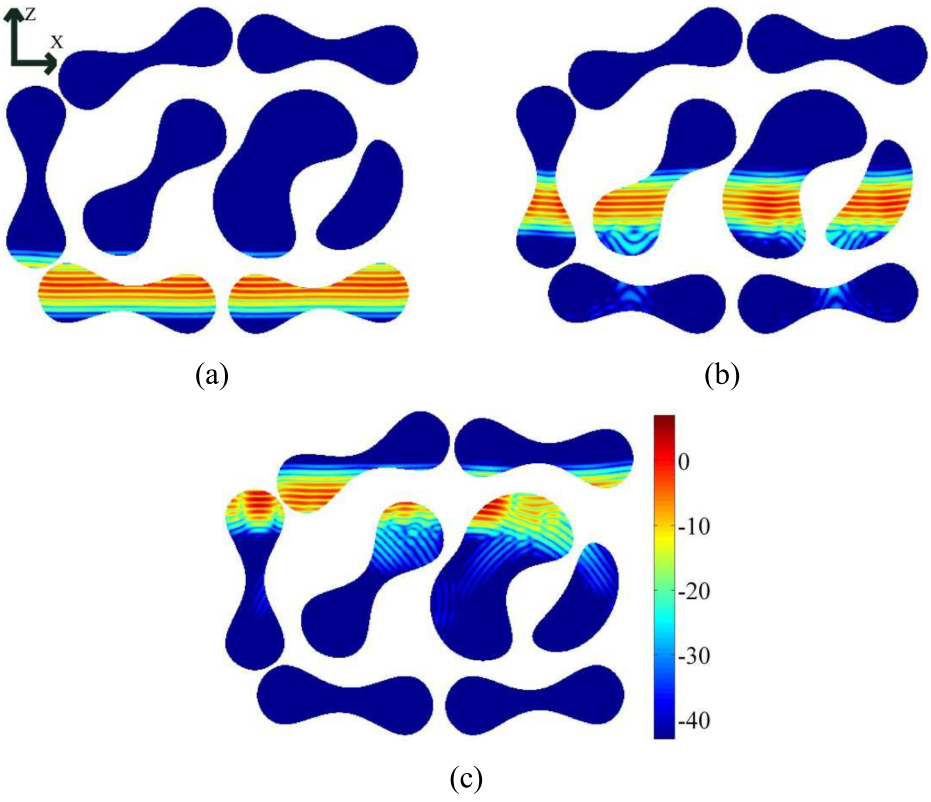


Figure 6.8 Snapshots of the total electric fields (in dB) induced in the red blood cell aggregation computed by PWTD-PC-EFVIE solver at (a) $t = 220\ \Delta t$, (b) $t = 340\ \Delta t$, (c) $t = 500\ \Delta t$.

6.4 Chapter Conclusion

This chapter presents a PWTD-enhanced, explicit MOT-based TD-EFVIE solver for analyzing electrically large transient scattering problems that involve inhomogeneous dielectric objects. The proposed solver leverages simple pulse spatial basis functions and Lagrange temporal basis functions for discretization of electric flux density, and employs finite difference to compute electric fields from magnetic vector potentials. An improved predictor-corrector scheme is proposed to maintain the stability of the solver. The computation of the vector potentials are accelerated by a scalar-field PWTD scheme, which reduces the overall memory and computational cost of the solver from $O(N_s^{4/3})$ and $O(N_t N_s^2)$ to $O(N_s \log N_s)$ and $O(N_t N_s)$, respectively. In addition, this PWTD-PC-EFVIE solver is parallelized on distributed-memory clusters by employing a hierarchical partitioning strategy that yields favorable load balance and parallel efficiencies when the number of processors is large. The proposed solver is applied to the analysis of transient scattering from canonical objects that involve 25 million spatial unknowns, which to the authors' knowledge, is the largest scattering problem ever simulated using TD-VIE solvers. In addition, the proposed solver is applied to the transient electromagnetic characterizations of a red blood cell aggregation that involves 12 million spatial unknowns.

CHAPTER 7
Conclusions

7.1 Summary

This thesis presents several techniques to advance the capabilities of multilevel PWTD-accelerated MOT-TDIE solvers in solving real-life and electrically large transient scattering problems.

Scalable CPU parallelization of the PWTD algorithm: We developed a highly scalable parallelization scheme of the multilevel PWTD algorithm for fast evaluation of transient fields due to surface-bound and volumetrically distributed source constellations. The load balance and scalability of the proposed parallelization scheme are rigorously proved and numerically validated. This parallel PWTD algorithm is capable of evaluating fields due to tens of millions of surface-bound and over one hundred million volumetrically distributed sources.

GPU implementations of the PWTD algorithm: We present GPU implementations of the multilevel PWTD algorithm using single and multiple GPUs. The single GPU implementation accelerates all stages of the PWTD algorithm and achieves 50X speedup compared to a serial CPU implementation; the multi-GPU implementation combines the single GPU acceleration and the CPU parallelization of the PWTD algorithm, and achieves 150X speedup compared to the serial CPU implementation.

Parallel PWTD-accelerated TD-SIE solver: We developed a highly scalable parallel PWTD-accelerated TD-SIE solver capable of analyzing transient scattering from electrically large canonical and real-life PEC objects that involve 10 million spatial unknowns.

Wavelet-enhanced PWTD-accelerated TD-SIE solver: We further enhance the capability of the abovementioned TD-SIE solver by leveraging a wavelet-domain PWTD algorithm. The computational and memory costs of this wavelet-enhanced PWTD-accelerated TD-SIE solver are theoretically proved and numerically validated for smooth quasi-planar PEC objects. The efficiency and accuracy of the proposed solver are demonstrated through its applications to the analysis of transient scattering from targets spanning well over one hundred wavelengths.

Explicit and PWTD-accelerated TD-EFVIE solver. We developed a PWTD-accelerated explicit MOT-TD-EFVIE solver capable of analyzing transient scattering from electrically large inhomogeneous dielectric objects. The efficiency and accuracy of the proposed solver are demonstrated through its applications to the analysis of transient scattering from canonical and real-life objects that involve 25 million spatial unknowns.

7.2 Future Work

First, although the CPU parallelized PWTD algorithm has been incorporated into the TD-EFIE/MFIE/CFIE and TD-EFVIE solvers for analyzing transient scattering from PEC and inhomogeneous dielectric objects residing in lossless background medium, its integration into TDIE solvers capable of handling transient analysis involving other types of objects and background media has not been studied. Second, the proposed GPU implementations of the PWTD algorithms have not been incorporated into MOT-TDIE solvers and may require more research work. Finally, the wavelet-enhanced PWTD-accelerated TD-SIE solver needs to be massively parallelized in order to solve large problems beyond those reported in this thesis.

7.3 Contributions

This research resulted in the following contributions [161-175]:

7.3.1 Journal Papers

[1] A. C. Yucel, Y. Liu, H. Bagci, and E. Michielssen, “Statistical characterization of electromagnetic wave propagation in mine environments,” *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 1602-1605, 2013.

[2] Y. Liu, A. C. Yucel, V. Lomakin, and E. Michielssen, “Graphics processing unit implementation of multilevel plane-wave time-domain algorithm,” *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 1671-1675, 2014.

[3] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, “A scalable parallel PWTD-accelerated surface integral equation solver for analysis of transient scattering from large-scale objects,” *IEEE Trans. Antennas Propag.* (in review).

[4] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, “A provable scalable parallel multilevel plane-wave time-domain algorithm,” (in preparation, will be submitted to *J. Comput. Phys.*)

[5] Y. Liu, A. Al-Jarro, H. Bagci, and E. Michielssen, “Explicit and PWTD-enhanced solution of the time domain volume integral equation,” (in preparation, will be submitted to *IEEE Trans. Antennas Propag.*)

[6] Y. Liu, and E. Michielssen, “Plane-wave time-domain algorithms for transient electromagnetic analysis: a review” (in preparation, will be submitted to *IEEE Antennas Propag.Mag.*)

[7] Y. Liu, A. C. Yucel, H. Bagci, A. C. Gilbert, and E. Michielssen, “Wavelet-enhanced plane-wave time-domain algorithm for analysis of transient scattering from electrically large conducting objects,” (in preparation, will be submitted to *IEEE Trans. Antennas Propag.*)

[8] A C. Yucel, Y. Liu, C. Zhou, H. Bagci, R. Jacksha, and E. Michielssen, “An FMM-FFT accelerated SIE simulator for characterizing EM wave propagation in mine environments loaded with conductors,” (in preparation, will be submitted to *IEEE Trans. Antennas Propag.*)

7.3.2 Conference Papers

[1] Y. Liu, V. Lomakin, and E. Michielssen, “Graphics processing unit-accelerated implementation of the plane wave time domain algorithm,” in *Proc. 28th Ann. Rev. Prog. Appl. Computat. Electromagn.*, 2012.

[2] Y. Liu, A. C. Yucel, V. Lomakin, and E. Michielssen, “A scalable parallel implementation of the plane wave time domain algorithm on graphics processing unit-augmented clusters,” in *Proc. IEEE Int. Symp. AP-S/URSI*, 2012.

[3] Y. Liu, H. Bagci, and E. Michielssen, “Progress in parallel implementation of the multilevel plane wave time domain algorithm,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2013.

[4] Y. Liu, H. Bagci, and E. Michielssen, “Solving very large scattering problems using a parallel PWTD-enhanced surface integral equation solver,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2013.

[5] Y. Liu, A. Al-Jarro, H. Bagci, and E. Michielssen, “Parallel, explicit, and PWTD-enhanced time domain volume integral equation solver,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2013.

[6] A. C. Yucel, Y. Liu, H. Bagci, and E. Michielssen, “A fast-multipole domain decomposition integral equation solver for characterizing electromagnetic wave propagation in mine environments,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2013.

[7] H. Guo, Y. Liu, H. Jun, and E. Michielssen, “A parallel MLMDA-based direct integral equation solver,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2013.

[8] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, “Parallel time domain solvers for electrically large transient scattering problems,” in *Proc. EUCAP,* 2014.

[9] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, “A parallel wavelet-enhanced PWTD algorithm for analyzing transient scattering from electrically very large PEC targets,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2014.

[10] A. C. Yucel, L. Gomez, Y. Liu, H. Bagci, and E. Michielssen, “A FMM-FFT accelerated hybrid volume surface integral equation solver for electromagnetic analysis of re-entry space vehicles,” in *Proc. IEEE Int. Symp. AP-S/URSI,* 2014.

[11] A. C. Yucel, Y. Liu, H. Bagci, and E. Michielssen, “An FMM-FFT accelerated integral equation solver for characterizing electromagnetic wave propagation

in mine tunnels and galleries loaded with conductors," in *Proc. IEEE Int. Symp. AP-S/URSI,* 2014.

[12] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, "A wavelet-based PWTD algorithm-accelerated time domain surface integral equation solver," in *Proc. IEEE Int. Symp. AP-S/URSI,* 2015.

APPENDIX A
Derivation of Equation (5.19)

To prove the temporal localization property of the convolution of the LCB function with the translation function $\mathcal{T}(\hat{\mathbf{k}}^v_{pq},t)*T^+_{ru}(t)$, we consider its Fourier transform $\overline{\mathcal{T}}(\hat{\mathbf{k}}^v_{qp},\omega)\overline{T}^+_{ru}(\omega)$. The Fourier transform $\overline{\mathcal{T}}(\hat{\mathbf{k}}^v_{qp},\omega)\cdot\overline{T}^+_{ru}(\omega)$ is:

$$\begin{aligned}\mathcal{T}(\hat{\mathbf{k}}^v_{qp},\omega)\overline{T}^+_{ru}(\omega) &= 2\imath\omega\overline{T}^+_{ru}(\omega)\times\\ &\sum_{k=0}^{K^v}(-\imath)^k\left(2k+1\right)j_k\left(\omega R_{c,\alpha\alpha'}/c\right)\Phi_k\left(\cos\theta\right)\end{aligned} \tag{A.1}$$

where $\imath$ is the imaginary unit and $j_k(\cdot)$ is the spherical Bessel function of order k. As discussed in Section 5.3.1, the spectrum $\overline{T}^+_{ru}(\omega)$ of the LCB function has central frequency $\omega_u = \pi(u-1/2)/(M^r\Delta t)$ and bandwidth ω_{bw}. Here $\omega_{bw} \approx C\pi/(M^r\Delta t)$ for some constant C. In other words, $\overline{T}^+_{ru}(\omega)=0$ if $\omega<\omega_u-\omega_{bw}$ or $\omega>\omega_u+\omega_{bw}$. When ω is sufficiently large (equivalently, u is large), the spherical Bessel function asymptotically behaves as

$$j_k(\omega R_{c,\alpha\alpha'}/c) \approx \frac{c\sin(\omega R_{c,\alpha\alpha'}/c-1/2k\pi)}{\omega R_{c,\alpha\alpha'}}. \tag{A.2}$$

Substituting (A.2) into (A.1), $\overline{\mathcal{T}}(\hat{\mathbf{k}}^v_{qp},\omega)\overline{T}^+_{ru}(\omega)$ can be re-written as

$$\begin{aligned}\overline{\mathcal{T}}(\hat{\mathbf{k}}^v_{qp},\omega)\overline{T}^+_{ru}(\omega) &\approx \frac{c}{R_{c,\alpha\alpha'}}e^{\imath\omega R_{c,\alpha\alpha'}/c}\overline{T}^+_{ru}(\omega)f_1(\cos\theta)\\ &-\frac{c}{R_{c,\alpha\alpha'}}e^{-\imath\omega R_{c,\alpha\alpha'}/c}\overline{T}^+_{ru}(\omega)f_2(\cos\theta)\end{aligned} \tag{A.3}$$

where $f_1(\cos\theta)=\sum_{k=0}^{K^v}(-1)^k\left(2k+1\right)\Phi_k\left(\cos\theta\right)$ and $f_2(\cos\theta)=\sum_{k=0}^{K^v}\left(2k+1\right)\Phi_k\left(\cos\theta\right)$. Upon inverse Fourier transform of (A.3), the convolution of the LCB function and the translation function, $\mathcal{T}(\hat{\mathbf{k}}_{pq}^v,t)*T_{ru}^{+}(t)$, can expressed as

$$\begin{aligned}\mathcal{T}(\hat{\mathbf{k}}_{pq}^v,t)*T_{ru}^{+}(t)\approx&\frac{c}{R_{c,\alpha\alpha'}}T_{ru}^{+}(t-R_{c,\alpha\alpha'}/c)f_1(\cos\theta)\\&-\frac{c}{R_{c,\alpha\alpha'}}T_{ru}^{+}(t+R_{c,\alpha\alpha'}/c)f_2(\cos\theta).\end{aligned}\tag{A.4}$$

REFERENCES

[1] 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, pp. 1064-1074, 2000.

[2] W. C. Chew, E. Michielssen, J. M. Song, and J. M. Jin, *Fast and efficient algorithms in computational electromagnetics*: Artech House, Inc., 2001.

[3] E. Michielssen, A. Yilmaz, H. Bagci, and J. Jin, "Fast time domain integral equation based electromagnetic analysis: A maturing technology," *Proc. Eur. Conf. Comput. Fluid Dyn.,* 2006.

[4] H. Bagci, A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "Fast and rigorous analysis of EMC/EMI phenomena on electrically large and complex cable-loaded structures," *IEEE Trans. Electromagn. Compat.,* vol. 49, pp. 361-381, 2007.

[5] H. Bagci, A. C. Yucel, J. S. Hesthaven, and E. Michielssen, "A fast stroud-based collocation method for statistically characterizing EMI/EMC phenomena on complex platforms," *IEEE Trans. Electromagn. Compat.,* vol. 51, pp. 301-311, 2009.

[6] N.-W. Chen, B. Shanker, and E. Michielssen, "Integral-equation-based analysis of transient scattering from periodic perfectly conducting structures," *IEE P-Microw. Anten. P.,* vol. 150, pp. 120-124, 2003.

[7] M. Pantoja, D. Werner, P. Werner, and A. Bretones, "TDIE modeling of carbon nanotube dipoles at microwave and terahertz bands," *IEEE Antennas Wireless Propag. Lett.,* vol. 9, pp. 32-35, 2010.

[8] 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, pp. 39-52, 1999.

[9] A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "Time domain adaptive integral method for surface integral equations," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 2692-2708, 2004.

[10] J. Meng, A. Boag, V. Lomakin, and E. Michielssen, "A multilevel Cartesian non-uniform grid time domain algorithm," *J. Comput. Phys.,* vol. 229, pp. 8430-8444, 2010.

[11] S. Rao, D. Wilton, and A. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.,* vol. 30, pp. 409-418, 1982.

[12] 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, pp. 527-532, 1997.

[13] 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, pp. 628-641, 2003.

[14] H. Mieras and C. Bennett, "Space-time integral equation approach to dielectric targets," *IEEE Trans. Antennas Propag.,* vol. 30, pp. 2-9, 1982.

[15] 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. Am. A,* vol. 19, pp. 716-726, 2002.

[16] D. A. Vechinski, S. M. Rao, and T. K. Sarkar, "Transient scattering from three-dimensional arbitrarily shaped dielectric bodies," *J. Opt. Soc. Am. A,* vol. 11, pp. 1458-1470, 1994.

[17] G. Pisharody and D. S. Weile, "Electromagnetic scattering from homogeneous dielectric bodies using time-domain integral equations," *IEEE Trans. Antennas Propag.,* vol. 54, pp. 687-697, 2006.

[18] B. Shanker, M. Lu, J. Yuan, and E. Michielssen, "Time domain integral equation analysis of scattering from composite bodies via exact evaluation of radiation fields," *IEEE Trans. Antennas Propag.,* vol. 57, pp. 1506-1520, 2009.

[19] N. T. Gres, A. A. Ergin, E. Michielssen, and B. Shanker, "Volume-integral-equation-based analysis of transient electromagnetic scattering from three-dimensional inhomogeneous dielectric objects," *Radio Sci.,* vol. 36, pp. 379-386, 2001.

[20] B. Shanker, K. Aygun, and E. Michielssen, "Fast analysis of transient scattering from lossy inhomogeneous dielectric bodies," *Radio Sci.,* vol. 39, 2004.

[21] G. Kobidze, J. Gao, B. Shanker, and E. Michielssen, "A fast time domain integral equation based scheme for analyzing scattering from dispersive objects," *IEEE Trans. Antennas Propag.,* vol. 53, pp. 1215-1226, 2005.

[22] A. E. Yilmaz, D. S. Weile, B. Shanker, J.-M. Jin, and E. Michielssen, "Fast analysis of transient scattering in lossy media," *IEEE Antennas Wireless Propag. Lett.,* vol. 1, pp. 14-17, 2002.

[23] H. Bagci, A. E. Yilmaz, V. Lomakin, and E. Michielssen, "Fast solution of mixed-potential time-domain integral equations for half-space environments," *IEEE Trans. Geosci. Remote Sens.,* vol. 43, pp. 269-279, 2005.

[24] M. Lu, K. Yegin, E. Michielssen, and B. Shanker, "Fast time domain integral equation solvers for analyzing two-dimensional scattering phenomena; Part I: temporal acceleration," *Electromagnetics,* vol. 24, pp. 425-449, 2004.

[25] M. Ghaffari-Miab, S. M. H. Haddad, and R. Faraji-Dana, "A new fast and accurate time domain formulation of the method of moment (TD-MoM) for thin-wire antennas," in *Asia Pac. Microw. Conf.*, ed, 2009, pp. 72-75.

[26] N.-W. Chen, K. Aygun, and E. Michielssen, "Integral-equation-based analysis of transient scattering and radiation from conducting bodies at very low frequencies," *IEE P-Microw. Anten. P.,* vol. 148, pp. 381-387, 2001.

[27] R. A. Wildman, G. Pisharody, D. S. Weile, S. Balasubramaniam, and E. Michielssen, "An accurate scheme for the solution of the time-domain integral equations of electromagnetics using higher order vector bases and bandlimited extrapolation," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 2973-2984, 2004.

[28] G. Pisharody and D. S. Weile, "Robust solution of time-domain integral equations using loop-tree decomposition and bandlimited extrapolation," *IEEE Trans. Antennas Propag.,* vol. 53, pp. 2089-2098, 2005.

[29] F. P. Andriulli, H. Bagci, F. Vipiana, G. Vecchi, and E. Michielssen, "A marching-on-in-time hierarchical scheme for the solution of the time domain electric field integral equation," *IEEE Trans. Antennas Propag.,* vol. 55, pp. 3734-3738, 2007.

[30] F. P. Andriulli, H. Bagci, F. Vipiana, G. Vecchi, and E. Michielssen, "Analysis and regularization of the TD-EFIE low-frequency breakdown," *IEEE Trans. Antennas Propag.,* vol. 57, pp. 2034-2046, 2009.

[31] H. Bagci, F. P. Andriulli, F. Vipiana, G. Vecchi, and E. Michielssen, "A well-conditioned integral-equation formulation for efficient transient analysis of electrically small microelectronic devices," *IEEE Trans. Adv. Packag.,* vol. 33, pp. 468-480, 2010.

[32] F. P. Andriulli, K. Cools, F. Olyslager, and E. Michielssen, "Time domain Calderón identities and their application to the integral equation analysis of scattering by PEC objects part II: Stability," *IEEE Trans. Antennas Propag.,* vol. 57, pp. 2365-2375, 2009.

[33] K. Cools, F. P. Andriulli, F. Olyslager, and E. Michielssen, "Time domain Calderón identities and their application to the integral equation analysis of scattering by PEC objects Part I: Preconditioning," *IEEE Trans. Antennas Propag.,* vol. 57, pp. 2352-2364, 2009.

[34] K. Cools, F. P. Andriulli, F. Olyslager, and E. Michielssen, "Calderón preconditioned time-domain integral equation solvers," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2007, pp. 4565-4568.

[35] F. P. Andriulli, K. Cools, F. Olyslager, and E. Michielssen, "Calderón stabilized time domain integral equation solvers," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2007, pp. 4573-4576.

[36] F. Valdes, F. Andriulli, H. Bagci, and E. Michielssen, "Time-domain single-source integral equations for analyzing scattering from homogeneous penetrable objects," *IEEE Trans. Antennas Propag.,* vol. 61, pp. 1239-1254, 2013.

[37] J.-L. Hu and C. H. Chan, "Improved temporal basis function for time domain electric field integral equation method," *Electron. Lett.,* vol. 35, pp. 883-885, 1999.

[38] J.-L. Hu, C. H. Chan, and Y. Xu, "A new temporal basis function for the time-domain integral equation method," *IEEE Microw. Wireless Compon. Lett.,* vol. 11, pp. 465-466, 2001.

[39] E. van't Wout, D. R. van der Heul, H. van der Ven, and C. Vuik, "Design of temporal basis functions for time domain integral equation methods with predefined accuracy and smoothness," *IEEE Trans. Antennas Propag.,* vol. 61, pp. 271-280, 2013.

[40] A. Geranmayeh, W. Ackermann, and T. Weiland, "Temporal discretization choices for stable boundary element methods in electromagnetic scattering problems," *Appl. Numer. Math.,* vol. 59, pp. 2751-2773, 2009.

[41] P. Wang, M. Y. Xia, J. M. Jin, and L. Z. Zhou, "Time-domain integral equation solvers using quadratic B-spline temporal basis functions," *Microw. Opt. Techn. Let.,* vol. 49, pp. 1154-1159, 2007.

[42] J. Knab, "Interpolation of band-limited functions using the approximate prolate series," *IEEE Trans. Inf. Theory,* vol. 25, pp. 717-720, 1979.

[43] D. S. Weile, G. Pisharody, N.-W. Chen, B. Shanker, and E. Michielssen, "A novel scheme for the solution of the time-domain integral equations of electromagnetics," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 283-295, 2004.

[44] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propag.,* vol. 45, pp. 329-342, 1997.

[45] R. A. Wildman and D. S. Weile, "An accurate broad-band method of moments using higher order basis functions and tree-loop decomposition," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 3005-3011, 2004.

[46] F. Valdes, F. P. Andriulli, K. Cools, and E. Michielssen, "High-order div-and quasi curl-conforming basis functions for Calderón multiplicative preconditioning of the EFIE," *IEEE Trans. Antennas Propag.,* vol. 59, pp. 1321-1337, 2011.

[47] F. Valdes, M. Ghaffari-Miab, F. Andriulli, K. Cools, and E. Michielssen, "High-order Calderón preconditioned time domain integral equation solvers," *IEEE Trans. Antennas Propag.,* vol. 61, pp. 2570-2588, 2013.

[48] R. A. Wildman and D. S. Weile, "Two-dimensional transverse-magnetic time-domain scattering using the Nyström method and bandlimited extrapolation," *IEEE Trans. Antennas Propag.,* vol. 53, pp. 2259-2266, 2005.

[49] N. V. Nair, A. J. Pray, and B. Shanker, "Analysis of transient scattering from PEC objects using the generalized method of moments," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2010, pp. 1-4.

[50] S. P. Walker, M. J. Bluck, and I. Chatzis, "The stability of integral equation time-domain scattering computations for three-dimensional scattering; similarities and differences between electrodynamic and elastodynamic computations," *Int. J. Numer. Model. El.,* vol. 15, pp. 459-474, 2002.

[51] B. P. Rynne, "Stability and convergence of time marching methods in scattering problems," *IMA J. Appl. Math.,* vol. 35, pp. 297-310, 1985.

[52] B. P. Rynne and P. D. Smith, "Stability of time marching algorithms for the electric field integral equation," *J. Electromagnet. Wave,* vol. 4, pp. 1181-1205, 1990.

[53] P. D. Smith, "Instabilities in time marching methods for scattering: Cause and rectification," *Electromagnetics,* vol. 10, pp. 439-451, 1990.

[54] D. A. Vechinski and S. M. Rao, "A stable procedure to calculate the transient scattering by conducting surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.,* vol. 40, pp. 661-665, 1992.

[55] A. Sadigh and E. Arvas, "Treating the instabilities in marching-on-in-time method from a different perspective," *IEEE Trans. Antennas Propag.,* vol. 41, pp. 1695-1702, 1993.

[56] P. J. Davies, "A stability analysis of a time marching scheme for the general surface electric field integral equation," *Appl. Numer. Math.,* vol. 27, pp. 33-57, 1998.

[57] Y. Beghein, K. Cools, H. Bagci, and D. De Zutter, "A space-time mixed Galerkin marching-on-in-time scheme for the time-domain combined field integral equation," *IEEE Trans. Antennas Propag.,* vol. 61, pp. 1228-1238, 2013.

[58] T. Abboud, J. Nedelec, and J. Volakis, "Stable solution of the retarded potential equations," in *17th Ann. Rev. Prog. Appl. Computat. Electromagn.*, ed, 2001.

[59] P. J. Davies and D. B. Duncan, "Stability and convergence of collocation schemes for retarded potential integral equations," *SIAM J. Numer. Anal.,* vol. 42, pp. 1167-1188, 2004.

[60] M. J. Bluck and S. P. Walker, "Time-domain BIE analysis of large three-dimensional electromagnetic scattering problems," *IEEE Trans. Antennas Propag.,* vol. 45, pp. 894-901, 1997.

[61] S. J. Dodson, S. P. Walker, and M. J. Bluck, "Implicitness and stability of time domain integral equation scattering analysis," *Appl. Comp. Electromag. Soc. J.,* vol. 13, pp. 291-301, 1998.

[62] X. Wang and D. S. Weile, "Electromagnetic scattering from dispersive dielectric scatterers using the finite difference delay modeling method," *IEEE Trans. Antennas Propag.,* vol. 58, pp. 1720-1730, 2010.

[63] X. Wang and D. S. Weile, "Implicit Runge-Kutta methods for the discretization of time domain integral equations," *IEEE Trans. Antennas Propag.,* vol. 59, pp. 4651-4663, 2011.

[64] X. Wang, R. A. Wildman, D. S. Weile, and P. Monk, "A finite difference delay modeling approach to the discretization of the time domain integral equations of electromagnetics," *IEEE Trans. Antennas Propag.,* vol. 56, pp. 2442-2452, 2008.

[65] M. Lu and E. Michielssen, "Closed form evaluation of time domain fields due to Rao-Wilton-Glisson sources for use in marching-on-in-time based EFIE solvers," in *Proc. IEEE Int. Symp. AP-S/URSI* vol. 1, ed, 2002, pp. 74-77.

[66] Y. Shi, M.-Y. Xia, R.-S. Chen, E. Michielssen, and M. Lu, "Stable electric field TDIE solvers via quasi-exact evaluation of MOT matrix elements," *IEEE Trans. Antennas Propag.,* vol. 59, pp. 574-585, 2011.

[67] A. C. Yucel and A. A. Ergin, "Exact evaluation of retarded-time potential integrals for the RWG bases," *IEEE Trans. Antennas Propag.,* vol. 54, pp. 1496-1502, 2006.

[68] H. A. Ulku and A. A. Ergin, "On the singularity of the closed-form expression of the magnetic field in time domain," *IEEE Trans. Antennas Propag.,* vol. 59, pp. 691-694, 2011.

[69] H. A. Ulku and A. A. Ergin, "Analytical evaluation of transient magnetic fields due to RWG current bases," *IEEE Trans. Antennas Propag.,* vol. 55, pp. 3565-3575, 2007.

[70] H. A. Ulku and A. A. Ergin, "Application of analytical retarded-time potential expressions to the solution of time domain integral equations," *IEEE Trans. Antennas Propag.,* vol. 59, pp. 4123-4131, 2011.

[71] J. Pingenot, S. Chakraborty, and V. Jandhyala, "Polar integration for exact space-time quadrature in time-domain integral equations," *IEEE Trans. Antennas Propag.,* vol. 54, pp. 3037-3042, 2006.

[72] A. J. Pray, N. V. Nair, and B. Shanker, "Stability properties of the time domain electric field integral equation using a separable approximation for the

convolution with the retarded potential," *IEEE Trans. Antennas Propag.,* vol. 60, pp. 3772-3781, 2012.

[73] M.-D. Zhu, X.-L. Zhou, and W.-Y. Yin, "Efficient evaluation of double surface integrals in time-domain integral equation formulations," *IEEE Trans. Antennas Propag.,* vol. 61, p. 4653, 2013.

[74] G. Pisharody and D. S. Weile, "Electromagnetic scattering from perfect electric conductors using an augmented time-domain integral-equation technique," *Microw. Opt. Techn. Let.,* vol. 45, pp. 26-31, 2005.

[75] 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, pp. 4389-4392, 2013.

[76] M. Vikram and B. Shanker, "Fast evaluation of time domain fields in sub-wavelength source/observer distributions using accelerated Cartesian expansions (ACE)," *J. Comput. Phys.,* vol. 227, pp. 1007-1023, 2007.

[77] B. Shanker and H. Huang, "Accelerated Cartesian expansions–A fast method for computing of potentials of the form R− ν for all real ν," *J. Comput. Phys.,* vol. 226, pp. 732-753, 2007.

[78] Z. Zhou and J. S. Tyo, "An adaptive time-domain integral equation method for transient analysis of wire scatterer," *IEEE Antennas Wireless Propag. Lett.,* vol. 4, pp. 147-150, 2005.

[79] G. Kaur and A. E. Yilmaz, "On the performance of envelope-tracking surface-integral equation solvers," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2011, pp. 2716-2719.

[80] A. Mohan and D. S. Weile, "A hybrid method of moments-marching on in time method for the solution of electromagnetic scattering problems," *IEEE Trans. Antennas Propag.,* vol. 53, pp. 1237-1242, 2005.

[81] S. P. Walker and M. J. Vartiainen, "Hybridization of curvilinear time-domain integral equation and time-domain optical methods for electromagnetic scattering analysis," *IEEE Trans. Antennas Propag.,* vol. 46, pp. 318-324, 1998.

[82] G. Kobidze, B. Shanker, and E. Michielssen, "Hybrid PO-PWTD scheme for analyzing of scattering from electrically large PEC objects," in *Proc. IEEE Int. Symp. AP-S/URSI* vol. 3, ed, 2003, pp. 547-550.

[83] R. Meng, Z. Dongming, L. Ying, and H. Jianguo, "Coupled TDIE-PO method for transient scattering from electrically large conducting objects," *Electron. Lett.,* vol. 44, pp. 258-260, 2008.

[84] W. Luo, W.-Y. Yin, M.-D. Zhu, and J.-Y. Zhao, "Hybrid TDIE-TDPO method for studying on transient responses of some wire and surface structures illuminated by an electromagnetic pulse," *Prog. Electromagn. Res.,* vol. 116, pp. 203-219, 2011.

[85] W. Luo, W.-Y. Yin, M.-D. Zhu, J.-F. Mao, and J.-Y. Zhao, "Investigation on time-and frequency-domain responses of some complex composite structures in the presence of high-power electromagnetic pulses," *IEEE Trans. Electromagn. Compat.,* vol. 54, pp. 1006-1016, 2012.

[86] D. Jiao, M. Lu, E. Michielssen, and J.-M. Jin, "A fast time-domain finite element-boundary integral method for electromagnetic analysis," *IEEE Trans. Antennas Propag.,* vol. 49, pp. 1453-1461, 2001.

[87] D. Jiao, A. A. Ergin, B. Shanker, E. Michielssen, and J.-M. Jin, "A fast higher-order time-domain finite element-boundary integral method for 3-D electromagnetic scattering analysis," *IEEE Trans. Antennas Propag.,* vol. 50, pp. 1192-1202, 2002.

[88] A. McCowen, A. J. Radcliffe, and M. S. Towers, "Time-domain modeling of scattering from arbitrary cylinders in two dimensions using a hybrid finite-element and integral equation method," *IEEE Trans. Magn.,* vol. 39, pp. 1227-1229, 2003.

[89] A. Monorchio, A. R. Bretones, R. Mittra, G. Manara, and R. G. Martín, "A hybrid time-domain technique that combines the finite element, finite difference and method of moment techniques to solve complex electromagnetic problems," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 2666-2674, 2004.

[90] A. E. Yilmaz, Z. Lou, E. Michielssen, and J.-M. Jin, "A single-boundary implicit and FFT-accelerated time-domain finite element-boundary integral solver," *IEEE Trans. Antennas Propag.,* vol. 55, pp. 1382-1397, 2007.

[91] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.,* vol. 31, pp. 1225-1251, 1996.

[92] A. E. Yilmaz, D. S. Weile, H.-M. Jin, and E. Michielssen, "A hierarchical FFT algorithm (HIL-FFT) for the fast analysis of transient electromagnetic scattering phenomena," *IEEE Trans. Antennas Propag.,* vol. 50, pp. 971-982, 2002.

[93] A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "Analysis of low-frequency electromagnetic transients by an extended time-domain adaptive integral method," *IEEE Trans. Adv. Packag.,* vol. 30, pp. 301-312, 2007.

[94] A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "A parallel FFT accelerated transient field-circuit simulator," *IEEE Trans. Microw. Theory Tech.,* vol. 53, pp. 2851-2865, 2005.

[95] H. Bagci, A. E. Yilmaz, and E. Michielssen, "An FFT-accelerated time-domain multiconductor transmission line simulator," *IEEE Trans. Electromagn. Compat.,* vol. 52, pp. 199-214, 2010.

[96] A. Boag, V. Lomakin, and E. Michielssen, "Nonuniform grid time domain (NGTD) algorithm for fast evaluation of transient wave fields," *IEEE Trans. Antennas Propag.,* vol. 54, pp. 1943-1951, 2006.

[97] B. Shanker, A. A. Ergin, K. Aygun, and E. Michielssen, "Analysis of transient electromagnetic scattering phenomena using a two-level plane wave time-domain algorithm," *IEEE Trans. Antennas Propag.,* vol. 48, pp. 510-523, 2000.

[98] J. Gao, B. Shanker, D. S. Weile, and E. Michielssen, "Analysis of transient scattering from multiregion bodies using the plane wave time domain algorithm," *Proc. IEEE Int. Symp. AP-S/URSI,* 2003.

[99] P. Jiang, K. Yegin, S. Q. Li, B. Shanker, and E. Michielssen, "An improved plane wave time domain algorithm for dissipative media," in *Proc. IEEE Int. Symp. AP-S/URSI* vol. 3, ed, 2003, pp. 563-566.

[100] P.-L. Jiang and E. Michielssen, "Temporal acceleration of time-domain integral-equation solvers for electromagnetic scattering from objects residing in lossy media," *Microw. Opt. Techn. Let.,* vol. 44, pp. 223-230, 2005.

[101] P. L. Jiang and E. Michielssen, "Multilevel plane wave time domain-enhanced MOT solver for analyzing electromagnetic scattering from objects residing in lossy media," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2005.

[102] P.-L. Jiang and E. Michielssen, "Multilevel PWTD-enhanced CFIE solver for analyzing EM scattering from PEC objects residing in lossy media," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2006, pp. 2967-2970.

[103] K. Aygun, M. Lu, B. Shanker, and E. Michielssen, "Analysis of PCB level EMI phenomena using an adaptive low-frequency plane wave time domain algorithm," in *Proc. IEEE Int. Symp. Electromagn. Compat.* vol. 1, ed, 2000, pp. 295-300.

[104] K. Aygun, B. Shanker, and E. Michielssen, "Low frequency plane wave time domain kernels," in *Proc. Int. Conf. Electromagn. Adv. Appl.*, ed, 2001, pp. 782-869.

[105] M. Lu, A. A. Ergin, and E. Michielssen, "Fast evaluation of transient fields in the presence of two half spaces using a plane wave time domain algorithm," *Proc. IEEE Int. Symp. AP-S/URSI,* 2000.

[106] N. Chen, M. Lu, B. Shanker, F. Capolino, and E. Michielssen, "Fast integral-equation-based analysis of transient scattering from doubly periodic perfectly conducting structures," in *Proc. IEEE Int. Symp. AP-S/URSI*, ed, 2003.

[107] M. Lu, E. Michielssen, and B. Shanker, "Fast time domain integral equation solvers for analyzing two-dimensional scattering phenomena; Part II: full PWTD acceleration," *Electromagnetics,* vol. 24, pp. 451-470, 2004.

[108] M. Lu, J. Wang, A. A. Ergin, and E. Michielssen, "Fast evaluation of two-dimensional transient wave fields," *J. Comput. Phys.,* vol. 158, pp. 161-185, 2000.

[109] M. Lu, M. Lv, A. A. Ergin, B. Shanker, and E. Michielssen, "Multilevel plane wave time domain-based global boundary kernels for two-dimensional finite difference time domain simulations," *Radio Sci.,* vol. 39, p. RS4007, 2004.

[110] M. Lu, B. Shanker, and E. Michielssen, "Elimination of spurious solutions associated with exact transparent boundary conditions in FDTD solvers," *IEEE Antennas Wireless Propag. Lett.,* vol. 3, pp. 59-62, 2004.

[111] B. Shanker, M. Lu, A. A. Ergin, and E. Michielssen, "Plane-wave time-domain accelerated radiation boundary kernels for FDTD analysis of 3-D electromagnetic phenomena," *IEEE Trans. Antennas Propag.,* vol. 53, pp. 3704-3716, 2005.

[112] A. A. Ergin, B. Shanker, and E. Michielssen, "Fast transient analysis of acoustic wave scattering from rigid bodies using a two-level plane wave time domain algorithm," *J. Acoust. Soc. Am.,* vol. 106, p. 2405, 1999.

[113] A. A. Ergin, B. Shanker, and E. Michielssen, "Fast analysis of transient acoustic wave scattering from rigid bodies using the multilevel plane wave time domain algorithm," *J. Acoust. Soc. Am.,* vol. 107, p. 1168, 2000.

[114] T. Takahashi, N. Nishimura, and S. Kobayashi, "A fast BIEM for three-dimensional elastodynamics in time domain," *Eng. Anal. Bound. Elem.,* vol. 27, pp. 491-506, 2003.

[115] 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* vol. 4, ed, 2004, pp. 4212-4215.

[116] F. Bost, L. Nicolas, and G. Rojat, "A time-domain integral formulation for the scattering by thin wires," *IEEE Trans. Magn.,* vol. 36, pp. 868-871, 2000.

[117] Z. Ji, T. K. Sarkar, B. H. Jung, Y.-S. Chung, M. Salazar-Palma, and M. Yuan, "A stable solution of time domain electric field integral equation for thin-wire antennas using the Laguerre polynomials," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 2641-2649, 2004.

[118] P. S. Excell, A. D. Tinniswood, and R. W. Clarke, "An independently fed log-periodic antenna for directed pulsed radiation," *IEEE Trans. Electromagn. Compat.,* vol. 41, pp. 344-349, 1999.

[119] A. O. Boryssenko and D. H. Schaubert, "Time-domain integral-equation-based solver for transient and broadband problems in electromagnetics," in *Proc. Int. Conf. UWB/SP Electromagn.*, ed: Springer, 2004, pp. 239-249.

[120] K. Aygun, B. Shanker, A. A. Ergin, and E. Michielssen, "A two-level plane wave time-domain algorithm for fast analysis of EMC/EMI problems," *IEEE Trans. Electromagn. Compat.,* vol. 44, pp. 152-164, 2002.

[121] 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.

[122] K. Aygun, B. Shanker, and E. Michielssen, "Fast time-domain characterization of finite size microstrip structures," *Int. J. Numer. Model. El.,* vol. 15, pp. 439-457, 2002.

[123] K. Aygun, B. C. Fischer, J. Meng, B. Shanker, and E. Michielssen, "A fast hybrid field-circuit simulator for transient analysis of microwave circuits," *IEEE Trans. Microw. Theory Tech.,* vol. 52, pp. 573-583, 2004.

[124] S. Velamparambil and W. C. Chew, "Analysis and performance of a distributed memory multilevel fast multipole algorithm," *IEEE Trans. Antennas Propag.,* vol. 53, pp. 2719-2727, 2005.

[125] S. Velamparambil, W. C. Chew, and J. Song, "10 million unknowns: is it that big?," *IEEE Trans. Antennas Propag.,* vol. 45, pp. 43-58, 2003.

[126] C. Waltz, K. Sertel, M. A. Carr, B. C. Usner, and J. L. Volakis, "Massively parallel fast multipole method solutions of large electromagnetic scattering problems," *IEEE Trans. Antennas Propag.,* vol. 55, pp. 1810-1816, 2007.

[127] J. Fostier and F. Olyslager, "Provably scalable parallel multilevel fast multipole algorithm," *Electron. Lett.,* vol. 44, pp. 1111-1113, 2008.

[128] 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, pp. 1740-1750, 2009.

[129] 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, pp. 23-34, 2010.

[130] O. Ergul and L. Gurel, "Rigorous solutions of electromagnetic problems involving hundreds of millions of unknowns," *IEEE Antennas Propag. Mag.,* vol. 53, pp. 18-27, 2011.

[131] V. Melapudi, B. Shanker, S. Seal, and S. Aluru, "A scalable parallel wideband MLFMA for efficient electromagnetic simulations on large scale clusters," *IEEE Trans. Antennas Propag.,* vol. 59, pp. 2565-2577, 2011.

[132] R. Jakob-Chien and B. K. Alpert, "A fast spherical filter with uniform resolution," *J. Comput. Phys.,* vol. 136, pp. 580-584, 1997.

[133] A. C. Yucel, "Helmholtz and high-frequency Maxwell multilevel fast multipole algorithms with self-tuning library," M.S. thesis, University of Michigan, Ann Arbor, 2008.

[134] J. Fostier and F. Olyslager, "An asynchronous parallel MLFMA for scattering at multiple dielectric objects," *IEEE Trans. Antennas Propag.,* vol. 56, pp. 2346-2355, 2008.

[135] S. H. Zainud-Deen, E. Hassan, M. S. Ibrahim, K. H. Awadalla, and A. Z. Botros, "Electromagnetic scattering using GPU-based finite difference frequency domain method," *Prog. Electromagn. Res.,* vol. 16, pp. 351-369, 2009.

[136] M. Cwikla, J. Aronsson, and V. Okhmatovski, "Low-frequency MLFMA on graphics processors," *IEEE Antennas Wireless Propag. Lett.,* vol. 9, pp. 8-11, 2010.

[137] S. Li, B. Livshitz, and V. Lomakin, "Fast evaluation of Helmholtz potential on graphics processing units (GPUs)," *J. Comput. Phys.,* vol. 229, pp. 8463-8483, 2010.

[138] S. Li, R. Chang, A. Boag, and V. Lomakin, "Fast electromagnetic integral-equation solvers on graphics processing units," *IEEE Trans. Magn.,* vol. 54, pp. 71-87, 2012.

[139] J. Guan, S. Yan, and J.-M. Jin, "An OpenMP-CUDA implementation of multilevel fast multipole algorithm for electromagnetic simulation on multi-GPU computing systems," *IEEE Trans. Antennas Propag.,* vol. 61, pp. 3607-3616, 2013.

[140] A. C. Yucel, L. J. Gomez, and E. Michielssen, "Compression of translation operator tensors in FMM-FFT accelerated SIE solvers via Tucker decomposition," *IEEE Antennas Wireless Propag. Lett.,* vol. early access, 2015.

[141] A. C. Yucel, "Uncertainty quantification for electromagnetic analysis via efficient collocation methods," Ph.D. thesis, University of Michigan, Ann Arbor, 2013.

[142] 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, pp. 1488-1493, 1997.

[143] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms I," *Comm. Pure Appl. Math.,* vol. 44, pp. 141-183, 1991.

[144] R. L. Wagner and C. Weng Cho, "A study of wavelets for the solution of electromagnetic integral equations," *IEEE Trans. Antennas Propag.,* vol. 43, pp. 802-810, 1995.

[145] W. L. Golik, "Wavelet packets for fast solution of electromagnetic integral equations," *IEEE Trans. Antennas Propag.,* vol. 46, pp. 618-624, 1998.

[146] H. Deng and H. Ling, "Fast solution of electromagnetic integral equations using adaptive wavelet packet transform," *IEEE Trans. Antennas Propag.,* vol. 47, pp. 674-682, 1999.

[147] G. W. Pan, Y. V. Tretiakov, and B. Gilbert, "Smooth local cosine based Galerkin method for scattering problems," *IEEE Trans. Antennas Propag.,* vol. 51, pp. 1177-1184, 2003.

[148] Z. Zhou and J. S. Tyo, "An adaptive time-domain integral equation method for transient analysis of wire scatterer," *IEEE Antennas Wirel. Propag. Lett.,* vol. 4, pp. 147-150, 2005.

[149] Y. Shifman and Y. Leviatan, "On the use of spatio-temporal wavelet expansions for transient analysis of wire antennas and scatterers," *IEEE Trans. Antennas Propag.,* vol. 52, pp. 2305-2318, 2004.

[150] R. R. Coifman and Y. Meyer, "Remarques sur l'analyse de Fourier à fenêtre," *Comptes rendus de l'Académie des sciences. Série 1, Mathématique,* vol. 312, pp. 259-261, 1991.

[151] C. Weng Cho, J. Jian-Ming, L. Cai-Cheng, E. Michielssen, and J. M. Song, "Fast solution methods in electromagnetics," *IEEE Trans. Antennas Propag.,* vol. 45, pp. 533-543, 1997.

[152] M. V. Wickerhauser, *Adapted wavelet analysis from theory to software*: A K Peters/CRC Press, 1994.

[153] K. Bittner and K. Gröchenig, "Direct and inverse approximation theorems for local trigonometric bases," *J. Approx. Theory,* vol. 117, pp. 74-102, 2002.

[154] R.-S. Wu, Y. Wang, and M. Luo, "Beamlet migration using local cosine basis," *Geophysics,* vol. 73, pp. S207-S217, 2008.

[155] D. H. Schaubert, D. R. Wilton, and A. W. Glisson, "A tetrahedral modeling method for electromagnetic scattering by arbitrarily shaped inhomogeneous dielectric bodies," *IEEE Trans. Antennas Propag.,* vol. 32, pp. 77-85, 1984.

[156] S. Bin Sayed, H. A. Ulku, and H. Bagci, "Transient analysis of electromagnetic wave interactions on high-contrast scatterers using volume electric field integral equation," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2014, pp. 751-752.

[157] A. Al-Jarro, M. A. Salem, H. Bagci, T. M. Benson, P. Sewell, and A. Vukovic, "Explicit solution of the time domain volume integral equation using a stable predictor-corrector scheme," *IEEE Trans. Antennas Propag.,* vol. 60, pp. 5203-5214, 2012.

[158] A. Al-Jarro, M. Cheeseman, and H. Bagci, "Distributed-memory parallelization of an explicit time-domain volume integral equation solver on Blue Gene/P," *Appl. Comp. Electromag. Soc. J.,* vol. 27, pp. 132-144, 2012.

[159] S. Feki, A. Al-Jarro, A. Clo, and H. Bagci, "Porting an explicit time-domain volume-integral-equation solver on GPUs with OpenACC," *IEEE Antennas Propag. Mag.,* vol. 56, pp. 265-277, 2014.

[160] J. Waldvogel, "The Newtonian Potential of Homogeneous Polyhedra," *J. Applied Math. Phys.,* vol. 30, pp. 388-398, 1979.

[161] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, "A scalable parallel PWTD-accelerated surface integral equation solver for analysis of transient scattering from large-scale objects," *IEEE Trans. Antennas Propag.,* (in review), 2015.

[162] Y. Liu, V. Lomakin, and E. Michielssen, "Graphics processing unit-accelerated implementation of the plane wave time domain algorithm," in *Proc. 28th Ann. Rev. Prog. Appl. Computat. Electromagn.*, 2012.

[163] Y. Liu, A. C. Yucel, V. Lomakin, and E. Michielssen, "A scalable parallel implementation of the plane wave time domain algorithm on graphics processing unit-augmented clusters," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2012.

[164] Y. Liu, H. Bagci, and E. Michielssen, "Progress in parallel implementation of the multilevel plane wave time domain algorithm," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2013.

[165] Y. Liu, H. Bagci, and E. Michielssen, "Solving very large scattering problems using a parallel PWTD-enhanced surface integral equation solver," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2013.

[166] Y. Liu, A. Al-Jarro, H. Bagci, and E. Michielssen, "Parallel, explicit, and PWTD-enhanced time domain volume integral equation solver," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2013.

[167] A. C. Yucel, Y. Liu, H. Bagci, and E. Michielssen, "A fast-multipole domain decomposition integral equation solver for characterizing electromagnetic wave propagation in mine environments," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2013.

[168] H. Guo, Y. Liu, H. Jun, and E. Michielssen, "A parallel MLMDA-based direct integral equation solver," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2013.

[169] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, "Parallel time domain solvers for electrically large transient scattering problems," in *Proc. EUCAP*, 2014.

[170] Y. Liu, A. C. Yucel, H. Bagci, and E. Michielssen, "A parallel wavelet-enhanced PWTD algorithm for analyzing transient scattering from electrically very large PEC targets," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2014.

[171] A. C. Yucel, L. Gomez, Y. Liu, H. Bagci, and E. Michielssen, "A FMM-FFT accelerated hybrid volume surface integral equation solver for electromagnetic analysis of re-entry space vehicles," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2014.

[172] A. C. Yucel, Y. Liu, H. Bagci, and E. Michielssen, "An FMM-FFT accelerated integral equation solver for characterizing electromagnetic wave propagation in mine tunnels and galleries loaded with conductors," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2014.

[173] Y. Liu, A. C. Yucel, A. C. G. H. Bagci, and E. Michielssen, "A wavelet-based PWTD algorithm-accelerated time domain surface integral equation solver," in *Proc. IEEE Int. Symp. AP-S/URSI*, 2015.

[174] Y. Liu, A. C. Yucel, V. Lomakin, and E. Michielssen, "Graphics processing unit implementation of multilevel plane-wave time-domain algorithm," *IEEE Antennas Wireless Propag. Lett.,* vol. 13, pp. 1671-1675, 2014.

[175] A. C. Yucel, Y. Liu, H. Bagci, and E. Michielssen, "Statistical characterization of electromagnetic wave propagation in mine environments," *IEEE Antennas Wireless Propag. Lett.,* vol. 12, pp. 1602-1605, 2013.