Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 227 (2007) 1007–1023

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Fast evaluation of time domain fields in sub-wavelength source/observer distributions using accelerated Cartesian expansions (ACE)

M. Vikram, B. Shanker *

Department of ECE, 2120 Engineering Building, Michigan State University, East Lansing, MI 48824, USA

Received 4 April 2007; received in revised form 13 August 2007; accepted 21 August 2007
Available online 1 September 2007

Abstract

Time domain integral equation solvers for transient scattering from electrically large objects have benefitted significantly from acceleration techniques like the plane wave time domain (PWTD) algorithm; these techniques reduce the asymptotic CPU and memory cost. However, PWTD breaks down when used in the analysis of structures that have sub-wavelength features or features whose length scales are orders of magnitude smaller than the smallest wavelength in the incident pulse. Instances of these occurring in electromagnetics range from antenna topologies, to feed structures, etc. In this regime, it is the geometric constraints that dictate the computational complexity, as opposed to the wavelength of interest. In this work, we present an approach for efficient analysis of such sub-wavelength source/observer distributions in time domain. The methodology that we seek to exploit is the recently developed algorithm based on Cartesian expansions for accelerating the computation of potentials of the form R^{ν}. In this paper, we present an efficient methodology for computing these polynomials for two different scenarios; where the size of the domain spans the distance travelled by light in (i) one time step and (ii) multiple time steps. These algorithms are cast within the framework of both uniform and non-uniform distributions. Results that demonstrate the efficiency and convergence of the proposed algorithm are presented.

PACS: 07.05.Tp

Keywords: Accelerated Cartesian expansion (ACE); Low-frequency; Sub-wavelength; Time domain; Retarded potentials; Fast algorithms

1. Introduction

Integral equation-based analysis of scattering from electrically large objects has been made possible via the development of acceleration techniques in both the frequency and time domain. In frequency domain, they are the fast multipole method (FMM) [1] and the adaptive integral method (AIM) [2], and their time domain

* Corresponding author. Tel.: +1 517 432 8136; fax: +1 517 353 1980.
E-mail address: bshanker@egr.msu.edu (B. Shanker).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.08.017

counterparts are the PWTD algorithm [3] and the time domain AIM [4]. These methods have ameliorated the computational cost when the size of the overall object is several wavelengths long and the smallest feature scale is a fraction of the wavelength. However, analysis of structures that contain a mix of feature scales, poses problems for both acceleration techniques. Here, it is the geometric constraint that dictate the computational complexity. For instance, to model fine features, it is necessary to discretize that domain at a considerably higher rate than that is dictated by the smallest wavelength to capture the geometric details, see Fig. 1. These features occur in the analysis of practical problems in applied electromagnetics, ranging from EMI/EMC applications to antenna topologies to feed structures to signal integrity analysis in high speed interconnects, etc. The solution to this problem is typically sought by devising a methodology that works at sub-wavelength scales, and developing a transition to higher frequencies so that it can be integrated with existing acceleration methodologies. As an aside, we note that modification of TDAIM for analyzing structures discretized at sub-wavelength scales exist [5], but it breaks down when the structure has multiple scales.

The problem encountered herein is not very different from those addressed in the frequency domain fast multipole method (FMM). FMM breaks down as the underlying kernel becomes unstable. Several solutions to this problem have been proposed, notably by [6–9], and many have been combined with classical FMM to yield robust and efficient methods for analyzing structures whose features span multiple scales [10,11,9]. The problems encountered in extending PWTD to analyzing sub-wavelength features is akin to those encountered in FMM. The PWTD algorithm is a time domain analogue of FMM, with one significant difference; the field due to a quasi-time limited and bandlimited source can be reconstructed to arbitrary accuracy using a discrete set of propagating plane waves provided certain separation conditions between the source and observers are met [12]. The separation criterion ensures that time gating can be employed to yield causal results. Unlike in the frequency domain, the cause of breakdown is not the expansions used in the algorithm; all functions used in the expansion are regular at zero. The breakdown occurs because domains that interact with each other via the PWTD algorithm are determined indirectly by the time step size. As the time step depends only on the maximum frequency of excitation and *not* on the smallest discretization, it implies that PWTD breaks down as an acceleration tool because most of the interactions would fall under the "near" field classification. However, these arguments suggest an approach for overcoming this hurdle; develop an acceleration procedure using adaptive time stepping. The main advantage of this procedure is the seamless manner in which it can be integrated with the classical PWTD scheme for high frequencies, resulting in an acceleration scheme that is valid at all length scales [13,14]. Alternatively, one can modify existing frequency domain low-frequency algorithm to construct time domain information [15]. This implies that one needs to develop the mechanism to transition from frequency to time domain and vice versa such that the resulting system can still be cast

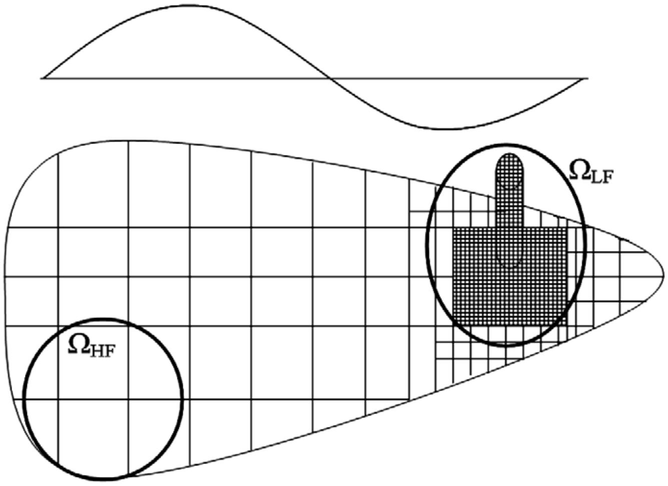

Fig. 1. Example of antenna feed geometry with low- and high-frequency regimes denoted by Ω_{LF} and Ω_{HF}, respectively. Smallest wavelength of incident pulse is also shown for reference.

within the framework that permits transient analysis within a marching-on-in-time framework. It has been shown that the latter approach is considerably faster than the former [15].

This paper presents an alternate method to arrive at the same objective and is founded on using Taylor expansions in a Cartesian framework. More specifically, the methodology presented herein will rely on the recently developed fast kernels for evaluating potential of the form R^v for $v \in \mathbb{R}$, and is very competitive in terms of speed for a given accuracy with the other two methods that exist [13–15]. Furthermore, it can be trivially extended to multiple time steps as well as integrated with PWTD. Thus, the main contribution in this work are

- Development of an acceleration technique to compute retarded potentials in the sub-wavelength regime. The method presented relies on representing the retarded potential as a function of potentials of the form R^v, and then accelerating this function. The presented method can be extended to other functional representations as well.
- Development of the requisite algorithmic structure to seamlessly extend this (with very little cost overhead) to multiple time steps. Extension to multiple time steps is done with the sole aim of integrating with the PWTD algorithm. The means to do so will be presented elsewhere.
- Development of techniques to further optimize the underlying kernels, and an algorithmic variation of the techniques presented in [16] for non-uniform geometries.

This paper is organized as follows: Section 2 formulates the problem, and casts it in terms of evaluation of potentials of the form R^v; followed by a brief description of necessary theorems of accelerated Cartesian expansion (ACE) method, for rapid evaluation of such potentials. Section 3 discuss in detail the application of ACE method in accelerating the evaluation of time domain fields from retarded potential and elucidates the steps necessary to implement this for multiple time steps. Finally, Section 4 presents plethora of results that demonstrate the efficiency of the proposed algorithm as well as its convergence. Conclusions of this study and proposed work are drawn in Section 5. Algorithms used here for non-uniform geometries and improvements to the kernel are presented in the Appendix A.

2. Preliminaries

2.1. Problem description

Consider a set of N_s sources that are randomly distributed in a domain Ω. The location of these sources will be denoted using $\mathbf{r}_n$ and their time signatures by $f_n(\mathbf{r}_n,t)$ for $n = 1, \ldots, N_s$. It is assumed that these functions are bandlimited to an angular frequency $\omega_{\max}$ and all sources are approximately quiescent for $t < 0$. As in all time domain solvers, the source functions $f_n(\mathbf{r}_n,t)$ are known only at evenly spaced time steps $t_k = k\Delta_t$ for $k = 1, \ldots, N_t$ where $\Delta_t = \pi/(\chi\omega_{\max})$, $N_t\Delta_t$ is the total simulation time and χ is an oversampling factor. Typically, $\chi > 1$ and chosen between 5 to 20 to accurately reconstruct functions $f_n(\mathbf{r}_n,t)$ from its samples. The field at any point $\mathbf{r}$ due to these sources is given by

$$\Phi(\mathbf{r},t) = \sum_{n=1}^{N_s} \frac{\delta(t - R_n/c)}{R_n} \star f_n(\mathbf{r}_n,t) \tag{1}$$

where c is the speed of light, $\star$ denotes convolution in time and $R_n = \|\mathbf{r} - \mathbf{r}_n\|$. It is apparent that the cost of computing (1) scales as $\mathcal{O}(N_t N_s^2)$. Finally, in keeping with the definition of sub-wavelength regime, we will assume that the $\mathrm{diam}(\Omega) = \mathcal{O}(c\Delta_t)$. Given the size of the domain, it is apparent that the PWTD scheme cannot be readily used; it has to be substantially modified in order to evaluate these potentials efficiently [14].

In developing this scheme, it is necessary that the source signatures in (1) be known so as to facilitate the integration of the proposed algorithm with existing marching-on-in-time solvers for time domain integral equations. The starting point of the proposed method arises from the representation of the source signal. We will assume that the source function can be represented in terms of $f_n(\mathbf{r}_n,t) = \sum_k I_{nk} T_k(t)$, where $T_k(t) = T(t - t_k)$ is a time basis function and I_k are the samples of the function at the discrete time step t_k. It follows from this representation that

$$\Phi(\mathbf{r},t)=\sum_{n=1}^{N_s}\sum_{k=0}^{N_t}I_{n,k}\frac{T_k(t-R_n/c)}{R_n} \tag{2}$$

This implies that to realize a fast algorithm, one needs to rapidly compute functions of the form $T_k(t-R_n/c)$. To illustrate the development of a fast algorithm, we will assume that the temporal basis functions are backward Lagrange polynomials. Note, however, that the methodology presented herein is *not* restricted to polynomials. To this end, we will define Kth order basis functions of the form

$$T(t)=\begin{cases} h_k(t)g_{K-k}(t) & \text{for } (k-1)\Delta_t\leqslant t\leqslant k\Delta_t;\quad k=0,\ldots,K \\ 0 & \text{otherwise}\end{cases} \tag{3a}$$

where

$$h_k(t)=\begin{cases} 1 & k=0 \\ \prod_{i=1}^{k}\frac{t-i\Delta_t}{-i\Delta_t} & k\neq 0\end{cases} \tag{3b}$$

and

$$g_{K-k}(t)=\prod_{i=1}^{K-k}\frac{t+i\Delta_t}{i\Delta_t} \tag{3c}$$

It follows, from the above equation that $T(t)=0$ for $t\notin(-\Delta_t,K\Delta_t)$, $T(0)=1$ and $T(t)=0$ for $t=-\Delta_t,\Delta_t,2\Delta_t,\ldots,(K-1)\Delta_t$. Using the functions in Eq. (2), and point testing in time, results in the potential function that is a polynomial of R_n. It means that one can directly exploit acceleration methods developed for kernels of the form R^v [17].

2.2. Accelerated Cartesian expansions (ACE)

The mathematical engine behind the fast method developed in this work relies on accelerated Cartesian expansions (ACE) [17]. This method uses a Taylor's series expansion to create addition theorems in terms of Cartesian tensors, exploits the fact that these tensors are totally symmetric and derives an exact algorithm for traversing up and down the tree. The proposed methodology is applicable to a wide range of non-oscillatory kernels with little change in the overall algorithmic structure. While this technique was introduced for kernels of the form R^v, the proposed technique has been extended to frequency domain sub-wavelength kernels [9], Yukawa (or shielded Coulomb) potentials and Gauss transforms. While similar methodologies have been introduced earlier [18,19], they are either not generalizable or offer only some of the advantages of this scheme. In what follows, a brief overview of ACE algorithm and the relevant definitions and theorems are presented.

Tensor analysis is an integral tool used in development of ACE algorithm. A Cartesian tensor of rank n is denoted by $\mathbf{A}^{(n)}$ or in component notation by $A^{(n)}_{\alpha_1\ldots\alpha_n}$, and is an array of 3^n components, for points in $\mathbb{R}^3$. A totally symmetric tensor is one that is independent of the permutation of indices $\alpha_1\ldots\alpha_n$ and in compressed form it contains $(n+1)(n+2)/2$ independent components. Alternatively, they can be represented in compressed form as $A^{(n)}(n_1,n_2,n_3)$ where $n_1+n_2+n_3=n$, and n_i is the number of times the index i is repeated. An n-fold contraction between two tensors $\mathbf{A}^{(n+m)}$ and $\mathbf{B}^{(n)}$ is represented using $\mathbf{C}^{(m)}=\mathbf{A}^{(n+m)}\cdot n\cdot\mathbf{B}^{(n)}$. An extensive exposition of theorems and formulae pertinent to the properties of compressed tensors, their application to the ACE algorithm, asymptotic cost scaling, etc. can be found in [17].

Next, we will briefly outline the theorems that permit the fast evaluation of functions. To this end, assume that domains Ω_s and Ω_o are sufficiently separated, and comprise of sources and observers, respectively. Also, $\Omega_s\subset\Omega^p_s$, $\Omega_o\subset\Omega^p_o$ and $\Omega^p_s\cap\Omega^p_o=\emptyset$. The centers of the domains Ω_s, Ω_o, Ω^p_s and Ω^p_o are denoted by $\mathbf{r}_s$, $\mathbf{r}_o$, $\mathbf{r}^p_s$ and $\mathbf{r}^p_o$, respectively. Further, denote the potential function that maps the effects of these sources on the observation points as $\psi(R)$, where $R=\|\mathbf{r}-\mathbf{r}'\|$ and k sources exist in Ω_s. Here, the function $\psi(R)$ can stand for any interpolation function $T(t)$ convolved with the retarded potential and observed at time $t=0$. An addition theorem for this function may be obtained using Taylor's expansion.

Theorem 2.1 (Taylor expansion). *The function $\psi(\mathbf{r}-\mathbf{r}')$ can be expressed about the origin using*

$$\psi(\mathbf{r}-\mathbf{r}') = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \mathbf{r}'^n \cdot n \cdot \nabla^n \psi(\mathbf{r}) \tag{4}$$

where $\mathbf{r} > \mathbf{r}'$

This theorem gives rise to the following corollary.

Corollary 2.2. *The function $\psi(\mathbf{r}-\mathbf{r}')$ takes the form*

$$\psi(\mathbf{r}-\mathbf{r}\prime) = \begin{cases} \sum\limits_{n=0}^{\infty} \mathbf{M}^{(n)} \cdot n \cdot \nabla^n \psi(\mathbf{r}) & \text{for } \mathbf{r} > \mathbf{r}' \\ \sum\limits_{n=0}^{\infty} \mathbf{r}'^n \cdot n \cdot \mathbf{L}^{(n)} & \text{for } \mathbf{r}' > \mathbf{r} \end{cases} \tag{5}$$

where $\mathbf{M}^{(n)}$ and $\mathbf{L}^{(n)}$ are the multipole and local expansions. These theorems may be used in concert to derive/prove the following five theorems that form the crux of ACE [17].

Theorem 2.3 (Multipole expansion). *The total potential at any point $\mathbf{r} \in \Omega_o$ due to k sources q_i, $i = 1, \ldots, k$ located at points $\mathbf{r}_i \in \Omega_s$ is given as*

$$\begin{aligned} \psi(\mathbf{r}) &= \sum_{n=0}^{\infty} \mathbf{M}^{(n)} \cdot n \cdot \nabla^n \psi(\mathbf{r}) \\ \mathbf{M}^{(n)} &= \sum_{i=1}^{k} (-1)^n \frac{q_i}{n!} (\mathbf{r}_i - \mathbf{r}_\mathrm{s})^n \end{aligned} \tag{6}$$

where $\mathbf{M}^{(n)}$ is the multipole tensor.

Theorem 2.4 (Multipole-to-multipole expansion). *Given a multipole expansion of k sources about $\mathbf{r}_s$*

$$\mathbf{O}^{(n)} = \sum_{i=1}^{k} (-1)^n \frac{q_i}{n!} (\mathbf{r}_i - \mathbf{r}_\mathrm{s})^n \tag{7a}$$

then the multipole expansion about the point $\mathbf{r}_\mathrm{s}^p$ can be expressed in terms of (7a) *as*

$$\mathbf{M}^{(n)} = \sum_{i=1}^{k} (-1)^n \frac{q_i}{n!} (\mathbf{r}_i - \mathbf{r}_\mathrm{s}^p)^n = \sum_{m=0}^{n} \sum_{P(m,n)} \frac{m!}{n!} (\mathbf{r}_\mathrm{s}^p - \mathbf{r}_\mathrm{s})^{n-m} \mathbf{O}^{(m)} \tag{7b}$$

It is evident that one can repeatedly use this theorem to translate the multipole expansion from $\mathbf{r}_s$ to $\mathbf{r}_\mathrm{s}^p$. This expression is exact [17].

Theorem 2.5 (Multipole-to-local translation). *Assume that the domains Ω_s^p and Ω_o^p are sufficiently separated, and the distance between their centers $r_{os}^p = |\mathbf{r}_{os}^p| = |\mathbf{r}_\mathrm{o}^p - \mathbf{r}_\mathrm{s}^p|$ is greater that diam$\{\Omega_\mathrm{s}^p\}$ and diam$\{\Omega_\mathrm{o}^p\}$. If a multipole expansion $\mathbf{M}^{(n)}$ is located at $\mathbf{r}_\mathrm{s}^p$, then another expansion $\mathbf{L}^{(n)}$ that produces the same field $\forall \mathbf{r} \in \Omega_\mathrm{o}^p$ is given by*

$$\begin{aligned} \psi(\mathbf{r}) &= \sum_{n=0}^{\infty} \boldsymbol{\rho}^n \cdot n \cdot \mathbf{L}^{(n)} \\ \mathbf{L}^{(n)} &= \sum_{m=n}^{\infty} \frac{1}{n!} \mathbf{M}^{(m-n)} \cdot (m-n) \cdot \tilde{\nabla}^m \psi(r_{os}^p) \end{aligned} \tag{8}$$

where $\boldsymbol{\rho} = \mathbf{r} - \mathbf{r}_\mathrm{o}^p$ and $\tilde{\nabla}$ is the derivative with respect to $\mathbf{r}_\mathrm{s}^p$.

Theorem 2.6 (Local-to-local expansion). *A local expansion $\mathbf{O}^{(n)}$ that exists in the domain Ω_o^p centered around $\mathbf{r}_\mathrm{o}^p$ can be shifted to the domain Ω_o centered at $\mathbf{r}_o$ using*

$$\mathbf{L}^{(n)} = \sum_{m=n}^{\infty} \begin{pmatrix} m \\ m-n \end{pmatrix} \mathbf{O}^{(m)} \cdot (m-n) \cdot (\mathbf{r}_o^{cp})^{m-n} \tag{9}$$

It can be shown that this expression is exact as well. Finally, the fields at a set of observation points can be computed using the following theorem

$$\psi(\mathbf{r}) = \sum_{n=0}^{\infty} \mathbf{L}^{(n)} \cdot n \cdot (\boldsymbol{\rho}_{oi})^{n} \tag{10}$$

Proofs for these theorems for $\psi(R) = R^{v}$ can be found in [17] and may be trivially extended to functions of the form $\psi(R) = span\{R^{-v}\}$ for $v = -1,0,1,\ldots,K$, or any other non-oscillatory function. Note, that when $\psi(R) = R^{-v}$, evaluating the multipole-to-local expansion using Theorem 2.5 implies the computation of $\nabla^{n} R^{-v}$ which can be efficiently effected through

$$\partial_i^{n_1}\partial_j^{n_2}\partial_k^{n_3}\left(\frac{1}{R^{v}}\right) = (-1)^{n} R^{-2n-v} \sum_{m_1=0}^{\lfloor\frac{n_1}{2}\rfloor}\sum_{m_2=0}^{\lfloor\frac{n_2}{2}\rfloor}\sum_{m_3=0}^{\lfloor\frac{n_3}{2}\rfloor}(-1)^{m}\begin{bmatrix} n_1 \\ m_1 \end{bmatrix}\begin{bmatrix} n_2 \\ m_2 \end{bmatrix}\begin{bmatrix} n_3 \\ m_3 \end{bmatrix} \times R^{2m} f(v, n-m-1) x^{n_1-2m_1} y^{n_2-2m_2} z^{n_3-2m_3} \tag{11}$$

where $R^2 = x^2 + y^2 + z^2$. As was pointed out in [17], a computation scheme based on these theorems have the following characteristics:

(1) The multipoles are independent of the function being translated. Only the translation operator depends on v. This fact will be of use in developing fast methods for evaluating the retarded potential.
(2) The multipole-to-multipole expansion (or the local-to-local expansion) is exact. This implies that the errors obtained do not depend on the height of the tree.
(3) The formulation in terms of totally symmetric tensors permits the realization of CPU cost savings of a factor of 1/720 over a simplistic implementation.
(4) Finally, since only the translation function depends on the potential function being used, it follows that the proposed methodology can be readily altered, with very little change in the overall algorithm, for other potential functions.

These theorems permit rapid evaluation of potential using either a standard or compressed oct-tree decomposition of the domain. A standard oct-tree is constructed by first embedding the entire domain in a fictitious cube that is then divided into eight sub-cubes and so on. This process continues recursively until the desired level of refinement is reached; an N_l-level scheme implies $N_l - 1$ recursive divisions of the domain. At any level, the domain that is being partitioned is called the *parent* of all the eight *children* that it is being partitioned into. At the lowest level, all source/observers are mapped onto the smallest boxes, *leaf boxes*. This hierarchical partitioning of the domain is referred to as a regular oct-tree data structure. The interactions between all source and observation points are now computed using traversal up and down the tree structure. At any level in the tree, all boxes/domains are classified as being either in the near or far-field of each other using the following dictum: two subdomains are classified as being in the far-field of each other if the distance between the centers is at least twice the sidelength of the domain, *and* their parents are in the near-field of each other. This definition will be used *unless* it is specially stated that an alternate definition is necessary. Appendix A.1 briefly delineates the methods used to efficiently create both the uniform and non-uniform oct-tree as well as interaction lists.

Next, we will describe modifications to this computational structure to accommodate interactions that occur within a time step and those that extend across multiple time steps.

3. Low-frequency time domain acceleration scheme

This section presents the necessary framework for rapidly computing retarded potentials (in the low-frequency regime) in two steps: (i) when the entire domain diam(Ω) $\leqslant c\Delta_t$ and (ii) when diam(Ω) $> c\Delta_t$.

3.1. Interaction within one time step

The field at any observation point $\mathbf{r} \in \Omega$, at time instance $i\Delta_t$, due to sources at $\mathbf{r}_n \in \Omega$ for $n = 1, \ldots, N_s$ can be obtained from Eq. (1)

$$\Phi(i\Delta_t, \mathbf{r}) = \sum_{n=1}^{N_s} \int_0^{\Delta_t} \frac{\delta(\tau - R_n/c)}{R_n} f_n(\mathbf{r}_n, i\Delta_t - \tau)\mathrm{d}\tau \tag{12}$$

where $R_n = \|\mathbf{r} - \mathbf{r}_n\|$and $f_n(\mathbf{r}_n,t)$ is the transient source strength at the nth spatial point. The limits $[0,\Delta_t]$, on above time integral is possible because $R_n/c \in [0,\Delta_t]$. Employing time domain basis function from Eq. (2) and evaluating the time integral in Eq. (12) results in

$$\Phi(i\Delta_t, \mathbf{r}) = \sum_{n=1}^{N_s} \sum_{j=i-K}^{i} I_{n,j} \frac{T((i-j)\Delta_t - R_n/c)}{R_n} \tag{13}$$

$$= \sum_{n=1}^{N_s} \sum_{j=0}^{K} I_{n,i-j} \frac{T(j\Delta_t - R_n/c)}{R_n} \tag{14}$$

where K is the order of temporal basis function $T(t)$. Since $T(t)$ is chosen to be a backward Lagrange polynomial, Eq. (13) can be expressed in terms of powers of R_n/c as

$$\Phi(i\Delta_t, \mathbf{r}) = \sum_{n=1}^{N_s} \sum_{j=0}^{K} \sum_{h=0}^{K} I_{n,i-j} \alpha(h,j) R_n^{h-1} \tag{15}$$

In Eq. (15), $\alpha(h,j)$ is the coefficient corresponding to the polynomial of degree $(h - 1)$ for the basis function at $(i - j)$-th time step, they also depend on Δ_t and c. Evaluating these polynomials of form R^v can be performed at $\mathcal{O}(N_s)$ cost using the ACE algorithm detailed in Section 2.2. Thus, the overall cost of this scheme scales as $\mathcal{O}(KN_s)$. Error bounds for using ACE to evaluate Eq. (15) can be obtained from the bounds derived in [17] and it can be proven that the upper bound of the error is determined by that for R^{-1}. Note, that the above derivation is not specific to using polynomials as temporal basis functions. Other basis functions may be dealt with in one of two ways; either by finding the appropriate translation functions, or by mapping these onto a space of polynomials. Using polynomials is fairly trivial as the framework for the R^v kernel is readily available [17].

The $\mathcal{O}(KN_s)$ reduction in cost, specified above, is for brute force implementation of the ACE algorithm. It is important to recognize that the above formulation demands evaluation of the kernel R^{-v} for different v's. However, most of the steps in the proposed algorithm are kernel independent. In that, Theorems 2.3 and 2.4 (multipole expansion and multipole-to-multipole translation) do not depend on the kernel. Similar observation holds for local-to-local translation and evaluation of potential from local expansion, Theorem 2.6 and Eq. (10). Thus, only the multipole-to-local translation, Theorem 2.5, depends on the kernel and requires the evaluation of $\nabla^n R^v$ for different v values. Therefore, evaluation of polynomials of form $\sum_v c_v R^v$ involves (almost) one tree traversal (up and down) irrespective of the kernel, only the multipole-to-local translations need to be done separately for each kernel or polynomials of different degrees. Thus, a careful implementation of the ACE algorithm results in an adaptable and significantly lower cost algorithm. Applying the multipole-to-local translation (Theorem 2.5) in Eq. (15) we get

$$\begin{aligned}\Phi(i\Delta_t, \mathbf{r}) &= \sum_{n=1}^{N_s} \sum_{j=0}^{K} \sum_{h=0}^{K} I_{n,i-j} \alpha(h,j) R_n^{h-1} = \sum_{n=1}^{N_s} \sum_{j=0}^{K} \sum_{h=0}^{K} I_{n,i-j} \alpha(h,j) \sum_{p=0}^{P} \nabla^p R_{\mathrm{o}}^{(h-1)} \cdot p \cdot R_n'^{(p)} \\ &= \sum_{p=0}^{P} \sum_{j=0}^{K} \left(\sum_{h=0}^{K} \alpha(h,j) \nabla^p R_{\mathrm{o}}^{(h-1)} \right) \cdot p \cdot \left(\sum_{n=1}^{N_s} I_{n,i-j} R_n'^{(p)} \right) = \sum_{j=0}^{K} \sum_{p=0}^{P} \mathcal{T}_j^{(p)} \cdot p \cdot \mathcal{M}_j^{(p)}\end{aligned} \tag{16}$$

where $R_{\mathrm{o}} = \|\mathbf{r} - \mathbf{r}_{\mathrm{o}}\|$, $R_n' = \|\mathbf{r}_{\mathrm{o}} - \mathbf{r}_n\|$ and $\mathbf{r}_{\mathrm{o}}$ is the center of sphere enclosing all sources. $\mathcal{T}_j^{(p)}$ and $\mathcal{M}_j^{(p)}$ are the optimal tensor representation of multipoles and translation operation of the ACE algorithm. Eq. (16) implies that upward tree traversal, i.e. multipole-to-multipole translation and multipole-to-local translation should be performed K times. This is to preserve the transient information, $I_{n,i-j}$ associated with each basis function for every source. However, downward tree traversal which include local-to-local translation and potential evaluation needs to be performed only once.

3.2. Multiple time step interaction

The above exposition was geared towards developing a scheme for computing interactions when $\mathrm{diam}(\Omega) < c\Delta_t$. Next, we will prescribe modifications to the methodology when $\mathrm{diam}(\Omega) > c\Delta_t$. Consider two domains Ω_1 and Ω_2 such that, $\forall \mathbf{r}_1 \in \Omega_1$ and $\mathbf{r}_2 \in \Omega_2$ satisfies $(N-1)\Delta_t \leqslant \|\mathbf{r}_1 - \mathbf{r}_2\|/c \leqslant N\Delta_t$, where N is any positive integer. Then, the field at any point $\mathbf{r}_1$ at ith time step, $\Phi(i\Delta_t, \mathbf{r}_1)$, due to N_{s,Ω_2} sources at $\mathbf{r}_n \in \Omega_2$ can be written as

$$\Phi(i\Delta_t,\mathbf{r}_1) = \sum_{n=1}^{N_{s,\Omega_2}} \int_{(N-1)\Delta_t}^{N\Delta_t} \frac{\delta(\tau - R_n/c)}{R_n} f_n(i\Delta_t - \tau, \mathbf{r}_n)\mathrm{d}\mathbf{r}\,\mathrm{d}\tau \tag{17}$$

where $R_n = \|\mathbf{r}_1 - \mathbf{r}_n\|$. Repeating the derivation presented for single time step interaction, we get

$$\Phi(i\Delta_t,\mathbf{r}_1) = \sum_{n=1}^{N_s} \sum_{j=0}^{K} I_{n,i-j-(N-1)} \frac{T((j+N-1)\Delta_t - R_n/c)}{R_n} \tag{18}$$

When $N = 1$, (18) reduces to the case for interaction within one time step (13). It is important to preserve R/c argument of the basis function in (18), as a polynomial representation is necessary for acceleration using the ACE algorithm. Thus, the key in multiple time step interaction is to identify groups Ω_1 and Ω_2, and it can be done using the following argument,

$$\text{find } d_{\min} \geqslant N\Delta_t \quad \text{and} \quad d_{\max} \leqslant (N+1)\Delta_t \tag{19}$$

where $d_{\max}$ and $d_{\min}$ are the maximum and minimum distance between any two points in Ω_1 and Ω_2, see Fig. 2. For example, consider spherical domains of radii r_1 and r_2 whose centers are separated by R_{o}; then $d_{\max} = -R_{\mathrm{o}} + r_1 + r_2$ and $d_{\min} = R_{\mathrm{o}} - r_1 - r_2$. From Eqs. (18) and (16) it can be inferred that the number of upward tree traversals (multipole-to-multipole and multipole-to-local translations) equals $N_{\max}K$, where $N_{\max}c\Delta_t$ is the diameter of the sphere encompassing the entire low-frequency region Ω. These constraints mandate a new definition be used when developing interaction lists in the oct-tree as follows:

Definition 1 (*Interaction list rule*). Consider two child boxes whose parent boxes are in near-field. They are in each other's far-field if the distance between their centers is at least twice the sidelength of the domain *and* they satisfy Eq. (19). Otherwise, they are in each other's near-field.

Some boxes may be well-separated in space and still not satisfy the temporal constraint in Eq. (19). For example, consider two spheres of radius $r_1 = r_2 = c\Delta_t/8$ whose centers are separated by $R_{\mathrm{o}} = Nc\Delta_t$, now $d_{\max} = c(N+1/2)\Delta_t$ and $d_{\min} = c(N-1/4)\Delta_t$ which do not satisfy Eq. (19). In such cases, one can choose either of the following options: (i) sub-divide the domains and perform interaction at next level (with smaller domain size); and (ii) consider the domains to be in near-field of each other and use direct evaluation. Sub-dividing the domain without limit has two disadvantages. First, the number of unknowns per smallest box, with increasing levels, can fall below the limit for optimal computational cost. Second, sub-division into smal-

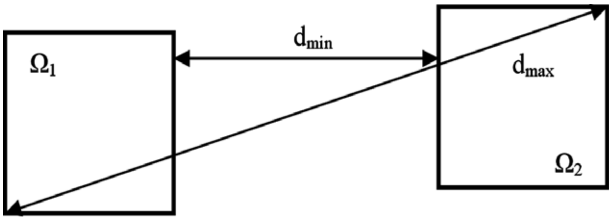

Fig. 2. Definition for domains interacting over multiple time steps.

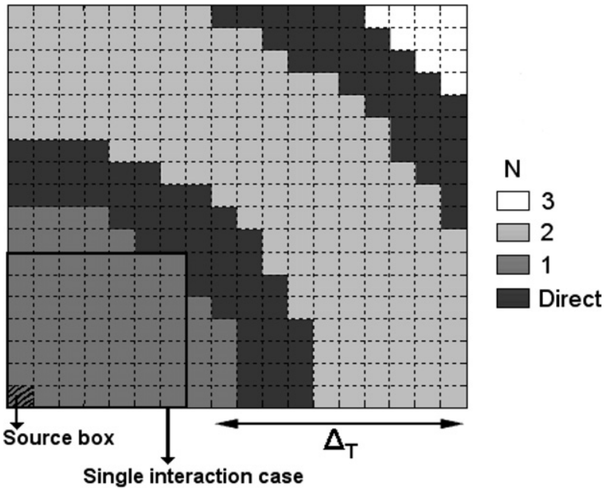

Fig. 3. Map of N in Eq. (19) for an example single level interaction.

ler size boxes does not *always* ensure compliance with constraint in Eq. (19); it can be shown that boxes who's centers are separated by multiples of $c\Delta_t$ ($Nc\Delta_t$), without regard to their size, will not follow the temporal constraint (19) and interaction between such boxes should be evaluated using direct methods, see Fig. 3. Further, using the second option on short trees can increase the total number of near-field interactions and dominate the overall computational cost. In this work, we consider an optimal implementation by combining both, i.e. sub-dividing up to a certain level and beyond this level domains violating Eq. (19) are placed in near-field interaction of each other. It is essential to note that the number of levels up to which sub-division is used can be geometry dependent. In essence, this procedure overcomes the multiple time step interaction with a slight cost overhead that should be optimized. Further observations are presented in next section.

4. Results

In this section, we present results that will substantiate the above claims and demonstrate the efficacy of the algorithm presented herein. As in all illustration of FMM methods, the goal is to demonstrate considerable speed-up with predetermined accuracy. Consequently, the results presented will demonstrate convergence as well as $\mathcal{O}(N_s)$ per time step CPU cost scaling. In all numerical experiments, the source/observer locations are randomly distributed. The corresponding standard/compressed oct-tree data structures (including interaction lists) are generated using the algorithmic procedure outlined in the Appendix A. The accuracy of the proposed algorithm is validated against analytical data for all cases where the unknown count is numerically small. The relative error at nth observer is evaluated as

$$\text{Error}_{\text{far}}(n) = \frac{||\Phi_{\text{fast,far}}(n,t) - \Phi_{\text{analytical,far}}(n,t)||_2}{||\Phi_{\text{analytical,far}}(n,t)||_2} \tag{20}$$

where $\|\cdot\|_2$ represents L_2-norm, $\Phi_{\text{fast, far}}(t)$ and $\Phi_{\text{analytical, far}}(t)$ represent the time history of the fields produced by the sources evaluated using proposed algorithm and analytical procedure, respectively. The error reported in this work is the average error over all observers [14] when the number of observers $N_s < 32{,}000$. For larger number of unknowns, the analytical data (and hence the error) is computed for randomly distributed unknowns (approximately 150). Hence, the reported data is an estimate of the expected error. These values are denoted using a †. Finally, as is usually done for all fast algorithms, analytical data is computed *only* for the source/observation pairs that are in the far-field of each other, and is consequently representative of an upper bound or worst-case error. The CPU timings (in seconds) are those taken for evaluating the field at a single time step using a 2.3 GHz Intel Pentium processor with 2 GB RAM running Linux OS. In all experiments that follow, the time signature that is associated with the nth source is given by Eq. (21)

$$f_n(t,\mathbf{r}_n) = \kappa_n \mathrm{e}^{-(t-t_p)^2/2\sigma^2} \tag{21}$$

where κ_n is the magnitude of the source randomly chosen between $[0,1]$, $\sigma = 6.366 \times 10^{-8}$ s and $t_p = 6\sigma$ s. The effective highest frequency and minimum wavelength associated with these signal parameters are $f_{\max} = 3/\pi\sigma = 15$ MHz and $\lambda_{\min} \approx 20$ m, respectively. As prescribed in MOT solvers, the time step is chosen as $\Delta_t = 1/(20 f_{\max}) = 3.334$ ns and is independent of geometric feature size and only a function of $f_{\max}$. The above parameters are chosen such that $c\Delta_t = 1$ m, thus, all geometric features smaller than 1 m would fall in the sub-wavelength category. In rest of the section, P denotes the number of ACE harmonics used and K denotes the order of the time basis function.

The first set of numerical simulation is performed to demonstrate the validity of the improvements made in the kernel that reduce the number of translations by approximately a factor of two without significantly affecting the order of the error (see Appendix A.1 for details). The numerical experiment performed is as follows; source points were randomly distributed within a cube of sidelength 0.5 m, i.e. all points interact within one time step. The number of source/observation points is varied (as is the height of the tree), the number of unknowns at a leaf box is approximately 64, and error is obtained for the "Old" and "New" schemes. The results presented in Table 1 indicate what is expected, viz., the computational cost is reduced approximately by a factor of two while the increase in error is almost always marginal (the order of magnitude of the error is unchanged).

Next, we demonstrate that the multipole-to-multipole and local-to-local operations are exact. An important ramification of this is that the error is *independent* of the height of the tree. This experiment is effected as follows: consider two cubical domains $\Omega_1 = (0,1/4)\times(0,1/4)\times(0,1/4)$ m^3 and $\Omega_2 = (1/2,3/4)\times(0,1/4)\times(0,1/4)$ m^3. Each domain contains 4000 randomly distributed source and observation points. In constructing interaction lists, it is ensured that *only* sources/observers in Ω_1 and Ω_2 interact, all others are ignored. Thus, as the number of levels in the tree are increased, the change in the error norm can be attributed solely to the multipole-to-multipole and local-to-local operations. Table 2 shows error computed for different $\{P,K\}$ pairs and different levels in tree, where dx_0 is the size (in meters) of smallest box. It is evident from Table 2 that, for a given $\{P,K\}$ pair, the variation in error obtained from using different levels in the tree is accurate to double precision. This is a consequence of the fact that Theorems 2.4 and 2.6 are exact, i.e. they produce the multipole (or local) expansion had the box size at that level been the leaf box. Consequently, the error bounds are much tighter. Details and proofs can be found in [17].

Next, results are presented for distribution wherein all source/observation pairs are distributed within a domain $\Omega < c\Delta_t$ and distribution sizes ranging from 8000 to 4,000,000 points. The number of unknowns per leaf box, on average, is chosen to lie between 60 and 70. From Table 1, it can be inferred that number of harmonics and order of time basis function are closely coupled, i.e. for a given K, arbitrarily increasing P does not improve the error and vice versa. This is true because the two sources for error (20) reported here are (a) approximation of a time signal with polynomial basis function of order K and (b) error in evaluating a

Table 1
Comparison between old and new (reduced) scheme for interaction list for different distribution sizes (N_s) and $\{P,K\}$ pairs

N_s	$\{P,K\}$	Old$_{\text{error}}$	New$_{\text{error}}$	Old$_{\text{time}}$	New$_{\text{time}}$	Old$_{\text{time}}$/New$_{\text{time}}$
12,000	{3,1}	3.40E−4	5.96E−4	0.21	0.11	1.96
12,000	{4,2}	6.94E−5	1.43E−4	0.54	0.23	2.4
12,000	{5,2}	2.73E−5	4.87E−5	1.00	0.42	2.37
12,000	{9,3}	2.37E−6	4.52E−6	9.49	3.6	2.63
12,000	{13,3}	8.00E−7	1.62E−6	53.47	18.5	2.89
32,000	{3,1}	2.36E−4	3.55E−4	0.67	0.38	1.75
32,000	{4,2}	3.61E−5	6.54E−5	1.97	0.91	2.16
32,000	{5,2}	1.70E−5	2.59E−5	3.65	1.68	2.17
32,000	{9,3}	1.38E−6	2.31E−6	28.54	14.42	1.98
64,000	{3,1}	2.86E−4	6.38E−4	1.56	0.79	1.98
64,000	{4,2}	6.42E−5	1.74E−4	3.94	1.9	2.07
64,000	{5,2}	1.82E−5	4.01E−5	7.17	3.18	2.26
64,000	{9,3}	4.02E−6	8.00E−6	66.02	27.95	2.36

Table 2
Exact multipole-to-multipole and local-to-local operators of ACE

dx_0	Levels	$\{P,K\}$		
$A = 0.0625$		$\{1,1\}$	$\{2,1\}$	$\{4,2\}$
A	4	1.8800972191556 69E−2	5.514103752372 495E−3	7.03463843261 4828E−4
$A/2$	5	1.8800972191556 66E−2	5.514103752372 538E−3	7.03463843261 3739E−4
$A/8$	7	1.8800972191556 70E−2	5.514103752372 537E−3	7.03463843261 3831E−4
$A/32$	9	1.8800972191556 70E−2	5.514103752372 536E−3	7.03463843261 3819E−4

Table 3
$\text{Error}_{\text{far}}$ in single time step interaction case ($C_s = 0.5$), for various N_s and $\{P,K\}$ pairs

N_s	Levels	$\text{Error}_{\text{far}}$, $\{P,K\}$						
		$\{1,1\}$	$\{2,1\}$	$\{3,1\}$	$\{4,2\}$	$\{5,2\}$	$\{9,3\}$	$\{13,3\}$
8000	4	5.81E−3	9.38E−4	3.41E−4	7.87E−5	2.09E−5	4.97E−6	8.85E−7
12,000	4	9.95E−3	1.55E−3	5.96E−4	1.43E−4	4.87E−5	4.52E−6	1.62E−6
32,000	4	4.74E−3	6.37E−4	3.55E−4	6.54E−5	2.59E−5	2.31E−6	1.03E−6
64,000†	5	8.08E−3	9.44E−4	6.38E−4	1.74E−4	4.01E−5	8.00E−6	2.43E−6
500,000†	6	1.50E−3	1.76E−3	1.18E−3	4.06E−4	2.48E−5	8.34E−6	–

Table 4
Comparison of run-time in single time step interaction case ($C_s = 0.5$)

N_s	T_{Direct}	T_{fast},$\{P,K\}$				
		$\{1,1\}$	$\{2,1\}$	$\{4,2\}$	$\{9,3\}$	$\{13,3\}$
8000	4.47	1.40E−2	3.18E−2	0.14	2.17	10.96
12,000	11.02	2.27E−2	4.61E−2	0.23	3.60	18.50
32,000	97.59	8.87E−2	0.18	0.91	14.42	85.2
64,000	–	0.20	0.44	1.90	27.95	173.38
500,000	–	1.94	3.82	15.67	245.37	–
1,000,000	–	3.78	7.19	30.98	498.03	–
2,000,000	–	7.71	13.33	60.18	742.21	–
4,000,000	–	16.06	27.46	121.72	1940.15	–

polynomial through ACE (limited P) due to far-field approximation. Hence, the results for time comparison are presented only for the *optimal pairs* $\{P,K\}$. For example, $\{4,2\}$ indicates simulation run with fourth order harmonic in ACE and second order temporal basis functions. In general first, second and third order temporal basis function can provide up to $\mathcal{O}(10^{-4})$, $\mathcal{O}(10^{-5})$ and $\mathcal{O}(10^{-7})$ accuracies respectively, for the given source signal parameters (21). Table 3 shows the relative error for different $\{P,K\}$ pairs and distribution sizes, N_s. It can be seen that for increasing $\{P,K\}$ combination the error decreases consistently. Table 4 presents the per time step computation time involved in both direct and proposed algorithm, the order of error corresponding to different $\{P,K\}$ pairs can be inferred from Table 3.

Similar results are presented for multiple time step interaction in Tables 5 and 6, where N denotes the number of distinct time step interactions and C_s denotes the sidelength of cube enclosing all sources/observers in meters. In Tables 3–6 empty entries, pertaining to large N_s and $\{P,K\}$ values, are due to insufficient computer memory on the chosen computer platform. Fig. 4 shows N_s vs. T_{far} graph in log scale for data in Table 4. The lines plotted in the graph corresponds to a least square error linear fit for different $\{P,K\}$ pairs. Slope of these line for different $\{P,K\}$ values was approximately 1.06, thus, validating the $\mathcal{O}(N_s)$ scaling of algorithm presented here.

The evident mismatch between timings in Tables 6 and 4 is explained as follows. In the case of single time step interaction, the size of smallest box was chosen to accommodate 60–70 unknowns per box on average. However the largest box, at top of the tree (level 1), is within $c\Delta_t$ dimensions; therefore, the height of the tree

Table 5
$\text{Error}_{\text{far}}$ in multiple time step interaction case, for various combination of N_s, N and $\{P,K\}$ pairs. N is the number of distinct time steps involved

N_s	Levels	C_s	N	$\text{Error}_{\text{far}}, \{P,K\}$					
				{1,1}	{2,1}	{3,1}	{4,2}	{5,2}	{9,3}
8000	4	1.0	2	2.92E−3	5.79E−4	3.15E−4	6.27E−5	2.67E−5	1.97E−6
12,000	4	1.0	2	2.97E−3	5.42E−4	3.07E−4	5.67E−5	2.45E−5	1.90E−6
32,000	4	1.0	2	3.50E−3	7.06E−4	4.53E−4	8.14E−5	2.99E−5	3.32E−6
32,000	5	2.0	3	2.11E−3	4.07E−4	3.22E−4	5.03E−5	1.54E−5	9.15E−7
$128{,}000^\dagger$	6	2.0	3	3.80E−3	3.33E−4	3.30E−4	7.19E−5	1.62E−5	–

Table 6
Comparison of run-time in multiple time step interaction case

N_s	C_s	N	T_{Direct}	T_{Fast}, $\{P,K\}$			
				{1,1}	{2,1}	{4,2}	{9,3}
8000	1.0	2	2.02	0.03	0.06	0.31	4.85
12,000	1.0	2	4.69	0.04	0.10	0.49	7.68
32,000	1.0	2	61.93	0.19	0.44	2.52	43.97
32,000	2.0	3	34.31	0.17	0.32	1.67	28.49
64,000	2.0	3	–	0.67	1.52	8.76	165.45
500,000	2.0	4	–	27.89	55.47	294.06	–
1,000,000	1.0	2	–	45.52	83.37	433.34	–

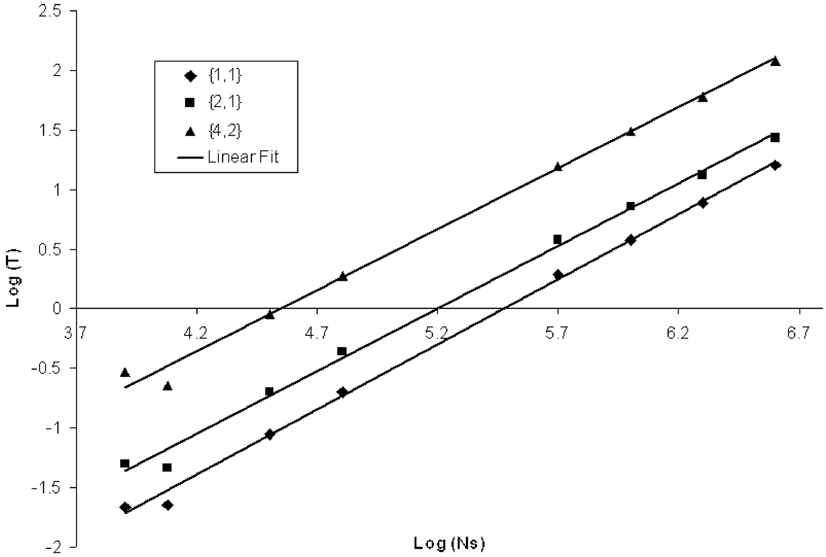

Fig. 4. log (N_s) vs. log (T_{far}) for single interaction case and uniform geometry.

increases as distribution size is increased. In case of multiple time step interactions one can keep the leaf box size constant and increase the level-1 box size for higher distribution size, to achieve ≈64 unknowns per leaf box. However, this does not imply a direct increase in tree height because the interactions at larger boxes also need to obey (19). For example, for two spheres of radius r_s to interact, the limiting condition based on (19), is $r_s \leqslant c\Delta_t/4$. Boxes greater than this size interact only through their child. This is the only limitation of the algorithm presented here, however, in practice the algorithm can be strictly used to compute field interacting in few time steps *only* and PWTD will interface with this method when $\lfloor R/(c\Delta_t)\rfloor$ is beyond a certain number of time steps. Thus, an ideal algorithm should switch between the proposed algorithm and PWTD seamlessly.

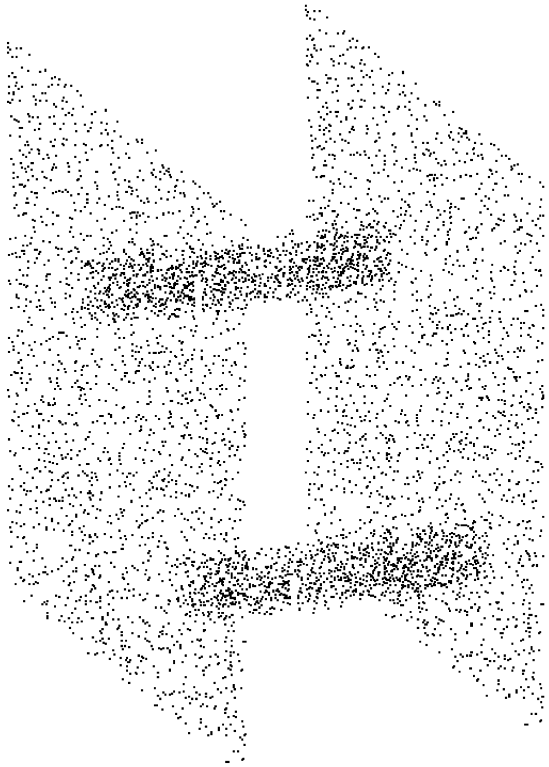

Fig. 5. Non-uniform geometry configuration 1, resembling interconnect in electronic chips ($N_s = 12{,}000$).

Table 7
Error_far for non-uniform geometry configuration 1

N_s	Levels	Error_far, $\{P,K\}$							
		{1,1}	{2,1}	{3,2}	{4,2}	{5,2}	{6,3}	{8,3}	{9,3}
8000	4	3.81E−3	9.46E−4	1.83E−4	3.78E−5	1.62E−5	3.06E−6	1.18E−6	7.52E−7
12,000	5	3.58E−3	8.62E−3	1.64E−4	3.43E−5	1.51E−5	2.59E−6	1.10E−7	7.27E−7
32,000	6	3.23E−3	6.75E−4	1.14E−4	2.69E−5	1.32E−5	2.02E−6	9.46E−7	6.93E−7

Table 8
Comparison of run-time for non-uniform geometry configuration 1

N_s	T_Fast, $\{P,K\}$			
	{1,1}	{2,1}	{4,2}	{6,3}
8000	0.03	0.05	0.14	0.49
16,000	0.05	0.09	0.32	1.03
32,000	0.11	0.20	0.63	2.28
64,000	0.25	0.43	1.42	4.8
250,000	1.19	1.75	5.33	18.86
500,000	2.61	3.58	10.86	40.08
1,000,000	6.11	7.8	23.25	79.63

Finally, results for an adaptive version of the algorithm introduced here is shown on two types of non-uniformly distributed geometries. The first closely resembles interconnects in electronic chips as shown in Fig. 5. The distribution of points between top and bottom planes and two interconnects were approximately the same. In applying the adaptive version, the number of unknowns per leaf node was approximately 64, was

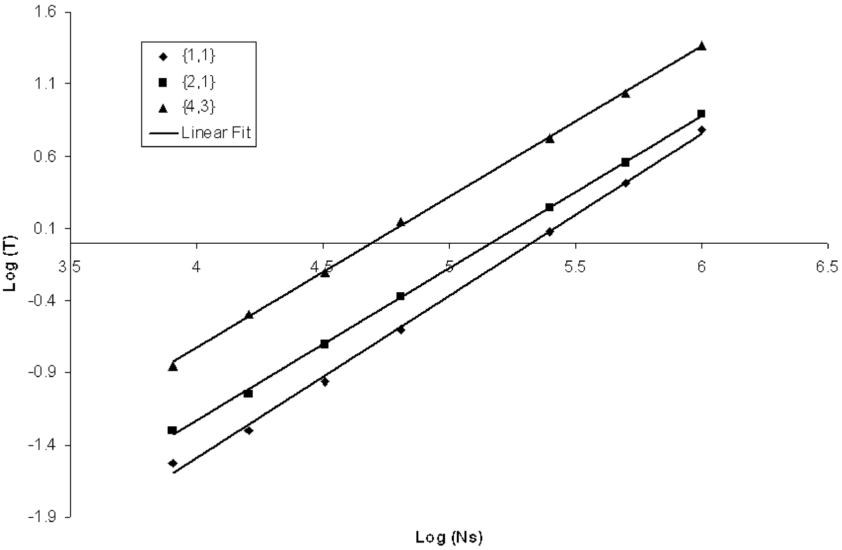

Fig. 6. log (N_s) vs. log (T_{far}) for single interaction case and non-uniform geometry.

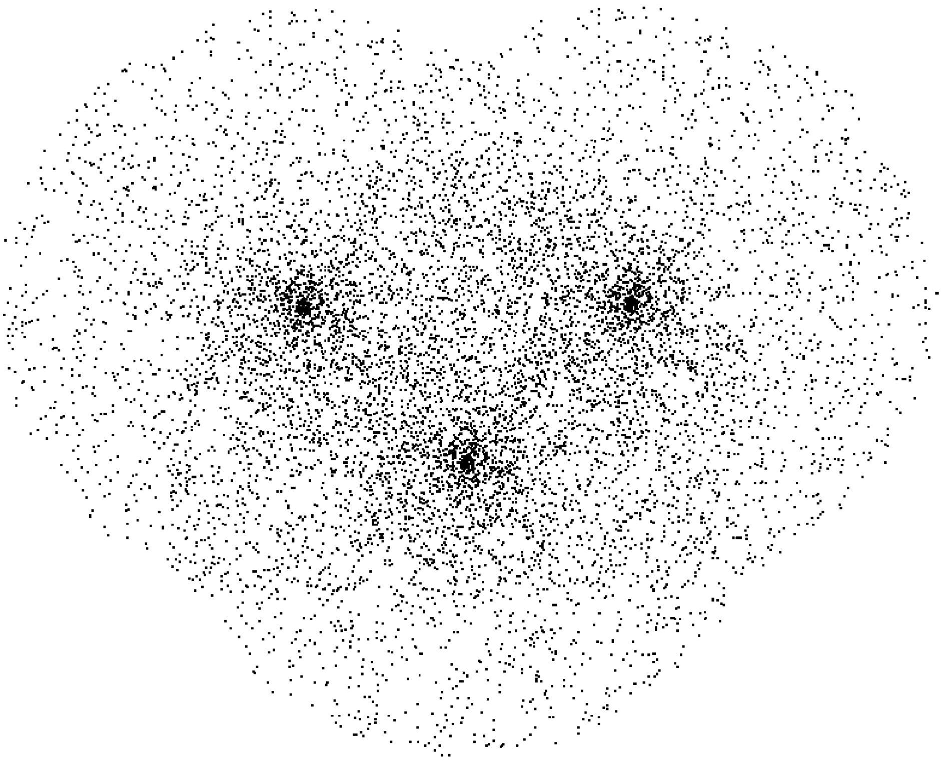

Fig. 7. Non-uniform geometry configuration 2 ($N_s = 9600$).

tested for source/observer distributions ranging from 8000 to 1,000,000. Table 7 presents the error obtained using the proposed algorithm, and was generated for different combinations of ACE harmonics (P) and order of time basis function (K). The rate of error convergence exhibited here is fast in comparison to those in Tables 3 and 5. This outcome is primarily attributed to the consideration of smallest box enclosure and stricter

Table 9
Comparison of run-time for non-uniform geometry configuration 2

N_s	T_{Fast}, $\{P,K\}$			
	$\{1,1\}$	$\{2,1\}$	$\{4,2\}$	$\{6,3\}$
9600	0.03	0.05	0.19	0.51
38,400	0.13	0.23	0.79	2.88
105,000	0.4	0.72	2.35	8.42
450,000	2.16	3.18	10.52	37.34
1,000,000	5.17	7.07	24.33	82.46

enforcement of error criteria in building the interaction list; see Appendix A.2 for details. The timing result for this geometry configuration is presented in Table 8. As explained above in uniform distribution, the timing results are presented only for certain combinations of $\{P,K\}$, each pair corresponding to different orders of accuracy given in Table 7. Fig. 6 shows N_s vs. T_{far} graph in log scale. The slope of the linear fit was approximately 1.06 for different pairs of $\{P,K\}$, exhibiting the $\mathcal{O}(N_s)$ scaling produced by the adaptive version of the algorithm. The second geometry configuration considered is made of three circles with points non-uniformly distributed in each of them as shown in Fig. 7. Each circle is 0.15 m in radius and the points were distributed so that density of points is inversely proportional to the radius. The adaptive version is applied on five different distribution sizes varying from 9600 to 1,000,000 and the results are shown in Table 9. As before, it can be verified that the time scaling is $\mathcal{O}(N_s)$.

5. Conclusion

In this paper, a novel fast method, based on Accelerated Cartesian Expansions (ACE), has been developed for computation of transient field due to sub-wavelength source/observer distributions. The proposed method reduces the computational cost from $\mathcal{O}(N_s^2)$ to $\mathcal{O}(N_s)$ per time step for distributions that lie within a time step. Though the algorithm is presented in detail for Lagrange-type basis functions, the extension to any class of basis function is straightforward and follows the same procedure and criteria used here. An adaptive version of the algorithm was also developed for analyzing non-uniformly distributed source/observers. Plethora of numerical results presented demonstrate both the accuracy and speed of the proposed algorithm. In the distributions tested, it is evident that the modification suggested to the kernel are effective as is the algorithm for the non-uniform distribution. Results have also been presented for analyzing distributions in domains whose extent is greater than $c\Delta_t$. It is evident that the proposed algorithm produces accurate results. However as noted, the algorithm presented herein is not optimal in an asymptotic sense for domain sizes that span multiple time steps. Ideally this algorithm will be used for source distributions in sub-wavelength domain and defer to other fast methods (specifically, the plane wave domain algorithm) at higher frequencies. As, far interactions in the PWTD algorithm are boxes that are separated by $\mathcal{O}(c\Delta_t)$, the presented algorithm can be used for all interactions wherein the separation distance is *less* than $\mathcal{O}(c\Delta_t)$. Thus, the asymptotic scaling of the presented algorithm due to separation distance greater than $\mathcal{O}(c\Delta_t)$ does not have a great impact. Work on implementing this algorithm with the framework of PWTD is currently underway, and will be presented elsewhere.

Acknowledgments

We are grateful to NSF for support under CCF-0306436 and CCF-0729157, and the comments by three anonymous reviews whose insights made this a much better paper.

Appendix A. Implementation details

This section briefly describes details useful for implementing this algorithm; this is done largely for completeness. If an algorithm already exists, then only variation that are used herein are presented.

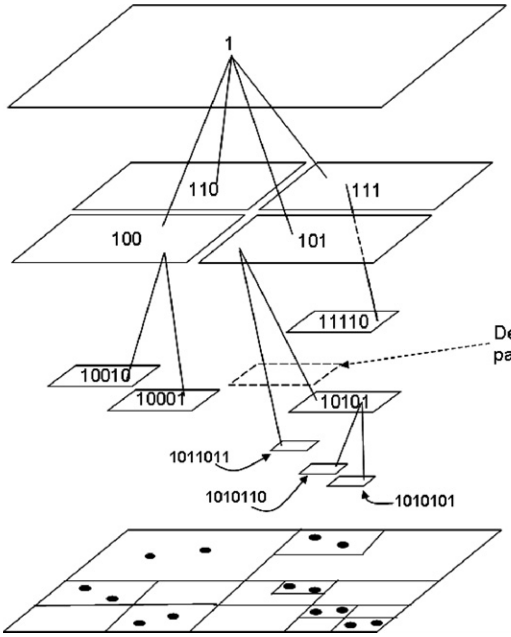

Fig. 8. An example of compressed-quadtree with binary key representation used to label the tree nodes.

A.1. Reduced interaction list

It is well known [20,16] that evaluation of multipole-to-local translation for pairs in interaction list forms the major part of the computation cost. Both the per-translation evaluation cost and number of interaction pairs (typically 189) are very high. To reduce this cost, we use a new definition to classify far-field pairs: *if box a (at level l + 1) interacts with all the children of box b (at level l) and box a, box b are in far-field of each other then box a interacts with box b*. Interaction between boxes at two consecutive levels is easily effected using Cartesian tensors. In fully populated oct-tree this results in a reduction in the number of translation operations by half with minimal increase in error, as is evident in Table 1.

A.2. Compressed oct-tree

In this work, we closely follow the approach presented in [16] for compressed oct-tree representation of non-uniform geometries. The main deviation from [16] is, we enclose the smallest box with some pre-fixed number of points per box, *s*. While this approach is not significantly different in terms of cost when compared with [16], it does provide the possibility of improving error with certain geometries as the error in multipole evaluation is reduced. With the elimination of single child parent nodes, the resulting oct-tree would have the same structure as in [16] except the leaf box size would be smaller here as shown in Fig. 8. Interaction list at the leaf nodes is compiled using the approach prescribed in [16]. Binary keys [21] were used to represent the nodes of oct-tree.

References

[1] R. Coifman, V. Rokhlin, S. Wandzura, The fast multipole method: a pedestrian prescription, IEEE Antennas Propag. Mag. 35 (3) (1993) 7–12.

[2] E. Bleszynski, M. Bleszynski, T. Jaroszewicz, Aim: adaptive integral method for solving large-scale electromagnetic scattering and radiation problems, Radio Sci. 31 (5) (1996) 1225–1251.

[3] A.A. Ergin, B. Shanker, E. Michielssen, The plane wave time domain algorithm for the fast analysis of transient wave phenomena, IEEE Antennas Propag. Mag. 41 (4) (1999) 39–52.

[4] A.E. Yilmaz, J.M. Jin, E. Michielssen, Time domain adaptive integral method for surface integral equations, IEEE Trans. Antennas Propag. 52 (10) (2004) 2692–2708.

[5] A.E. Yilmaz, J.M. Jin, E. Michielssen, Analysis of low-frequency electromagnetic transients by an extended time domain adaptive integral method, IEEE Trans. Adv. Pack. 30 (2) (2007) 301–312.

[6] J.-S. Zhao, W.C. Chew, Three-dimensional multilevel fast multipole algorithm from static to electrodynamic, Microwave Opt. Technol. Lett. 26 (1) (2000) 43–48.

[7] E. Darve, P. Havé, Efficient fast multipole method for low-frequency scattering, J. Comput. Phys. 197 (1) (2004) 341–363.

[8] L. Greengard, J. Huang, V. Rokhlin, S. Wandzura, Accelerating fast multipole methods for the Helmholtz equation at low frequencies, IEEE Comput. Sci. Eng. 05 (3) (1998) 32–38.

[9] M. Vikram, H. Griffith, H. Huang, B. Shanker, A uniform framework for time and frequency domain low-frequency problems, in: Proceedings of the 2007 IEEE International Symposium on Antennas and Propagation Society International, 2007.

[10] H. Cheng, W.Y. Crutchfield, Z. Gimbutas, L.F. Greengard, J.F. Ethridge, J. Huang, V. Rokhlin, N. Yarvin, J. Zhao, A wideband fast multipole method for Helmholtz equation in three dimensions, J. Comput. Phys. 216 (2006) 300–325.

[11] L.J. Jiang, W.C. Chew, A mixed-form fast multipole algorithm, IEEE Trans. Antennas Propog. 53 (12) (2005) 4145–4156.

[12] B. Shanker, A. Ergin, M. Lu, E. Michielssen, Fast analysis of transient electromagnetic scattering phenomena using the multilevel plane wave time domain algorithm, IEEE Trans. Antennas Propag. 51 (3) (2003) 628–641.

[13] K. Aygün, Novel Fast Algorithms for Time Domain Integral Equation Analysis of Antennas and Circuits, Ph.D. Thesis, University of Illinois at Urbana, 2002.

[14] J. Meng, M. Lu, E. Michielssen, Towards efficient and stable low-frequency time domain integral equation solvers, in: Proceedings of the 2006 IEEE International Symposium on Antennas and Propagation, 2006, pp. 2955–2958.

[15] J. Meng, M. Lu, E. Michielssen, A fast space-adaptive algorithm to evaluate transient wave fields due to low-frequency source constellations, in: Proceedings of the 2005 IEEE International Symposium on Antennas and Propagation, vol. 3A, 2005, pp. 171–174.

[16] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (2) (1999) 468–498.

[17] B. Shanker, H. Huang, Accelerated Cartesian expansions – a fast method for computing of potentials of the form $r^{-\nu}$ for all real ν, J. Comput. Phys. (2007), doi:10.1016/j.jcp.2007.04.033.

[18] J. Zheng, R. Balasundaram, S.H. Gehrke, G.S. Heffelfinger, W.A. Goddard III, S. Jiang, Cell multipole method formolecular simulations in bulk and confined systems, J. Chem. Phys. 118 (12) (2003) 5347–5355.

[19] F. Zhao, An $\mathcal{O}(n)$ Algorithm for Three-dimensional n-Body Simulation, Masters Thesis, Massachusetts Institute of Technology.

[20] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.

[21] M.S. Warren, J.K. Salmon, A parallel hashed oct-tree n-body algorithm, in: Supercomputing'93: Proceedings of the 1993 ACM/IEEE Conference on Supercomputing, ACM Press, New York, NY, USA, 1993, pp. 12–21.