Journal of Computational Physics 229 (2010) 9119–9134

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Accelerated Cartesian expansion (ACE) based framework for the rapid evaluation of diffusion, lossy wave, and Klein–Gordon potentials

M. Vikram [a], A. Baczewski [a,b], B. Shanker [a,b,*], L. Kempel [a]

[a] Michigan State University, Department of Electrical and Computer Engineering, East Lansing, MI, USA
[b] Michigan State University, Department of Physics and Astronomy, East Lansing, MI, USA

ARTICLE INFO

Article history:
Received 20 September 2009
Received in revised form 9 August 2010
Accepted 20 August 2010
Available online 27 August 2010

Keywords:
Accelerated Cartesian expansion (ACE)
Diffusion
Lossy wave
Transient
Block-Toeplitz
Fast multipole methods

ABSTRACT

Diffusion, lossy wave, and Klein–Gordon equations find numerous applications in practical problems across a range of diverse disciplines. The temporal dependence of all three Green's functions are characterized by an infinite tail. This implies that the cost complexity of the spatio-temporal convolutions, associated with evaluating the potentials, scales as $\mathcal{O}\left(N_s^2 N_t^2\right)$, where N_s and N_t are the number of spatial and temporal degrees of freedom, respectively. In this paper, we discuss two new methods to rapidly evaluate these spatio-temporal convolutions by exploiting their block-Toeplitz nature within the framework of accelerated Cartesian expansions (ACE). The first scheme identifies a convolution relation in time amongst ACE harmonics and the fast Fourier transform (FFT) is used for efficient evaluation of these convolutions. The second method exploits the rank deficiency of the ACE translation operators with respect to time and develops a recursive numerical compression scheme for the efficient representation and evaluation of temporal convolutions. It is shown that the cost of both methods scales as $\mathcal{O}(N_s N_t \log^2 N_t)$. Several numerical results are presented for the diffusion equation to validate the accuracy and efficacy of the fast algorithms developed here.

1. Introduction

The diffusion equation is used to model a number of different physical phenomena, including eddy currents [1], heat conduction [2], crystal growth [3], and pharmacokinetics [4]. Likewise, solutions to the lossy wave equation finds use in many fields, ranging from wave propagation physics [5] to relativistic diffusion phenomena [2]. The Klein–Gordon equation holds a special place in theoretical physics as the relativistic equation of motion for massive scalar particles [6]. Given the many fields in which these equations find use, methods for their efficient solution have widespread impact. Typically, these equations are solved using differential equation based methods (e.g. finite differences, finite elements, etc) due to ease of implementation and a plethora of readily available codes. However, these methods generally demand the discretization of the entire domain and employ artificial or approximate boundary conditions to truncate the domain. Additionally, they are susceptible to grid dispersion. These features translate to a higher computational cost at reduced accuracy. Integral equation formulations offer a means of overcoming these computational bottlenecks. However, the associated solvers are expensive as the cost of evaluating the necessary spatio-temporal convolutions scales as $\mathcal{O}(N_s^2 N_t^2)$, where N_s and N_t represent the spatial and temporal degrees of freedom, respectively.

* Corresponding author at: Michigan State University, Department of Electrical and Computer Engineering, East Lansing, MI, USA. Tel.: +1 517 432 8136; fax: +1 517 353 1980.
E-mail addresses: vikramr@msu.edu (M. Vikram), baczewsk@msu.edu (A. Baczewski), bshanker@msu.edu (B. Shanker), kempel@egr.msu.edu (L. Kempel).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.08.025

Methods to ameliorate the cost of using integral equation based solvers have been a topic of considerable intellectual interest. These include algorithms based upon multipole expansions [7], the fast Fourier transform (FFT) [8], and numerical compression [9]. Among these, the fast multipole method (FMM) has enjoyed widespread popularity due to its mathematically rigorous formulation, which enables the derivation of strict bounds on cost and error. FMM was first introduced for the fast evaluation of Coulomb potentials in large systems [7,10]. This method and its variations have been widely used in a number of different applications including molecular dynamics, astrophysics [11], electrical engineering [12,13], and fluid mechanics [14]. One such variation was developed for the fast computation of Gauss transforms [15], which play an important role in several areas [16]. Over the years, this method has seen considerable improvement [17]. The direct application of this method to the convolution between the diffusion kernel and some source signature reduces the cost to $\mathcal{O}\left(N_s N_t^2\right)$. Acceleration of the temporal convolution was first addressed by the separation of time scales [16], and was used in the solution of the heat equation [18]. A recent work [19] employs Chebyshev polynomial expansions in both the space and time domains to develop an acceleration scheme whose cost scales as $\mathcal{O}(N_s N_t)$. All of these methods accelerate the solution of the diffusion equation completely in real space. Alternatively, the diffusion equation can be solved in Fourier space [20,21] where the cost scales as $\mathcal{O}(N_s N_t \log N_s)$.

The fast evaluation of potentials associated with the lossy wave equation is a more recent endeavour. One method used to arrive at such an algorithm relies upon relating the three-dimensional Green's function to the one-dimensional Green's function via a spectral integral and evolving the fields rapidly in one-dimension [22,23]. The asymptotic cost of this approach is $\mathcal{O}(N_s N_t \log(N_s)\log(N_t))$. While this has not been done, it is possible to extend this method to the analysis of the Klein–Gordon equation.

In this paper, we propose a common framework for the fast evaluation of time domain potentials that arise in the solution of the diffusion, lossy wave, and Klein–Gordon equations. The proposed methodology is based upon integrating the recently developed accelerated Cartesian expansion (ACE) [24] algorithm for spatial acceleration with either the Fast Fourier Transform (FFT) or numerical compression methods for temporal acceleration. ACE is a FMM-type fast algorithm formulated in terms of Cartesian tensors and employs the generalized Taylor expansion to derive addition theorems for non-oscillatory potentials. Further, ACE exploits the fact that these tensors are totally symmetric, and derives an exact algorithm for upward and downward tree traversals. It must be noted that, similar methodologies have been introduced earlier [25,26] but they are either not generalizable or offer only some of the advantages of this scheme. ACE has been successfully used to accelerate a number of different kernels, including London [24], low frequency Helmholtz (extensible to Yukawa) [27,28], and retarded wave [29]. The temporal acceleration schemes, both FFT and numerical compression, are also independent of the form of the kernel. Thus, the fast method resulting from the combination of these approaches serve as as a uniform framework for the evaluation of different potentials. In this paper, the details of the unified framework are presented in the context of the diffusion kernel. The same framework can be used for both the lossy wave and Klein–Gordon equations by using different ACE translation operators. The derivation of the translation operators for these equations are provided in Appendix A and B, respectively, and are referenced at appropriate junctures in the main discussion. Further, in this paper, we focus on the development of these acceleration methods for computing potentials and *not* solving these equations for any application specific problem. To summarize, the main contributions of this work are

- Development of a modified fast Gauss transform (FGT) based upon ACE, which provides exact translation operators for upward and downward tree traversal.
- Integration of ACE with FFT based temporal acceleration schemes. The overall cost of this scheme scales as $\mathcal{O}(N_s N_t \log^2 N_t)$.
- Development of a numerical compression scheme for temporal acceleration within the ACE framework. The overall cost of this scheme scales as $\mathcal{O}(N_s N_t \log^2 N_t)$.
- Development of ACE translation operators for diffusion, lossy wave, and Klein–Gordon equations.

The rest of the paper is organized as follows: Section 2 introduces the appropriate partial differential equations and their associated potentials. Section 3 describes the details of the proposed acceleration methods in both space and time. Here, we first discuss the ACE algorithm used for acceleration of spatial convolution followed by the details of the two acceleration schemes, introduced in this work, for the evaluation of the spatio-temporal convolution. Section 4 presents several numerical results for the case of the diffusion potential to demonstrate the efficacy of our algorithm in terms of both speed and accuracy. Finally, the conclusions of this work are drawn in Section 5.

2. Mathematical preliminaries

Consider a domain, $\Omega \times t \subset \mathbb{R}^3 \times \mathbb{R}^+$, that contains a distribution of sources, $f(\mathbf{r},t)$, that is bandlimited to f_{max} and approximately time limited to T. The field due to this distribution of sources satisfies the lossy wave Eq. (1).

$$\begin{aligned} &\left(\nabla^2 - \alpha^2\partial_t - \frac{1}{c^2}\partial_t^2\right)\Phi(\mathbf{r},t) = f(\mathbf{r},t) \quad \text{for } \mathbf{r}\times t \in \Omega \times \mathbb{R}^+, \\ &\Phi(\mathbf{r},0) = u_0(\mathbf{r}), \partial_t\Phi(\mathbf{r},0) = v_0(\mathbf{r}) \quad \text{for } \mathbf{r} \in \Omega. \end{aligned} \tag{1}$$

Here, $\Phi(\mathbf{r},t)$ is the lossy wave potential at position $\mathbf{r} = r_x\hat{x} + r_y\hat{y} + r_z\hat{z}$ and time t, $u_0(\mathbf{r})$ and $v_0(\mathbf{r})$ are the initial conditions, and α and $c \in \mathbb{R}$ are problem dependent constants. We recover the diffusion Eq. (2) from (1) in the limit $c \to \infty$,

$$(\nabla^2 - \alpha^2\partial_t)\Phi(\mathbf{r},t) = f(\mathbf{r},t) \quad \text{for } \mathbf{r}\times t \in \Omega\times\mathbb{R}^+. \tag{2}$$

Replacing $\partial_t\Phi(\mathbf{r},t)$ by $\Phi(\mathbf{r},t)$ in (1) yields the Klein–Gordon equation (3)

$$\left(\nabla^2 - \alpha^2 - \frac{1}{c^2}\partial_t^2\right)\Phi(\mathbf{r},t) = f(\mathbf{r},t) \quad \text{for } \mathbf{r}\times t \in \Omega\times\mathbb{R}^+. \tag{3}$$

Subject to free-space boundary conditions, the Green's function for Eqs. (1)–(3) are given as [2]

Lossy wave equation:

$$G(\mathbf{r},t) = e^{-\frac{\alpha^2c^2t}{2}}\left[\frac{\delta(t-\|\mathbf{r}\|/c)}{\|\mathbf{r}\|} + \frac{\alpha^2 cI_1\left(\frac{\alpha^2c}{2}\sqrt{t^2-(\|\mathbf{r}\|/c)^2}\right)}{2\sqrt{t^2-(\|\mathbf{r}\|/c)^2}}\Theta(t-\|\mathbf{r}\|/c)\right]. \tag{4a}$$

Diffusion equation:

$$G(\mathbf{r},t) = \frac{\alpha}{(4\pi t)^{3/2}}g(r_x,t)g(r_y,t)g(r_z,t)\Theta(t), \quad \textit{where } g(r_l,t) = e^{-\alpha^2r_l^2/4t}, \quad l\in\{x,y,z\}. \tag{4b}$$

Klein-Gordon equation:

$$G(\mathbf{r},t) = \frac{\delta(t-\|\mathbf{r}\|/c)}{\|\mathbf{r}\|} - \frac{\alpha}{\sqrt{t^2-(\|\mathbf{r}\|/c)^2}}J_1\left(\alpha c\sqrt{t^2-(\|\mathbf{r}\|/c)^2}\right)\Theta(t-\|\mathbf{r}\|/c). \tag{4c}$$

Here, $\delta(x)$ is a Delta distribution, $\Theta(x)$ is a Heaviside distribution, $I_1(x)$ is a first order modified Bessel function of the first kind, and $J_1(x)$ is a first order Bessel function of the first kind.

Evaluation of $\Phi(\mathbf{r},t)$ at all times can be performed using either an explicit or implicit scheme while marching on in time. Explicit time stepping schemes are straight-forward as they require the evaluation of the potential $\Phi(\mathbf{r},t)$ at each time step using only the potentials evaluated at past time steps. In contrast, implicit time stepping schemes requires one to *solve* for the potential $\Phi(\mathbf{r},t)$ at both present and future time steps, simultaneously, while using the potentials evaluated at past time steps as fictitious sources. In an integral equation solution of the above governing Eqs. (1)–(3) using implicit time stepping schemes, requires one to *solve* for fictitious sources defined in Ω. Such integral equation formulations are well known and their derivation is not repeated here; interested readers are referred to [16,19,21] and references therein. Whether an implicit or explicit time stepping scheme is employed, both of them demand repeated evaluation of the potential and, in general, the potential at any point can be written as,

$$\Phi(\mathbf{r},t) = \Phi_1(\mathbf{r},t) + \Phi_2(\mathbf{r},t), \tag{5a}$$

$$\Phi_1(\mathbf{r},t) = \int_\Omega \mathcal{L}\{G(\mathbf{r}-\mathbf{r}',t), u_0(\mathbf{r}'), v_0(\mathbf{r}')\}d\mathbf{r}', \tag{5b}$$

$$\Phi_2(\mathbf{r},t) = \int_0^t\int_\Omega f(\mathbf{r}',t')G(\mathbf{r}-\mathbf{r}',t-t')d\mathbf{r}'dt', \tag{5c}$$

where $G(\mathbf{r},t)$ represents the Green's function specific to the choice of the governing Eq. (1)–(3), $\mathcal{L}$ is some appropriate linear functional propagating the effect of the initial conditions, and $f(\mathbf{r}',t')$ is the unknown source distribution. It is evident that the computation of Φ_1 in (5b) can be considered as a subset of computing Φ_2 in (6). Consequently, in the rest of the paper, we devote our attention to the rapid evaluation of Φ_2. Without loss of generality, the discrete version of (5c) for a uniform sample size in time, Δ_t, can be written as

$$\Phi_2(\mathbf{r},t) = \int_{\lfloor t/\Delta_t\rfloor\Delta_t}^t\int_\Omega f(\mathbf{r}',t')G(\mathbf{r}-\mathbf{r}',t-t')d\mathbf{r}'dt' + \sum_{l=0}^{\lfloor t/\Delta_t\rfloor}\sum_{i=1}^{N_s} w(\mathbf{r}_i',l\Delta_t)G(\mathbf{r}-\mathbf{r}_i',t-l\Delta_t)f_i(\mathbf{r}_i',l\Delta_t), \tag{6}$$

$$\Phi_2(\mathbf{r},t) = F_{self}(\mathbf{r},\Delta_t) + \sum_{l=0}^{\lfloor t/\Delta_t\rfloor}\sum_{i=1}^{N_s} w(\mathbf{r}_i',l\Delta_t)G(\mathbf{r}-\mathbf{r}_i',t-l\Delta_t)f_i(\mathbf{r}_i',l\Delta_t), \tag{7}$$

where N_s is the number of points in the spatial discretization, $\lfloor\cdot\rfloor$ denotes the floor operation, $w(\mathbf{r}_i',l\Delta_t)$ is an appropriate quadrature weight, and $\mathbf{r}_i'$ and f_i are the position and strength associated with the i-th spatial source, respectively. The first term in the right hand side of (7), F_{self}, contains the singular contributions to the potential and the second summation corresponds to the partial contribution of sources up to the discrete time step before the potential is being evaluated. Accurate evaluation of the singular contributions, in F_{self}, is crucial for integral equation solution and is well documented in the literature [18,19], as are acceleration techniques for its evaluation with a cost that scales as $\mathcal{O}(N_sN_t)$. Furthermore, note that their evaluation can be accelerated by realizing that they are Toeplitz in time. As a result, methods presented here also can be easily adapted to evaluating it with comparable cost. As a result, the evaluation of F_{self} will not be considered in the rest of the discussion. Testing (6) at N_s spatial points and at time t, leads to the following matrix equation

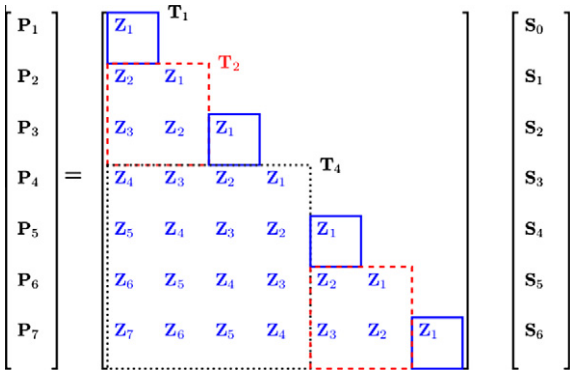

Fig. 1. Rewriting (8) in matrix form without $\mathbf{F}_{self}$. Further, illustration of the block-Toeplitz nature of the matrix system.

$$\mathbf{P}_k = \sum_{l=0}^{k-1} \mathbf{Z}_{k-l}\mathbf{S}_l + \mathbf{F}_{self}, \tag{8}$$

where the vectors $\mathbf{P}_k = \{\Phi_2(\mathbf{r}_1, k\Delta_t), \Phi_2(\mathbf{r}_2, k\Delta_t), \cdots, \Phi_2(\mathbf{r}_{N_s}, k\Delta_t)\}^T$, $\mathbf{S}_l = \{f(\mathbf{r}_1, l\Delta_t), f(\mathbf{r}_2, l\Delta_t), \cdots, f(\mathbf{r}_{N_s}, l\Delta_t)\}^T$, $\mathbf{Z}_{k-l}$ is a $N_s \times N_s$ matrix whose elements are $\mathbf{Z}_{k-l}(i,j) = w(\mathbf{r}_j, l\Delta_t)G(\mathbf{r}_i - \mathbf{r}_j, (k-l)\Delta_t)$ and $k = \lfloor t/\Delta_t \rfloor + 1$. The above summation can be written as a space-time matrix vector product [30] as shown in Fig. 1. The computational cost associated with a single matrix vector product $\mathbf{Z}_{k-l}\mathbf{S}_l$ is $\mathcal{O}(N_s^2)$ and hence the cost of evaluating $\mathbf{P}_k$ at all N_t time steps scales as $\mathcal{O}(N_s^2 N_t^2)$. Parenthetically, we note that potential evaluators for the wave equation (in both two and three dimensions) and the lossy wave equation take a form that is similar to (8) [31–33,23,29]. With simple algebraic manipulation, it is possible to retrofit these fast schemes into marching on in time (MOT) solvers for solving time domain integral equations. Of course, this comes with the additional cost of storing the temporal history at each source/observation point, and a naive implementation has a memory requirement for storing the history that scales as $\mathcal{O}(N_t N_s)$. For the equations considered here, it is possible to use compression algorithms to reduce this cost. MOT solvers that are augmented by these fast techniques have found extensive application; see Refs. [34–39], and references therein for details on integration of fast potential evaluators with integral equation solvers. In the following Section, methodologies are presented to reduce this computational cost in both space and time. As mentioned in Section 1, the description that follows is presented explicitly in the context of the diffusion potential. Relevant comments pertaining to the lossy wave and Klein–Gordon potentials are included at appropriate junctures.

3. Acceleration schemes

In this Section, we detail the development of a uniform framework for the rapid evaluation of the discrete spatial and temporal convolutions in (7). The strategy used in developing a space-time acceleration scheme lies in the fact that the discrete convolution is block-Toeplitz in time. This is evident from examining the space-time matrix Fig. 1, which is a reformulation of (8). Each of the blocks that form this Toeplitz structure are denoted by $\mathbf{Z}_k$. Thus, an efficient algorithm to evaluate the temporal convolution can be realized using block computation similar to that presented in Ref. [30] provided it can be integrated with a method to rapidly effect the product $\mathbf{Z}_{k-l}\mathbf{S}_l$. Our choice for rapidly computing $\mathbf{Z}_{k-l}\mathbf{S}_l$ is the ACE algorithm. Consequently, in what follows, we shall briefly review the salient features of ACE and present relevant theorems. This will be followed by a detailed presentation of integration of ACE with a block computation scheme, using FFT and numerical compression techniques.

3.1. Accelerated Cartesian expansion (ACE)

Each term of the summation in (8) corresponds to a fixed source and observation time ($l\Delta_t$) and ($k\Delta_t$), respectively, which is indicated by the subscript. The single matrix vector product $\mathbf{Z}_{k-l}\mathbf{S}_l$ corresponds to the evaluation of the inner sum in (6). This evaluation is equivalent to an N-body problem and can be written as,

$$\mathbf{V}_{k,l} = \mathbf{Z}_{k-l}\mathbf{S}_l, \tag{9}$$

$$\mathbf{V}_{k,l}^m = \phi_{k,l}(\mathbf{r}_m) = \sum_{n=1}^{N_s} w(\mathbf{r}_n, l\Delta_t)G(\mathbf{r}_m - \mathbf{r}_n, (k-l)\Delta_t)f_n(\mathbf{r}_n, l\Delta_t), \tag{10}$$

where $\mathbf{V}_{k,l}$ is an intermediate vector of dimension N_s. It is evident that the direct cost of computing the entire vector $\mathbf{V}_{k,l}$ scales as $\mathcal{O}(N_s^2)$, suggesting that a spatial acceleration technique such as ACE can be used to efficiently compute $\mathbf{V}_{k,l}$ for all pertinent source-observer time intervals. Next, a brief overview of the ACE algorithm is presented with relevant definitions and theorems.

3.1.1. ACE preliminaries

The mathematical machinery of the ACE algorithm is expressed in the parlance of Cartesian tensors and hence warrants a brief overview of tensor notations. A Cartesian tensor of rank n is denoted by $\mathbf{A}^{(n)}$, and contains 3^n components for points in $\mathbb{R}^3$. A totally symmetric tensor is one that is independent of the permutation of indices $\alpha_1,\ldots,\alpha_n$; in compressed form it contains $(n+1)(n+2)/2$ independent components, and is denoted by $\mathbf{A}^{(n)}(n_1,n_2,n_3)$, where $n_1+n_2+n_3=n$. The n-fold contraction between two tensors, $\mathbf{A}^{(n+m)}$ and $\mathbf{B}^{(n)}$ results in a tensor $\mathbf{C}^{(m)}=\mathbf{A}^{(n+m)}\cdot n\cdot\mathbf{B}^{(n)}$.

Using this notation, we may now outline the Theorems that permit the fast evaluation of the potential in (8). To this end, we assume that the subdomain containing the sources, $\Omega_s\subset\ \Omega$, and the subdomain containing the observation points, $\Omega_o\subset\Omega$, are sufficiently far from each other. We also assume that $\Omega_s\subset\Omega_s^p,\ \Omega_o\subset\Omega_o^p$ and $\Omega_s^p\cap\Omega_o^p=\emptyset$. The centers of the domains Ω_s, Ω_o, Ω_s^p and Ω_o^p are denoted by $\mathbf{r}_s,\mathbf{r}_o,\ \mathbf{r}_s^p$ and $\mathbf{r}_o^p$, respectively. Further, let S denote the number of sources in Ω_s and let $G(\mathbf{r}-\mathbf{r}',t)$ denote the Green's function that maps the effects of these sources onto the observation points.

Theorem 1 (*Multipole Expansion*). *The total potential $\phi_{k,l}(\mathbf{r})$, in (10), at any point $\mathbf{r}\in\Omega_o$ and time $k\Delta_t$ due to S sources $f_i(\mathbf{r}_i',l\Delta_t),\ i=1,\ldots,S$ located at points $\mathbf{r}_i'\in\Omega_s$ and excited at time $l\Delta_t$ is given as*

$$\begin{aligned}\phi_{k,l}(\mathbf{r})&=\sum_{i=1}^{S}f_i(\mathbf{r}_i',l\Delta_t)G(\mathbf{r}-\mathbf{r}_i',(k-l)\Delta_t)=\sum_{n=0}^{\infty}\mathbf{M}_l^{(n)}\cdot n\cdot\nabla^n G(\mathbf{r},(k-l)\Delta_t),\\ \mathbf{M}_l^{(n)}&=\sum_{i=1}^{S}(-1)^n\frac{f_i(\mathbf{r}_i',l\Delta_t)}{n!}(\mathbf{r}_i'-\mathbf{r}_s)^n,\end{aligned}\tag{11}$$

where $\mathbf{M}_l^{(n)}$ is the multipole tensor evaluated at time $l\Delta_t$.

Theorem 2 (*Multipole-to-multipole expansion*). Given a multipole expansion of S sources about $\mathbf{r}_s$ at time $l\Delta_t$

$$\mathbf{O}_l^{(n)}=\sum_{i=1}^{S}(-1)^n\frac{f_i(\mathbf{r}_i,l\Delta_t)}{n!}(\mathbf{r}_i-\mathbf{r}_s)^n,\tag{12a}$$

then the multipole expansion about the point $\mathbf{r}_s^p$ and at time $l\Delta_t$ can be expressed in terms of (12a) as

$$\mathbf{M}_l^{(n)}=\sum_{i=1}^{S}(-1)^n\frac{f_i(\mathbf{r}_i,l\Delta_t)}{n!}(\mathbf{r}_i-\mathbf{r}_s^p)^n=\sum_{m=0}^{n}\sum_{P(m,n)}\frac{m!}{n!}(\mathbf{r}_s^p-\mathbf{r}_s)^{n-m}\mathbf{O}_l^{(m)},\tag{12b}$$

where $P(m,n)$ denotes the permutation of all partitions of n into sets $n-m$ and m. It is evident that one can repeatedly use this Theorem to translate the multipole expansion from $\mathbf{r}_s$ to $\mathbf{r}_s^p$. It is important to note that this operation is exact [24] and, hence, there is no error accumulation during upward tree traversal.

Theorem 3 (*Multipole-to-local translation*). *Assume that the domains Ω_s^p and Ω_o^p are sufficiently separated, and the distance between their centers $r_{os}^p=|\mathbf{r}_{os}^p|=|\mathbf{r}_o^p-\mathbf{r}_s^p|$ is greater than $\mathrm{diam}\{\Omega_s^p\}$ and $\mathrm{diam}\{\Omega_o^p\}$. If a multipole expansion $\mathbf{M}_l^{(n)}$ is located at $\mathbf{r}_s^p$, then another expansion $\mathbf{L}_{k,l}^{(n)}$ that produces the same field $\forall\mathbf{r}\in\Omega_o^p$ is given by*

$$\begin{aligned}\phi_{k,l}(\mathbf{r})&=\sum_{n=0}^{\infty}\boldsymbol{\rho}^n\cdot n\cdot\mathbf{L}_{k,l}^{(n)},\\ \mathbf{L}_{k,l}^{(n)}&=\sum_{m=n}^{\infty}\mathbf{M}_l^{(m-n)}\cdot(m-n)\cdot\frac{1}{n!}\nabla^m G(\mathbf{r}_{os}^p,(k-l)\Delta_t),\end{aligned}\tag{13}$$

where $\boldsymbol{\rho}=\mathbf{r}-\mathbf{r}_o^p$.

Theorem 4 (*Local-to-local expansion*). *A local expansion $\mathbf{O}_{k,l}^{(n)}$ that exists in the domain Ω_o^p centered around $\mathbf{r}_o^p$ can be shifted to the domain Ω_o centered at $\mathbf{r}_o$ using*

$$\mathbf{L}_{k,l}^{(n)}=\sum_{m=n}^{\infty}\binom{m}{m-n}\mathbf{O}_{k,l}^{(m)}\cdot(m-n)\cdot(\mathbf{r}_o-\mathbf{r}_o^p)^{m-n},\tag{14}$$

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

$$\phi_{k,l}(\mathbf{r})=\sum_{n=0}^{\infty}\mathbf{L}_{k,l}^{(n)}\cdot n\cdot(\boldsymbol{\rho}_{oi})^n,\tag{15}$$

The above Theorems of the ACE algorithm permit the rapid evaluation of potentials within a multi-level tree framework, similar to FMM [24,40]. Several salient features of the ACE algorithm are detailed in [24]. Relevant to this work is the fact that all operations *other* than the multipole-to-local-expansion translation operation are independent of the form of the kernel. The translation operator for the diffusion potential is presented below and that of the lossy wave and Klein–Gordon potentials are derived in Appendix A and B, respectively.

3.1.2. ACE translation operator for diffusion kernels

Rapid evaluation of $\mathbf{V}_{k,l}$ through the ACE algorithm requires the definition of $\nabla^p G(\mathbf{r},t) = \mathbf{H}^{(p)}$ to evaluate multipole-to-local expansion using Theorem 3. As shown below,

$$\mathbf{H}^{(p)}(p_1,p_2,p_3) = \frac{\alpha}{(4\pi t)^{3/2}} \partial_x^{p_1} g(r_x,t) \partial_y^{p_2} g(r_y,t) \partial_z^{p_3} g(r_z,t), \tag{16}$$

$$\partial_l^k g(r_l,t) = \frac{2r_l}{4t} \partial_l^{k-1} g(r_l,t) - 2(k-1)\partial_l^{k-2} g(r_l,t) l \in \{x,y,z\}$$

is a recursive relation that permits the efficient evaluation of each component of $\nabla^p G(\mathbf{r},t)$. It can be easily shown that Theorems 1-4 with (16) result in an algorithm that bears a striking resemblance to FGT [15] (as both are founded on Taylor series expansion of the Green's function; see (6) in [15]). However, it is interesting to note the similarities and differences between the ACE procedure and the FGT [15] for the evaluation of Gaussian kernels. The use of generalized Taylor expansions and Cartesian tensors for Gaussian kernels is essentially a re-formulation of the FGT, where expansions in terms of Hermite polynomials are employed. The cost and storage savings achieved with the use of totally symmetric tensors is identical to that of the graded lexicographic order representation proposed in the improved FGT [17]. An additional advantage of the ACE algorithm over FGT is the use of *exact translation operators* for multipole-to-multipole and local-to-local expansions in Theorems 2 and 4. It is important to note that the accuracy of the hierarchical tree computation is dominated by the number of terms used in the alternate representation of the potential in Eq. (11) and this is *exactly* the same for all three methods. Consequently, the exact upward and downward tree traversal operations of the ACE algorithm allow it to achieve better accuracy for multilevel trees, but only within the desired order of magnitude. Results to this end are not included as a fair comparison between ACE and FGT is not possible using random source-observer distributions.

3.2. Spatio-temporal acceleration methods

It is well known that diffusion, lossy wave, and Klein–Gordon kernels possess an infinite temporal tail. Consequently, the cost complexity of evaluating (8), using only the spatial acceleration scheme detailed in the previous section, is $\mathcal{O}(N_s N_t^2)$. Here, we present two different schemes for ameliorating the remaining quadratic cost complexity in time. Both schemes are formulated in a manner such that *causality* is not violated, i.e. evaluation of a potential at time step $k\Delta_t$ assumes the knowledge of sources at time steps $l\Delta_t$, $l < k$ only. Thus, the proposed acceleration schemes are in conformance with, and can be readily integrated into, an existing integral equation solver.

3.2.1. Method 1: Block-Toeplitz computation and FFT scheme

It is well known that causality prevents the use of the entire space-time matrix (shown in Fig. 1) together with existing FFT schemes for temporal acceleration. Instead, a block-Toeplitz computation can be used to overcome this bottleneck [30] . In this approach, as indicated in Fig. 1, the matrix is divided into sub-matrices $\mathbf{T}_N$, defined as follows

$$\mathbf{T}_N = \begin{bmatrix} \mathbf{Z}_N & \dots & \mathbf{Z}_1 \\ \vdots & \ddots & \vdots \\ \mathbf{Z}_{2N-1} & \dots & \mathbf{Z}_N \end{bmatrix}, \tag{17}$$

Evaluation of the potential $\mathbf{P}_k$ requires the computation of the product between submatrix $\mathbf{T}_N$ and vector $\{\mathbf{S}_{k-N},\dots,\mathbf{S}_{k-1}\}$, for different values of N. Notice that the value of N depends on k and this computation will also result in the partial evaluation of potentials at $N-1$ future time steps. For example, the evaluation of $\mathbf{P}_2$ involves the multiplication of vector $\{\mathbf{S}_0,\mathbf{S}_1\}$ with submatrix $\mathbf{T}_2$, which also results in the *partial computation* of $\mathbf{P}_3$ due to these source vectors. In the next time step, $\mathbf{P}_3$ is completely evaluated with the computation of the matrix vector product $\mathbf{T}_1\mathbf{S}_2$. Since each of the submatrix blocks is Toeplitz, FFT can be used to accelerate the computation. Thus the block computation scheme shown in Fig. 1 allows the fast evaluation of potentials using FFT within a time stepping framework. It is important to note that this acceleration methodology does not involve any approximations and is exact. Thus this scheme can be used to accelerate both *near* and *far* time interactions.

Next, we consider integrating the above acceleration scheme within the context of the ACE algorithm introduced in the previous Section. The following two observations form the basis for integrating the block-Toeplitz based temporal convolution and ACE based spatial acceleration schemes,

(1) Except for the multipole-to-local translation operators, all other operations in ACE depend only on either the source (l) or observer (k) time.
(2) The multipole-to-local translation operation in Theorem 3 preserves the convolution in time.

These statements are easily proved. Consider the evaluation of the potential $\widetilde{\Phi}$ at observation points in domain Ω_o at time $k\Delta_t$ due to sources located in domain Ω_s and excited at $l\Delta_t$. Using Theorem 3 and (15) of the ACE algorithm, we get

$$\widetilde{\Phi}(\mathbf{r}, k\Delta_t) = \sum_{n=0}^{\infty} (\boldsymbol{\rho}_{oi})^{(n)} \cdot n \cdot \mathbf{L}_k^{(n)}, \tag{18a}$$

$$\mathbf{L}_k^{(n)} = \sum_{m=n}^{\infty} \left(\mathbf{M}_l^{(m-n)} \cdot (m-n) \cdot \frac{1}{n!} \mathbf{H}_{k-l}^{(m)} \right), \tag{18b}$$

where, $\mathbf{H}_{k-l}^{(m)} = \nabla^m G(\mathbf{r}_{os}^p, (k-l)\Delta_t)$.

From Eq. (18b) we infer that the evaluation of the local expansions at the k-th time step is equivalent to the evaluation of the potential at that time step. Using this equivalence, (8) is written as

$$\mathbf{P}_k = \sum_{l=0}^{k-1} \sum_{n=0}^{\infty} (\boldsymbol{\rho}_{oi})^{(n)} \cdot n \cdot \left[\sum_{m=n}^{\infty} \left(\mathbf{M}_l^{(m-n)} \cdot (m-n) \cdot \frac{1}{n!} \mathbf{H}_{k-l}^{(m)} \right) \right] \tag{19}$$

$$= \sum_{n=0}^{\infty} (\boldsymbol{\rho}_{oi})^{(n)} \cdot n \cdot \sum_{m=n}^{\infty} \left[\sum_{l=0}^{k-1} \left(\mathbf{M}_l^{(m-n)} \cdot (m-n) \cdot \frac{1}{n!} \mathbf{H}_{k-l}^{(m)} \right) \right]. \tag{20}$$

Evaluation of the ACE local expansion at the k^{th} time instant requires the temporal convolution of multipole $\mathbf{M}_l^{(n)}$ and translation operator $\mathbf{H}_{k-l}^{(n)}$ while performing the tensor contraction. As shown in Fig. 2, each tensor component of the translation operator $\mathbf{H}_{k-l}^{(n)}$ is written in the block-Toeplitz form, where

$$\mathbf{T}_N^{(n)}(n_1, n_2, n_3) = \begin{bmatrix} \mathbf{H}_N^{(n)}(n_1, n_2, n_3) & \dots & \mathbf{H}_1^{(n)}(n_1, n_2, n_3) \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{2N-1}^{(n)}(n_1, n_2, n_3) & \dots & \mathbf{H}_N^{(n)}(n_1, n_2, n_3) \end{bmatrix}. \tag{21}$$

This allows the use of the FFT for the efficient evaluation of the temporal convolution of each of the $(n+1)(n+2)/2$ terms in the tensor contraction. The Fourier transform of a n-th rank Cartesian tensor $\mathbf{M}_l^{(n)}$ is defined as follows

$$\widetilde{\mathbf{M}}^{(n)}(n_1, n_2, n_3; \omega_k) = \mathcal{F}\left\{ \mathbf{M}_l^{(n)}(n_1, n_2, n_3) \right\}, \quad \textit{where } n_1 + n_2 + n_3 = n, \tag{22}$$

where $\mathcal{F}$ denotes the forward Fourier transform operator and $\widetilde{\mathbf{M}}^{(n)}$ denotes the n-th rank Cartesian tensor in Fourier space. Given this definition, the local expansion due to all past sources at the k-th time step, $\mathbf{L}_k^{(n)}$, is evaluated as,

$$\mathbf{L}_k^{(n)}(n_1, n_2, n_3) = \sum_{m=n}^{\infty} \frac{1}{n!} \mathcal{F}^{-1} \left\{ \sum_k \widetilde{\mathbf{H}}^{(m)}(\omega_k) \cdot (m-n) \cdot \widetilde{\mathbf{M}}^{(m-n)}(\omega_k) \right\}, \tag{23}$$

where $n_1 + n_2 + n_3 = n$, and $\mathcal{F}^{-1}$ is the inverse Fourier transform operator. Since the Fourier transform is applied only in the time domain, the tensor contraction definition is valid in the Fourier domain as well.

A significant feature of this approach is that the temporal acceleration is evaluated *exactly*. The accuracy of the overall approach, combining ACE and FFT, is only governed by the order of harmonics used in the ACE algorithm. Hence the block-Toeplitz scheme with the direct evaluation of spatial convolutions can be used for the rapid evaluation of nearfield interactions also.

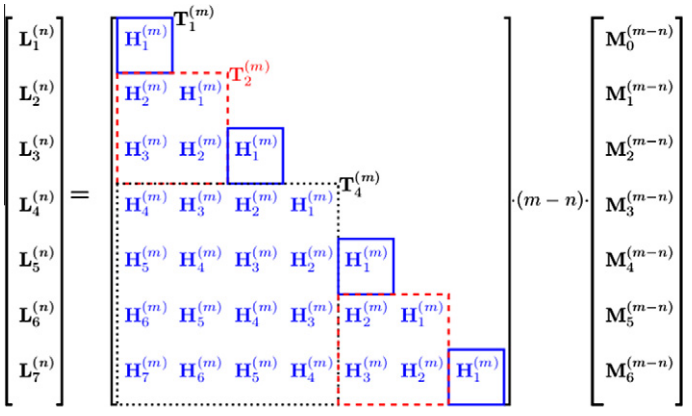

Fig. 2. Illustration of the block-Toeplitz computational scheme in terms of time dependant ACE harmonics.

3.2.2. Method 2: Numerical compression

The FFT based temporal acceleration scheme described in the previous Section is independent of the form of the kernel. We next develop an alternative acceleration scheme that exploits the temporal rank deficiency of the block-Toeplitz matrices, $\mathbf{T}_N^{(n)}$ (see Fig. 2), that arise specifically during the computation of the diffusion, lossy wave, and Klein–Gordon potentials. This scheme also follows the block-Toeplitz strategy to respect the causality requirement.

As mentioned before, the spatio-temporal convolution during the multipole-to-local translation operation can be considered as the evaluation of a matrix-vector product between tensor components of the multipoles, $\left\{\mathbf{M}_k^{(n)}\right\}$, and the block translation operator, $\mathbf{T}_N^{(n)}$. The cost for direct evaluation of this matrix-vector product scales as $\mathcal{O}(N^2)$. However, when the submatrix $\mathbf{T}_N^{(n)}$ is rank deficient, numerical compression schemes can be utilized to reduce the cost to $\mathcal{O}(N)$. Let r be the rank of the matrix (defined in Appendix C), then the numerical compression scheme requires the following decomposition of the matrix

$$\mathbf{T}_N^{(n)}(n_1, n_2, n_3) = \mathbf{UV}, \tag{24}$$

where the matrices $\mathbf{U}$ and $\mathbf{V}$ are of dimension $N \times r$ and $r \times N$, respectively. In rest of the paper, this decomposition of a matrix is referred to as its *UV decomposition*. With such a decomposition the matrix-vector product can be performed at $\mathcal{O}(rN)$ cost. However, the cost associated with the computation of the *UV decomposition* of a matrix can be high and dominate the overall cost. Thus, the use of a numerical compression scheme is advantageous provided that r is sufficiently small (i.e. $r \ll N$) *and* the cost of computing a *UV decomposition* is less than $\mathcal{O}(N^2)$.

All kernels under consideration in this paper have an infinite temporal extent as well as infinitely differentiable/smooth tails. While the lossy wave and Klein–Gordon kernels also have contributions due to 'local' terms proportional to $\delta(t - \|\mathbf{r}\|/c)$, the evaluation of these terms are considered in the Appendices, and here we consider only the portion arising from these infinite temporal tails. These tails can be well-approximated by only a few basis functions in time and hence the rank of the temporal convolution matrix is asymptotically independent of N, a result which is well-supported by previous work on 2D wave propagation [30,33]. Consequently, for problems in which N_t is larger than the rank of a typical submatrix, we can consider the resultant convolution matrices to be low rank. This rank deficiency is demonstrated explicitly for our three potentials of interest in Section 4 and serves as an empirical justification for our first criterion, $r \ll N$, in using a numerical compression scheme.

From (21), it is evident that

$$\mathbf{T}_{2N}^{(n)} = \begin{bmatrix} \mathbf{H}_{2N}^{(n)} & \cdots & \mathbf{T}_N^{(n)} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{4N-1}^{(n)} & \cdots & \mathbf{H}_{2N}^{(n)} \end{bmatrix}. \tag{25}$$

Exploiting the above redundancies in a Toeplitz matrix, we devise a recursive algorithm to compute the *UV decomposition* of $\mathbf{T}_N^{(n)}$ at $\mathcal{O}(Nr^2 \log(N))$ cost to meet our second criterion for using a numerical compression scheme. algorithm are provided in Appendix C. This algorithm is used to construct a compressed factorization of the $\mathbf{T}_N^{(n)}(n_1, n_2, n_3)$ with $2Nr$ entries for each of the $(n+1)(n+2)/2$ tensor components. To this end, it is important to note that the *UV decompositions* should be stored at each time step to take full advantage of the block-Toeplitz structure of the translation operator.

The error incurred in this acceleration scheme is a combination of the error from truncating the expansion of the translation operator, as well as the error incurred during the numerical compression of the translation operator. The compression error is dependent upon the threshold (ϵ) used to determine the rank r of the *UV decompositions*. Consequently, it is optimal to set this parameter such that the error incurred in compression is of the same order as the error incurred in truncating ACE expansions. The specifics of this will be discussed in Section 4.

3.2.3. Multi-level algorithm and cost estimate

The above discussions of the spatio-temporal acceleration schemes pertained to a single level ACE algorithm. Extension to a multilevel algorithm is fairly straight-forward and requires the definition of different sets of $\mathbf{T}_N^{(n)}$ and $\mathbf{M}^{(n)}$, for each level of the tree. The upward and downward tree traversal are performed at each time step. The new multipoles, due to sources excited at that time step, are computed during the upward tree traversal. During the downward tree traversal, potentials at that time instant are computed while *updating* the potentials at a set of future time instants. Following the usual dictum [10,13], the multipole-to-local translation operation is performed for all interacting boxes; where the temporal convolution is accelerated using either the FFT or the numerical compression scheme. Note that both the schemes require the storage of multipole coefficients for the entire simulation time. In the remainder of this section, we discuss the cost associated with both the schemes.

In the FFT and ACE based spatio-temporal convolution, the multipole-to-local translation operation involves the temporal convolution in (18b) that is evaluated in Fourier domain (23) using FFTs. This evaluation is carried out for each of the $(n+1)(n+2)/2$ tensor components. The cost of one upward and downward tree traversal scales as $\mathcal{O}(N_s)$. The cost of computing multipole expansions at all time instants scales as $\mathcal{O}(N_s N_t)$. The cost of multipole-to-local expansion and downward tree traversal for a Toeplitz system of size N scales as $\mathcal{O}(N_s N \log(N))$ and $\mathcal{O}(N_s N)$ respectively. In the entire scheme, Toeplitz systems of size $N_t/2$ occur once, $N_t/4$ occurs twice and so on. Thus the total cost of this scheme scales as

$$\mathcal{O}\left(N_s N_t + N_s\left(\frac{N_t}{2}\left(\log\left(\frac{N_t}{2}\right)+1\right)+2\frac{N_t}{4}\left(\log\left(\frac{N_t}{4}\right)+1\right)+\ldots\right)\right)\approx\mathcal{O}(N_s N_t \log^2(N_t)). \tag{26}$$

The derivation of the total cost of the numerical compression scheme and ACE algorithm is identical to that of the FFT and ACE scheme. The primary difference is that we use *UV decompositions*, instead of FFT operations, to evaluate the temporal convolution during multipole-to-local translation (18b). Evaluation of these translations with rank r approximation of the $N \times N$ Toeplitz sub-matrices requires $\mathcal{O}(Nr^2\log(N))$ operations. The cost analysis derived for the FFT scheme can be used when r is asymptotically independent of N_t. Thus the total cost of this approach is given as,

$$\mathcal{O}\left(N_s N_t + N_s\left(\frac{N_t}{2}\left(r^2\log(\frac{N_t}{2})+1\right)+2\frac{N_t}{4}\left(r^2\log(\frac{N_t}{4})+1\right)+\ldots\right)\right)\approx\mathcal{O}(N_s N_t r^2\log^2(N_t))\approx\mathcal{O}(N_s N_t\log^2(N_t)).$$

Both the FFT and numerical compression based algorithms have similar asymptotic costs with respect to memory as well. For both methods, the entire time history of the source multipole expansions must be stored, requiring $\mathcal{O}(N_s)$ storage per time step, yielding a total cost of $\mathcal{O}(N_s N_t)$. The only other burden on memory is the storage of the translation operators at each time step. For the FFT method, we exploit the block-Toeplitz nature of the matrices making up the translation operator to achieve an $\mathcal{O}(N_s N_t)$ storage requirement. This is due to the fact that any $N \times N$ Toeplitz matrix can be represented in terms of $2N-1$ unique quantities. In the case of the numerical compression method, *UV decompositions* need to be created and stored to represent the components of the translation operators for each time step. While these decompositions aren't themselves Toeplitz, the *UV decomposition* of an $N \times N$ submatrix of rank r requires the storage of only $2Nr$ quantities. Consequently, the storage required for the numerical compression method is $\mathcal{O}(N_s N_t)$, as well. These two primary costs dictate that the total memory requirement for both methods is $\mathcal{O}(N_s N_t)$.

4. Results

In this Section, results are presented to substantiate the above claims and demonstrate the efficacy of the algorithm developed in this work. The results presented here will demonstrate convergence in error as well as the predicted CPU cost scaling. Experiments with temporal degrees of freedom $N_t \leqslant 256$ were executed on a desktop with a 2.3 GHz Intel Pentium processor and 2 GB RAM running Linux OS. Experiments with $N_t > 256$ were executed on a 1.6 GHz Itanium2 processors with 8 GB RAM at the High Performance Computing Center at Michigan State University.

In all cases, the diffusion potential between N_s source/observers pairs was evaluated at one second time intervals, $\Delta_t = 1$ s. The locations of the N_s points were randomly chosen from a uniform distribution and the time signature associated with the n-th source is given by (27)

$$f_n(t) = \kappa_n e^{-(t-t_p)^2/2\sigma^2}, \tag{27}$$

where κ_n is the source strength that was randomly chosen between $[0,1]$, $\sigma = 1.9$ s and $t_p = 6\sigma$. The α in the diffusion Eq. (2) was fixed at unity. These parameters were chosen arbitrarily. The accuracy of the proposed algorithm is validated against direct computation for all cases when the unknown count is numerically small. The relative error at the n-th observer is evaluated as

$$Error_{far}(\mathbf{r}_n) = \sqrt{\frac{\sum_{l=0}^{N_t}|\Phi_{fast,far}(\mathbf{r}_n, l\Delta_t) - \Phi_{direct,far}(\mathbf{r}_n, l\Delta_t)|^2}{\sum_{l=0}^{N_t}|\Phi_{direct,far}(\mathbf{r}_n, l\Delta_t)|^2}}, \tag{28}$$

where $n \in [1, N_s]$, $|\cdot|$ represents the absolute value, $\Phi_{fast,far}(t)$ and $\Phi_{direct,far}(t)$ represent the time history of the fields evaluated using the proposed algorithms and the direct approach, respectively. The final error reported throughout this work is the average error over all N_s observers. Error is computed only with the farfield potential, i.e. direct data is computed *only* for the source/observation pairs that are in the farfield of each other. This represents the worst-case error.

The first set of results demonstrates the exactness of the multipole-to-multipole and local-to-local translation operators of the ACE algorithm, Theorems 2 and 4 respectively. An important implication of this feature of the ACE algorithm is that the error does not increase as the height of the tree is increased. Consider two domains $\Omega_1 \subset \Omega$ and $\Omega_2 \subset \Omega$ of size $(0,0.5)\times(0,0.5)\times(0,0.5)\,\mathrm{m}^3$ and $(1,1.5)\times(1,1.5)\times(1,1.5)\,\mathrm{m}^3$ respectively. In each domain, 4000 source/observer points are randomly distributed and we consider the interaction between these two domains only, all other interactions are neglected. As the number of levels in the tree is increased, the change in the error norm can be attributed solely to the error in multipole-to-multipole and local-to-local translations. Table 1 shows errors computed for trees of varying height with different values of P. For a given accuracy (fixed P) it is evident that, to double precision, there is no variation in the error obtained from using trees of different height. This is a consequence of the fact that Theorems 2 and 4 are exact.

The next set of results in Table 2 pertains to the accuracy of computing the time domain diffusion potential in (6) using the fast algorithm relative to the direct approach. The N_s source/observer points were distributed randomly within a cube of size d and the number of time discretizations was fixed at $N_t = 256$. The order of the ACE expansion, P, was varied from 0 to 9 for different cases of N_s and d, as indicated in Table 2, and the temporal convolution was performed using the FFT scheme. As expected, the error uniformly decreases to double precision as the number of ACE harmonics is increased. The *UV decompo-*

Table 1
Exact translation operator in ACE algorithm, P denotes the order of ACE harmonics.

Levels	$P = 3$	$P = 6$
3	5.343762051614 770E-006	7.0123317207 40178E-008
4	5.343762051614 130E-006	7.0123317207 70545E-008
5	5.343762051614 279E-006	7.0123317207 63711E-008

Table 2
Error convergence for different order of ACE harmonics (P) and different source/observer configuration (N_s, d).

	N_s, d			
P	8000, 1.0	4000, 1.0	8000, 0.5	4000, 0.5
0	4.92E−02	5.01E−02	1.39E−02	1.74E−02
1	1.12E−02	1.04E−02	3.71E−03	4.70E−03
3	9.61E−05	9.39E−05	1.22E−05	2.21E−05
5	1.20E−06	1.36E−06	3.54E−08	9.04E−08
7	1.87E−08	2.28E−08	1.20E−10	3.30E−10
9	2.77E−10	3.44E−10	6.40E−13	1.21E−12

Table 3
Numerical rank of an $N \times N$ Toeplitz matrix for a desired accuracy of $\epsilon = \mathcal{O}(10^{-5})$ using SVD, QR, and MGS.

N	r_{SVD}	r_{QR}	r_{MGS}
16	5	5	6
32	6	6	10
64	7	7	11
128	8	7	11
256	9	8	12
512	10	9	12

sition can be used to achieve the same error convergence provided that the precision in determining the rank is fixed appropriately.

Table 3 exhibits the numerical rank of some of the Toeplitz sub-matrices of different sizes, N, that form the translation operator for the diffusion kernel. The precision used in determining the rank using (C.1) was fixed at $\epsilon = \mathcal{O}(10^{-5})$. The different matrix decomposition methods used to obtain the *UV decomposition* in (C.2) are DGESVD from LAPACK [41], DGEQPX from RRQR package (based upon ACM algorithm 782) [42] and modified Gram-Schmidt (MGS) with partial pivoting. The MGS routine was developed in-house. As seen from Table 3, all three methods produce approximately the same rank for a fixed accuracy. This validates our implementation of the MGS method to compute the rank and QR decomposition of a given matrix. In general, we found the MGS routine to be the fastest among the selected procedures. In the rest of the results, the MGS routine is used to compute the *UV decomposition*.

Next, we illustrate the temporal rank deficiency of the translation operators for the diffusion, lossy wave, and Klein–Gordon kernels. For all kernels, $\alpha = 1$ and for the lossy wave and Klein–Gordon kernels, $c = 1$. In the case of the lossy wave and Klein–Gordon kernels, the Delta distribution term is ignored as its contribution to the potential is accounted for separately (see Appendix A, B). Hence we consider only the tails of the associated Green's functions. Figs. 3–5 shows the average rank of the nearfield interactions between 1000 point sources randomly distributed in a unit cube for $N_t = 1024$ time steps. The nearfield interaction was chosen to demonstrate the following

(1) Upper bounds on ranks for spatial farfield interactions.
(2) Utility of the *UV decomposition* approach for spatial nearfield interactions that will not be accelerated by ACE.

To elaborate on the first point, since all Green's functions decay in space, as well as time, we expect lower ranks in the spatial farfield as their tails become increasingly smooth. While nearfield interactions are not subject to acceleration using ACE, the kernels themselves are representative of the $p = 0$ component of the translation operator, so this behavior will be indicative of an upper bound on rank deficiency for spatial farfield interactions. It can be seen from Figs. 3–5 that as N varies from 16 to 512 the ranks do not vary much. The most significant variations in rank occur for the Klein–Gordon kernel, which is to be expected as, while its tail follows a decaying envelope in time, it is also oscillatory. Later, we will show that the average rank of the translation operator for the diffusion equation is quasi-independent of N_t. Additionally, we note that high accuracy can be achieved for all kernels for very low numerical rank (i.e. high compression). For the lossy wave and diffusion kernels, at an accuracy of $\mathcal{O}(10^{-10})$ we see a factor of $\approx$10 improvement in the number of operations required for 512×512

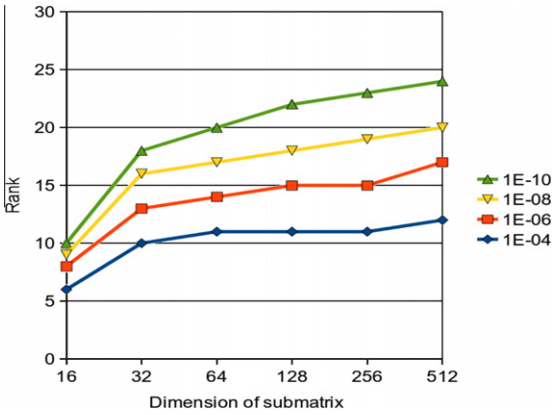

Fig. 3. Average numerical ranks of block-Toeplitz sub-matrices for the nearfield contribution to the diffusion potential over $N_t = 1024$ time steps.

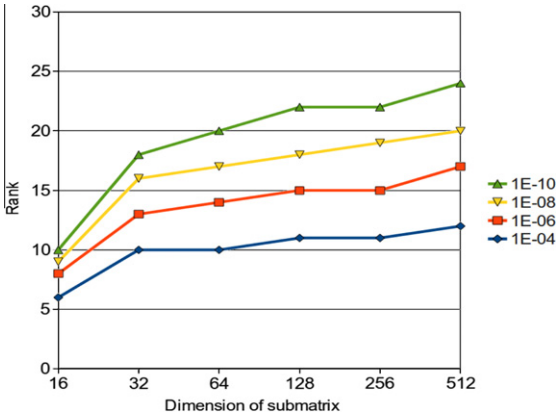

Fig. 4. Average numerical ranks of block-Toeplitz sub-matrices for the nearfield contribution to the lossy wave potential over $N_t = 1024$ time steps.

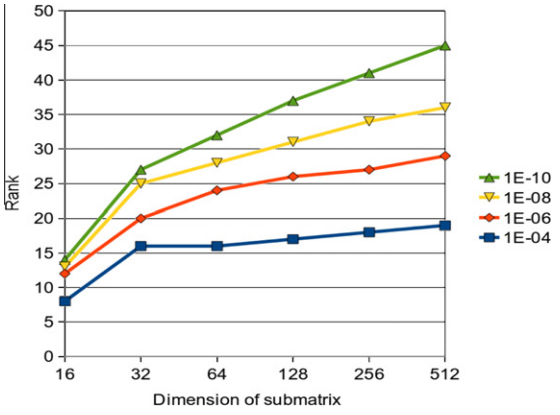

Fig. 5. Average numerical ranks of block-Toeplitz sub-matrices for the nearfield contribution to the Klein–Gordon potential over $N_t = 1024$ time steps.

matrix-vector multiplication, and for an accuracy of $\mathcal{O}(10^{-5})$ we see an improvement factor of ≈21. For the Klein–Gordon kernel, these factors are reduced to ≈5 and ≈13, respectively.

Table 4 is an extension of the above experiment to demonstrate the variation of the average rank of the nearfield MOT matrices with respect to the parameter α. For the diffusion and lossy wave kernels, the average ranks of submatrices of all sizes are demonstrated to be independent of α over three orders of magnitude. However, this is not the case for the

Table 4
Numerical ranks of block-Toeplitz sub-matrices of a 1024×1024 translation operator for diffusion (D), lossy wave (LW), and Klein–Gordon (KG) potentials evaluated for different values of α with $\epsilon = \mathcal{O}(10^{-6})$.

	$\alpha = 10^0$			$\alpha = 10^{-1}$			$\alpha = 10^{-2}$		
N	D	LW	KG	D	LW	KG	D	LW	KG
16	8	8	12	8	8	8	8	8	8
32	13	13	20	13	13	13	13	13	13
64	14	14	24	14	13	16	14	14	14
128	15	15	26	15	15	20	15	15	16
256	15	15	27	15	16	25	15	16	17
512	17	17	29	17	17	26	17	17	19

Table 5
Time for different problem size (N_s) within a cube of sidelength $d = 0.5$ m. In all cases $N_t = 256$, $P = 3$ and $\epsilon = \mathcal{O}(10^{-5})$).

N_s	T_{FFT}	T_{UV}	$rank_{avg}$	T_{Direct}
4000	71.98	81.45	7.93	17195.35
8000	98.32	102.95	7.93	68781.25
16000	226.8	170.07	7.93	275125.60
32000	472.66	454.16	7.93	1100502.40
64000	978.07	928.21	8.07	–
128000	2282.01	2036.79	7.93	–
256000	4652.78	–	–	–

Table 6
T_{fast} for different N_t size. In all cases $N_s = 8{,}000$, $d = 0.5$, $P = 3$ ($\epsilon = \mathcal{O}(1E-5)$).

N_t	T_{FFT}	T_{UV}	$rank_{avg}$	T_{Direct}
256	94.17	121.14	8.07	44169.36
512	231.95	299.11	8.19	176677.44
1024	554.57	699.46	8.25	706709.76
2048	1395.52	1641.67	8.28	2826839.04
4096	3438.00	3892.61	8.30	–
8192	9494.96	9232.98	8.31	–
16384	28410.84	20511.44	8.31	–

Klein–Gordon kernel, where the average ranks are found to be lowest for the smallest value of α. We attribute this to the oscillatory nature of the kernel's tail. As α decreases, the kernel oscillates more slowly and the rank of the associated matrices decreases accordingly. This demonstrates a limitation of our numerical compression method as this method is not efficient when the parameter α is such that the submatrices have a relatively high rank. In such cases, the FFT acceleration scheme is clearly the optimal choice as it can be utilized for any $\alpha \neq 0$.

The time taken by the proposed algorithms, utilizing the FFT and the *UV decomposition* schemes, are compared with direct computation in Tables 5 and 6. In Table 5, we present the run-times for evaluating time domain diffusion potential for different values of N_s while the following parameters were kept as constants: total number of time discretizations, $N_t = 256$, size of domain, $d = 0.5$, and order of ACE expansions, $P = 3$, corresponding to an error of $\mathcal{O}(1E-5)$. It can be seen that the proposed algorithm is 230 times faster than the direct method even for a small problem size of $N_s = 4000$ and $N_t = 256$. Here we note that direct evaluation was explicitly computed for the smallest problem in each table and was then extrapolated to larger problems, assuming a quadratic scaling. Fig. 6 shows a log-scale graph of T_{fast} vs N_s for values in Table 5, where T_{fast} denotes either T_{FFT} or T_{UV}. The lines in the graph correspond to a least-square linear fit the slopes of which are found to be $\approx$1.1. This validates the $\mathcal{O}(N_s)$ cost scaling of the proposed algorithm.

In Table 6 the run-time of the proposed algorithms are shown for different values of N_t for both the FFT and the *UV decomposition* approaches. Aside from N_t, all other parameters were kept constant at $N_s = 8{,}000$, $d = 0.5$ m, and $P = 3$. When using the FFT approach, the expected $\mathcal{O}(N_t \log^2(N_t))$ scaling cannot be verified because of the following implementation details. In our work we use the FFTW3 [43], an open source software, to compute the FFT and this does not exhibit uniform $N_t \log(N_t)$ scaling. This is the case with many other performance oriented FFT packages as well. However, the $\mathcal{O}(N_t \log^2(N_t))$ scaling of the *UV decomposition* method is evident. Moreover, it was observed that it is more efficient (faster) to use direct evaluation rather than the FFT procedure for Toeplitz systems of smaller sizes. This is important in terms of overall performance as smaller Toeplitz systems occur more frequently, i.e. Toeplitz systems of size 1×1 and 2×2 occur $N_t/2$ and $N_t/4$ times, respectively. In this work direct evaluation was used for any Toeplitz matrix of size $\leqslant 16 \times 16$. The optimal size for direct evaluations may vary based on the computer platform and FFT library in use. It is also worth noting that while the FFT approach outperforms *UV decomposition* method by a factor of $\approx$1.3 for $N_t = 256$, the two methods require approximately the same time for $N_t = 8{,}192$, representing a break-even point between the two algorithms. Furthermore, the *UV decomposition*

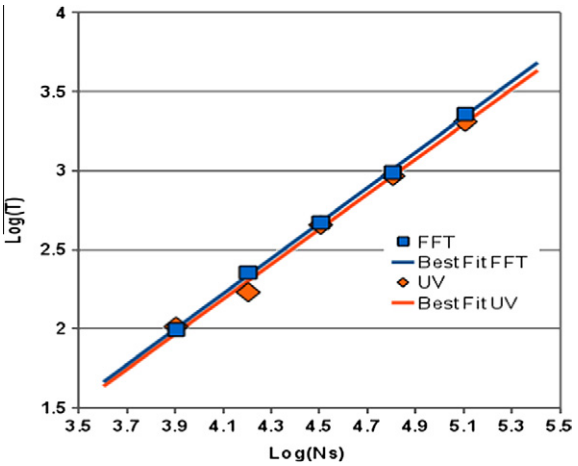

Fig. 6. Plot of *log* T_{fast} vs. *log* N_s from Table 5 where T_{fast} denotes either T_{FFT} or T_{UV}; slope of linear fit ≈ 1.1 for both.

approach outperforms the FFT approach for $N_t = 16{,}384$ by a factor of $\approx$1.4. This demonstrates that while the FFT approach is preferred for shorter simulations, the *UV decomposition* approach is optimal for longer simulations.

5. Conclusions

In this paper, we have detailed a generic framework to accelerate the evaluation of potentials due to the diffusion, lossy wave, and Klein–Gordon equations. Two methodologies have been introduced to reduce the computational cost associated with the spatio-temporal convolutions from $\mathcal{O}\left(N_s^2 N_t^2\right)$ to $\mathcal{O}(N_s N_t \log^2(N_t))$. Both algorithms exploit the block-Toeplitz structure of the temporal convolution matrices to fit within a MOT framework, and utilize ACE expansions to derive an alternate form of the Green's function which is used in the acceleration of the requisite spatial convolutions. The first method naturally combines the FFT with the ACE algorithm. The second method, utilizes a numerical compression scheme which capitalizes on the temporal rank deficiency of the temporal convolution matrices, also in conjunction with ACE. To this effect, we have also developed an improved scheme for the efficient evaluation of the *UV decomposition* of a Toeplitz matrix. The memory requirements of both the methods scale as $\mathcal{O}(N_s N_t)$. Several results have been presented to validate the utility and efficiency of the proposed scheme. Presently, work is underway to extend the FFT method presented in this paper to the evaluation of retarded wave potentials, as well. In a future paper, we will explore the extension of this method to time domain electromagnetics problems, wherein this method serves as the basis for a manifestly multiscale alternative to the popular PWTD algorithm.

Acknowledgements

We thank NSF(CCF-0729157, DMS-0811197) for generously funding our research and high performance computing (HPC) center at Michigan State University and National Center for Supercomputing Applications (NCSA) at University of Illinois at Urbana-Champaign (UIUC) for providing the computational resources. Finally, we thank the three anonymous reviewers, whose comments were invaluable in the refinement of this paper.

Appendix A. ACE translation operator for lossy wave kernels

The determination of $\nabla^p G(\mathbf{r},t) = \mathbf{H}^{(p)}$, where G is defined in (4a), for the multipole-to-local translation is slightly more complicated for the lossy wave kernel. Transforming the lossy wave Eq. (1) from the time domain into the frequency domain via Fourier transform yields the Helmholtz equation with a complex value of $k^2 = \frac{\omega^2}{c^2} - i\omega\alpha^2$. In [27], it was shown that the expansion of the Helmholtz kernel is given as:

$$\begin{aligned}\mathbf{H}^{(p)}_{Helmholtz}(p_1,p_2,p_3) &= \partial_x^{p_1}\partial_y^{p_2}\partial_z^{p_3}\left(\frac{e^{-ikr}}{r}\right)\\ &= \sum_{m_1=0}^{\lfloor \frac{p_1}{2} \rfloor}\sum_{m_2=0}^{\lfloor \frac{p_2}{2} \rfloor}\sum_{m_3=0}^{\lfloor \frac{p_3}{2} \rfloor}(-1)^{p+m} r^{2m-2p-1} \times \frac{p_1!}{m_1!(p_1-m_1)!}\frac{p_2!}{m_2!(p_2-m_2)!}\\ &\quad \times \frac{p_3!}{m_3!(p_3-m_3)!} r_x^{p_1-2m_1} r_y^{p_2-2m_2} r_z^{p_3-2m_3}\mathcal{G}_{\tilde{p}}(ikr). \end{aligned} \tag{A.1}$$

Here, $p_i \epsilon \{0, 1, ..p\}$, $\sum_{i=1}^{3} p_i = p$, $\mathcal{G}_{\tilde{p}}(ikr) = \sqrt{2/\pi(ikr)^{\tilde{p}}} k_{\tilde{p}}(ikr)$, where $k_{\tilde{p}}$ is the $\tilde{p}$th order modified spherical Bessel function of the second type and $\tilde{p} = p - m$. Using recurrence relations between the Bessel functions [44], we can derive the following recursive expression in the frequency domain.

$$\mathcal{G}_{\tilde{p}+1}(ikr) = r^2 \left(\frac{\omega^2}{c^2} - i\omega\alpha^2 \right) \mathcal{G}_{\tilde{p}-1}(ikr) + 2(\tilde{p}+1)\mathcal{G}_{\tilde{p}}(ikr). \tag{A.2}$$

Taking the inverse Fourier transform of (A.1), we are left with an expression for the time domain lossy wave translation operator.

$$\mathbf{H}^{(p)}(p_1, p_2, p_3) = \sum_{m_1=0}^{\lfloor \frac{p_1}{2} \rfloor} \sum_{m_2=0}^{\lfloor \frac{p_2}{2} \rfloor} \sum_{m_3=0}^{\lfloor \frac{p_3}{2} \rfloor} (-1)^{p+m} r^{2m-2p-1} \times \frac{p_1!}{m_1!(p_1-m_1)!} \frac{p_2!}{m_2!(p_2-m_2)!} \times \frac{p_3!}{m_3!(p_3-m_3)!} r_x^{p_1-2m_1} r_y^{p_2-2m_2} r_z^{p_3-2m_3} G_{\tilde{p}}(\mathbf{r}, t). \tag{A.3a}$$

Here we can derive the $G_{\tilde{p}}(\mathbf{r}, t)$ terms of the sum by expressing the recursive relation in (A.2) in the time domain as:

$$G_{\tilde{p}+1}(\mathbf{r}, t) = r^2 \left(-\frac{1}{c^2} \partial_t^2 - \alpha^2 \partial_t \right) G_{\tilde{p}-1}(\mathbf{r}, t) + 2(\tilde{p}+1) G_{\tilde{p}}(\mathbf{r}, t), \tag{A.4}$$

Where the $G_0(\mathbf{r}, t)$ is the 3D Green's function given in (4a) and $G_{-1}(\mathbf{r}, T)$ is the 1D Green's function [2]

$$G_{-1}(\mathbf{r}, t) = 2\pi c e^{-\frac{\alpha^2 c^2 t}{2}} I_0 \left[\frac{\alpha^2 c^2}{2} \sqrt{t^2 - (\|\mathbf{r}\|/c)^2} \right] \Theta(t - \|\mathbf{r}\|/c). \tag{A.5}$$

As is evident from Eq. (A.4), deriving higher order components of the translation operator will involve taking derivatives of Heaviside and Delta distributions. The resultant translation operator can then be written as:

$$\mathbf{H}^{(p)} = \mathbf{\Lambda}^{(p)} + \mathbf{\Sigma}^{(p)}, \tag{A.6}$$

$$\mathbf{\Lambda}^{(p)}(p_1, p_2, p_3) = \sum_{m=0}^{p} \xi_m(\mathbf{r}, t) \delta^{(m)}(t - \|\mathbf{r}\|/c), \tag{A.7}$$

$$\mathbf{\Sigma}^{(p)}(p_1, p_2, p_3) = \psi(\mathbf{r}, t) \Theta(t - \|\mathbf{r}\|/c). \tag{A.8}$$

Here, $\xi_m(\mathbf{r}, t)$ contains the terms that arise in $\nabla^p G(\mathbf{r}, t)$ which are coefficient to the $m-$ th derivative of a Delta distribution, while $\psi(\mathbf{r}, t)$ contains all of the terms coefficient to a Heaviside distribution. This decomposition separates the translation operator into terms that contain an infinite temporal tail ($\mathbf{\Sigma}^{(p)}$), and those that do not ($\mathbf{\Lambda}^{(p)}$). The FFT and numerical compression methods outlined in this paper can be used to accelerate the evaluation of the terms in the potential which arise due to the infinite temporal tail present in $\mathbf{\Sigma}^{(p)}$, even for arbitrarily small values of α. However, we must augment our algorithm to include contributions due to $\mathbf{\Lambda}^{(p)}$, as they possess no such smooth tail. We instead handle these contributions using methods similar to those given in [29], wherein the ACE algorithm is used to accelerate the computation of the retarded potential in a similar MOT framework. It is significant to note that this is the same problem that arises in the $\alpha = 0$ limit of Eq. (1), i.e. the lossless wave equation, in which $\mathbf{\Sigma}^{(p)} = 0$. For contributions of this nature, we find that our numerical compression scheme is of limited utility, given the lack of a smooth tail. However, as these terms will still spawn matrices which are Toeplitz in time, our FFT-based scheme can still be used. Our group is in fact presently working on adapting this method to time domain electromagnetics problems as a manifestly multiscale alternative to the methods employed in [29].

One further difficulty that should be commented on, is the proper treatment of singular contributions to the potential due to the Delta distribution and its derivatives, found in $\mathbf{\Lambda}^{(p)}$. We can accommodate these contributions through the use of smooth temporal basis functions, upon which the offending singularities can be distributed in temporal convolutions. A full discussion of this procedure, using Lagrange polynomials, is provided in [29].

Appendix B. ACE translation operator for Klein–Gordon kernels

A similar transformation can be made in deriving an expression for $\nabla^p G(\mathbf{r}, t)$, where G is defined in (4c), for the Klein–Gordon equation. We Fourier transform Eq. (3) in time, yielding a Helmholtz equation with $k^2 = \frac{\omega^2}{c^2} - \alpha^2$. Using the same expression for $\mathbf{H}^{(p)}_{Helmholtz}$ and recurrence relation between Bessel functions as before, we arrive at a similar recursive relationship in the frequency domain:

$$\mathcal{G}_{\tilde{p}+1}(ikr) = r^2 \left(\frac{\omega^2}{c^2} - \alpha^2 \right) \mathcal{G}_{\tilde{p}-1}(ikr) + 2(\tilde{p}+1)\mathcal{G}_{\tilde{p}}(ikr). \tag{B.1}$$

We can now re-express $\nabla^p G(\mathbf{r}, t)$ as in (A.3a), where each term can be calculated using the following time domain expression:

$$G_{\tilde{p}+1}(\mathbf{r}, t) = r^2 \left(-\frac{1}{c^2} \partial_t^2 - \alpha^2 \right) G_{\tilde{p}-1}(\mathbf{r}, t) + 2(\tilde{p}+1) G_{\tilde{p}}(\mathbf{r}, t), \tag{B.2}$$

Where $G_0(\mathbf{r},t)$ is the 3D Green's function given in (4c), and $G_{-1}(\mathbf{r},t)$ is the 1D Green's function [2]

$$G_{-1}(\mathbf{r},t) = 2\pi c J_0(\alpha c\sqrt{t^2 - (\|\mathbf{r}\|/c)^2})\Theta(t - (\|\mathbf{r}\|/c)). \tag{B.3}$$

It is again evident that deriving higher order components of the translation operator using Eq. B.2 will involve taking derivatives of distributions. As with the lossy wave kernel, we can split the resultant translation operator into $\mathbf{\Sigma}^{(p)}$ and $\mathbf{\Lambda}^{(p)}$, and proceed as before (removed sentence on numerical compression being less important for KG- this is now discussed briefly in results).

Appendix C. Efficient UV decomposition of Toeplitz matrices

Let $\mathbf{A}$ denote an arbitrary matrix of size $N \times N$ and let σ_i denote its N singular values, ordered such that $\sigma_i > \sigma_{i+1}$. For a desired precision, ϵ, the rank of the matrix $\mathbf{A}$ is r if

$$\sigma_r/\sigma_1 < \epsilon. \tag{C.1}$$

The matrix is said to be rank deficient if $r < N$ and when $r \ll N$ it be represented in an approximate form as

$$\mathbf{A} \approx \mathbf{UV}, \tag{C.2}$$

where $\mathbf{U}$ and $\mathbf{V}$ are suitable matrix decompositions with dimension $N \times r$ and $r \times N$, respectively. It is evident that the error in this approximation is governed by the precision ϵ used in determining the rank. In the rest of the paper, this factorization is referred to as a *UV decomposition*. Such a decomposition can be computed using one of several available rank-revealing matrix factorization methods. This representation requires the storage of only $2Nr$ elements.

Let $\mathbf{S}$ be a vector of size N, then the matrix-vector product $\mathbf{AS}$ can be computed in an efficient manner as

$$\textit{Step } 1: \quad \mathbf{y} = \mathbf{VS}, \tag{C.3a}$$

$$\textit{Step } 2: \quad \mathbf{AS} \approx \mathbf{Uy}. \tag{C.3b}$$

The cost associated with each step is $\mathcal{O}(Nr)$. This, however, is not the dominant cost in constructing a *UV decomposition*, as the cost associated with rank-revealing matrix factorizations is typically greater than $\mathcal{O}(N^2)$. SVD requires $\mathcal{O}(N^3)$ operations and rank-revealing QR decompositions scale as $\mathcal{O}(N^2 r)$. We can circumvent this limitation by taking advantage of the Toeplitz structure of the entire system which implies repetition of some sub-structures. Specifically, we develop a recursive procedure to construct a *UV decomposition* for $\mathbf{A}$ based upon the assumption that we have previously computed *UV decompositions* of Toeplitz submatrices above the diagonal, as would be the case in the MOT scheme in this paper.

We first divide the entire $N \times N$ system, $\mathbf{A}$, into $b \times b$ irreducible sub-matrices, $\{\mathbf{B}_i | i \in 1,\ldots,\gamma^2\}$, where $\gamma = N/b$. A rank-revealing factorization of each irreducible submatrix is computed such that $\mathbf{B}_i \approx \mathbf{U}_i\mathbf{V}_i$. We merge these irreducible sub-matrices in groups of four to construct *UV decompositions* of $\gamma^2/2$ sub-matrices of dimension $2b \times 2b$ via merging operations outlined in [9]. Here we take advantage of the Toeplitz nature of $\mathbf{A}$ by noting that the diagonal blocks are equal, as illustrated by the merging operation below.

$$\left(\begin{array}{c|c} \mathbf{U}_i\mathbf{V}_i & \mathbf{U}_{i-1}\mathbf{V}_{i-1} \\ \hline \mathbf{U}_{i+1}\mathbf{V}_{i+1} & \mathbf{U}_i\mathbf{V}_i \end{array}\right) \rightarrow \left(\tilde{\mathbf{U}}_L\tilde{\mathbf{V}}_L \mid \tilde{\mathbf{U}}_R\tilde{\mathbf{V}}_R\right) \rightarrow \mathbf{UV} \tag{C.4}$$

We continue merging *UV decompositions* of larger and larger sub-matrices until we recover a single *UV decomposition* of $\mathbf{A}$ itself. Here, we choose $b \sim \mathcal{O}(r)$ so that we as to take advantage of rank deficiency in merging the blocks as well. Also, it is important to exploit all redundancies resulting from the Toeplitz nature of $\mathbf{A}$. This will be explicitly noted in the following cost analysis.

As $\mathbf{A}$ is Toeplitz, only $3\gamma/2 - 1$ of the γ^2 irreducible sub-matrices are unique. *UV decompositions* of these sub-matrices are computed using a rank-revealing decomposition with $\mathcal{O}(b^2 r)$ cost, where r is the maximum rank. The resultant factorizations are merged to form $3\gamma/4 - 1$ unique *UV decompositions* of dimension $2b \times 2b$ at $\mathcal{O}(2br^2)$ cost per merge. Recursively repeating this merging process over $L = \log_2(\gamma)$ steps will result in the *UV decomposition* of $\mathbf{A}$. The costs associated with the jth step of this process involves merging $(3\gamma/2^j - 1)$ unique *UV decompositions*. Assuming vertical merges are performed first, there will be $(3\gamma/2^j - 1)$ vertical merges and $(3\gamma/2^{j+1} - 1)$ horizontal merges and the cost of each merging operation is $\mathcal{O}(2^j br^2)$. Thus, the total cost of computing the *UV decomposition* is

$$\left(\frac{3\gamma}{2} - 1\right)\mathcal{O}(br^2) + \sum_{j=1}^{L}\left(\frac{3\gamma}{2^j} - 1 + \frac{3\gamma}{2^{j+1}} - 1\right)\mathcal{O}(2^j br^2) \sim \mathcal{O}(Nr^2\log(N)).$$

It is evident that this is faster than the straightforward $\mathcal{O}(N^2)$ approaches. While the cost of matrix-vector multiplication itself will scale as $\mathcal{O}(Nr^2)$, the process of generating an appropriate *UV decomposition* will dominate the cost, at a complexity of $\mathcal{O}(Nr^2\log(N))$.

References

[1] A. Tamburrino, R. Fresa, S. Udpa, Y. Tian, Three-dimensional defect localization from time-of-flight/eddy current testing data, IEEE Trans. Magnetics 40 (2) (2004) 1148–1151.
[2] P.M. Morse, H. Feshbach, Methods of Theoretical Physics, McGraw-Hill Science/Engineering/Math, 1953.
[3] W. Boettinger, J. Warren, C. Beckermann, A. Karma, Phase-simulation of solidification, Annu. Rev. Mater. Res. 32 (2002) 163–194.
[4] D. Arifin, L. Lee, C. Wang, Mathematical modeling and simulation of drug release from microspheres: Implications to drug delivery systems, Adv. Drug Delivery Rev. 58 (2006) 1274–1325.
[5] A.E. Yilmaz, D.S. Weile, B. Shanker, J.-M. Jin, E. Michielssen, Fast analysis of transient scattering in lossy media, IEEE Antennas Wirel. Propag. Lett. 1 (2002) 14–17.
[6] K. Huang, Quantum Field Theory: From Operators to Path Integrals, John Wiley & Sons, 1998.
[7] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys. 60 (1985) 187–207.
[8] 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.
[9] S. Kapur, D.E. Long, Ies3: Efficient electrostatic and electromagnetic simulation, IEEE Computational Science and Engineering 5 pp. 60–67.
[10] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the laplace equation in three dimensions, Acta Numer. 6 (1997) 229–269.
[11] M. Griebel, S. Knapek, G. Zumbusch, Numerical Simulation in Molecular Dynamics, Springer, Berlin, Heidelberg, 2007.
[12] K. Nabors, W. Jacob, Fast capacitance extraction of general three dimensional structures, IEEE Trans. MTT 40 (1992) 1496–1505.
[13] M. Vikram, B. Shanker, An incomplete review of fast multipole methods -from static to wideband -as applied to problems in computational electromagnetics, Appl. Comput. Electromagn. Soc. J. 24 (2) (2009) 79–108.
[14] N. Nishimura, Fast multipole accelerated boundary integral equation methods, Appl. Mech. Rev. 55 (2002) 299–324.
[15] L. Greengard, J. Strain, The fast gauss transform, SIAM J. Sci. Stat. Comput. 12 (1) (1991) 79–94.
[16] J. Strain, Fast adaptive methods for the free-space heat equation, SIAM J. Sci. Statist. Comput. 15 (1992) 186–206.
[17] C. Yang, R. Duraiswami, N.A. Gumerov, L. Davis, Improved Fast Gauss Transform and Efficient Kernel Density Estimation, Proceedings of the Ninth IEEE International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA,, 2003. pp. 464.
[18] L. Greengard, J. Strain, A fast algorithm for the evaluation of heat potentials, Comm. Pure Appl. Math. 43 (8) (1990) 949–963.
[19] J. Tausch, A fast method for solving the heat equation by layer potentials, J. Comput. Phys. 224 (2) (2007) 956–969.
[20] L. Greengard, P. Lin, Spectral approximation of the free-space heat kernel, Appl. Comput. Harmon. Anal. 9 (1) (2000) 83–97.
[21] J.-R. Li, L. Greengard, On the numerical solution of the heat equation i: fast solvers in free space, Journal of Computational Physics.
[22] P. Jiang, E. Michielssen, Temporal acceleration of time-domain integral-equation solvers for electromagnetic scattering from objects residing in lossy media, Microw. Opt. Technol. Lett. 44 (2005) 223–230.
[23] P. Jiang, Time domain integral equation-based methods for analyzing electromagnetic scattering from objects residing in lossy media, Ph.D. thesis, University of Illinois at Urbana Champaign (2007).
[24] B. Shanker, H. Huang, Accelerated Cartesian expansions - a fast method for computing of potentials of the form R^{-v} for all real v, J. Comput. Phys. 226 (2007) 732–753.
[25] J. Zheng, R. Balasundaram, S.H. Gehrke, G.S. Heffelfinger, W.A.G.S. Jiang III, Cell multipole method for molecular simulations in bulk and confined systems, J. Chem. Phys. 118 (12) (2003) 5347–5355.
[26] F. Zhao, An $\mathcal{O}(n)$ algorithm for three-dimensional n-body simulation, Masters thesis, Massachusetts Institute of Technology.
[27] M. Vikram, H. Griffith, H. Huang, B. Shanker, Accelerated Cartesian Harmonics for Fast Computation of Time and Frequency Domain Low-Frequency Kernels, Antennas and Propagation International Symposium, IEEE, 2007. pp. 5607–5610.
[28] M. Vikram, H. Huang, B. Shanker, T. Van, A novel wideband fmm for fast integral equation solution of multiscale problems in electromagnetics, Antennas and Propagation, IEEE Trans. on 57 (7) (2009) 2094–2104.
[29] M. Vikram, B. Shanker, Fast evaluation of time domain fields in sub-wavelength source/observer distributions using accelerated Cartesian expansions (ACE), J. Comput. Phys. 227 (2007) 1007–1023.
[30] M. Lu, K. Yegin, B. Shanker, E. Michielssen, Fast time domain integral equation solvers for analyzing two-dimensional scattering phenomena, part i: temporal acceleration, Electromagnetics 24 (6) (2004) 425–449.
[31] A.A. Ergin, B. Shanker, E. Michielssen, Fast evaluation of three-dimensional transient wave fields using diagonal translation operators, J. Comput. Phys. 146 (1) (1998) 157–180.
[32] A. Ergin, B. Shanker, E. Michielssen, The plane-wave time-domain algorithm for the fast analysis of transient wave phenomena, Antennas Propagation Mag. IEEE 41 (4) (1999) 39–52.
[33] M. Lu, J. Wang, A. Ergin, E. Michielssen, Fast evaluation of two-dimensional transient wave fields, J. Comput. Phys. 158 (2000) 161–185.
[34] B. Shanker, A.A. Ergin, E. Michielssen, The Multilevel Plane Wave Time Domain Algorithm for the Fast Analysis of Transient Scattering Phenomena, IEEE Antennas and Propagation Society International Symposium, Vol. 2, IEEE, Orlando, FL, 1999. pp. 1342–1345.
[35] B. Shanker, A.A. Ergin, K. Aygün, E. Michielssen, Analysis of transient electromagnetic scattering phenomena using a two-level plane wave time domain algorithm, IEEE Trans. Antennas Propagation 48 (2000) 510–523.
[36] B. Shanker, A.A. Ergin, E. Michielssen, Analysis of transient scattering from penetrable bodies using using the multilevel plane wave time domain algorithm, J. Opt. Soc. Amer. 19 (2002) 716–726.
[37] M. Lu, B. Shanker, E. Michielssen, Fast time domain integral equation solvers for analyzing two-dimensional scattering phenomena; part ii: full pwtd acceleration, Electromagnetics 24 (2004) 451–470.
[38] M. Lu, M. Lv, A.A. Ergin, B. Shanker, E. Michielssen, Multilevel plane wave time domain-based global boundary kernels for two-dimensional finite difference time domain simulations, Radio Science 39.
[39] A.E. Yilmaz, J.M. Jin, E. Michielssen, Time domain adaptive integral method for surface integral equations, IEEE Trans. Antennas Propagation 52 (10) (2004) 2692–2708.
[40] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (2) (1999) 468–498.
[41] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J.D. Cruz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, Society for Industrial and Applied Mathematics, 1999, pp. 239.
[42] C. Bischof, G. Quintana-Orti, Computing rank-revealing qr factorizations of dense matrices, ACM Trans. Math. Softw. 24 (2) (1998) 226–253.
[43] The Design and Implementation of FFTW3, Vol. 93.
[44] G. Watson, A Treatise on the Theory of Bessel Functions, Cambridge University Press, 1922.