A parallel fast algorithm for computing the Helmholtz integral operator in 3-D layered media

Min Hyung Cho, Wei Cai

Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC 28223-0001, USA
Institute of Natural Sciences, Shanghai Jiaotong University, Shanghai 200240, PR China

Article history:
Received 17 September 2011
Received in revised form 12 April 2012
Accepted 13 May 2012
Available online 26 May 2012

Keywords:
Helmholtz equation
Layered media
Green’s function
Window function
Fast multipole method
Tree-code

Abstract
In this paper, we propose a parallel fast algorithm for computing the product of the discretized Helmholtz integral operator in layered media and a vector in $O(N_q N x N y N z \log N x N y)$ operations. Here $N x N y N z$ is the number of sources and $N q$ is the number of quadrature points used in the evaluation of the Sommerfeld integral in the definition of layered media Green’s function (for problems in thin-layer media, $N z = O(1)$). Such a product forms the key step of many iterative solvers (such as the Krylov subspace based GMRES and BiCGSTAB) for linear systems arising from the integral equation methods for the Helmholtz equations. The fast solver is based on two important techniques which reduce the cost of quadrature summations in the Sommerfeld contour integral for Green’s functions in 3-D layered media. The first technique is the removal of surface pole effects along the real axis integration contour by identifying the pole locations with a discrete wavelet transform; in the second technique, we apply a window-based high frequency filter to shorten the contour length. As a result, the integral operator for the 3-D layered media can be efficiently written as a sum of 2-D Hankel cylindrical integral operators, and the latter can be calculated by either a tree-code or a 2-D wideband fast multipole method in a fast manner. Numerical results show the efficiency and parallelism of the proposed fast algorithm.

1. Introduction

The electromagnetic simulation of scattering from objects embedded in a homogeneous or layered structure has many practical applications ranging from geoscience to quantum mechanics. Among the many numerical techniques, integral equation methods [19,25] have several advantages over the conventional finite difference method [32] or finite element method [30], such as the built-in radiation conditions at infinity, easy mesh generations for complex geometries, and a reduced number of unknowns. However, the main issue with integral equation methods in layered media, in contrast to the case of homogeneous media where fast solvers such as the fast multipole method (FMM) are available, is the lack of a fast solver for the dense matrix system arising from the discretized integral equation.

In general, to solve the matrix equation from the integral equation, a Krylov subspace based iteration method, such as GMRES [29] or BiCGSTAB [31], is used. A main step for the iterative solver involves the product of the matrix and a vector. A direct multiplication will cost $O(N^2)$ operations for a full $N \times N$ matrix. Many algorithms exist for a homogeneous background medium to reduce this cost to the order of $O(N)$ or $O(N \log N)$; among them are the FMM [27,28], the adaptive integral method (AIM) [3], and the FFT method [4]. However, the speed-up of integral equation methods for layered media still faces...
much challenges. There have been several attempts to construct fast integral methods for layered media [20]. The natural way is to formulate the scattering problems using Green’s functions of the layered media. As the layered media Green’s function is expressed as a Hankel transform, defined in terms of the Sommerfeld integral, its calculation is time consuming and also dictates how the Helmholtz integral operator can be implemented. Many methods have been proposed to address the difficulties of the Sommerfeld integral, for example, the complex image method (CIM) via Prony’s approximation [2,12], the fast Hankel transform [22], and the steepest descent path for the Sommerfeld integration [13,14,24]. In this paper, techniques such as high frequency filtering and an adaptive quadrature formula (with surface pole identification by a discrete wavelet transform) will be used to minimize the cost of computing the Helmholtz integral operator. The key component of the fast algorithm relies on decomposing the integral operator for the 3-D layered media as a series of operators defined by cylindrical waves (Hankel kernels) within a range of wave numbers resulting from the quadrature formula for the Sommerfeld integral. The cylindrical wave integral operator will be then implemented by a local expansion tree-code for the Bessel function and a wideband FMM (wFMM) [9–11,15], and the latter can be downloaded from http://fastmultipole.org/ [16].

This paper will be organized as follows: in Section 2, the spectral form of the Green’s function will be derived and validated with known analytical solutions. Section 3 starts with the Green’s function in physical space using the Hankel transform, and then techniques mentioned above are presented to minimize the quadrature cost in the contour integral for the Hankel transform. In Section 4, we will present the fast algorithm for implementing the Helmholtz integral operator in 3-D layered media with the local expansion tree-code and wFMM, and numerical performance of the proposed fast algorithms for thin-layer structures will also be presented. Conclusions are given in Section 5.

2. Spectral form for Green’s functions in layered media

The 3-D scalar Helmholtz equation with a point source at \( r' = (0,0,z') \) in an \( N \)-layer structure (Fig. 1) with layer locations \( d_i, \ 0 \leq i \leq N - 1 \), is defined by the following equation,

\[
\nabla \frac{1}{m_i} \nabla G_i(r, r') + k_i^2 G_i(r, r') = -\delta(r, r'), \quad i = 0, 1, \ldots, N, \tag{1}
\]

where \( i \) indicates the \( i \)th layer. By taking the Fourier transform in the horizontal plane to transform \( x \) and \( y \) to spectral variables \( k_x \) and \( k_y \), respectively, we obtain

\[
\frac{d}{dz} \frac{1}{m_i} \frac{d}{dz} \hat{G}_i(k_x, k_y, z; z') - (k^2 - k_i^2) \hat{G}_i(k_x, k_y, z; z') = -\frac{1}{2\pi} \delta(z - z'), \tag{2}
\]

where

\[
k_p^2 = k_x^2 + k_y^2. \tag{3}
\]

Due to cylindrical symmetry in the \((k_x, k_y)\) variables, Eq. (2) simplifies to

\[
\frac{d}{dz} \frac{1}{m_i} \frac{d}{dz} \hat{G}_i(k_p, z; z') - u_i^2 \hat{G}_i(k_p, z; z') = -\frac{1}{2\pi} \delta(z - z'), \tag{4}
\]

Due to cylindrical symmetry in the \((k_x, k_y)\) variables, Eq. (2) simplifies to

\[
\frac{d}{dz} \frac{1}{m_i} \frac{d}{dz} \hat{G}_i(k_p, z; z') - u_i^2 \hat{G}_i(k_p, z; z') = -\frac{1}{2\pi} \delta(z - z'), \tag{4}
\]

Fig. 1. Layered structure with a local coordinate system, \( z_i = z - d_i \).
where

\[ u_i^2 = k_i^2 - k_0^2. \]  

Now the system of ordinary differential equation (4) can be solved analytically for each interval in \( z \) while imposing the matching conditions between the \( i \)th and \( (i-1) \)th layer.

\[ \alpha_{i-1} \hat{G}_{i-1} = \alpha_i \hat{G}_i, \]

\[ \beta_{i-1} \frac{d}{dz} \hat{G}_{i-1} = \beta_i \frac{d}{dz} \hat{G}_i, \]

where

\[ \alpha_i = 1 \quad \text{and} \quad \beta_i = \frac{1}{m_i}. \]

Additionally, some decay conditions in the top- and bottom-most layers will be needed for \( z \to \pm \infty \).

### 2.1. Layer without a source

We first consider the \( i \)th layer without a source, where the right hand side of Eq. (4) become zero, the solution can be written as a linear combination of \cosh\ and \sinh\ with unknown coefficients \( a_i \) and \( b_i \), namely,

\[ \hat{G}_i = a_i \cosh(\sqrt{m_i}u_i z_i) + b_i \sinh(\sqrt{m_i}u_i z_i), \]

where the local coordinate \( z_i \) is defined by

\[ z_i = z - d_i. \]

Applying the interface condition Eqs. (6) and (7) at \( z_i = D_i \) or \( z_{i-1} = 0 \) (here \( D_i \) denotes the width of the \( i \)th layer), we obtain the relationship between the unknown coefficients in the \( i \)th and \( (i-1) \)th layer as

\[ a_{i-1} \cosh(\sqrt{m_{i-1}} u_{i-1} 0) + b_{i-1} \sinh(\sqrt{m_{i-1}} u_{i-1} 0) = a_i \cosh(\sqrt{m_i} u_i D_i) + b_i \sinh(\sqrt{m_i} u_i D_i), \]

\[ \frac{1}{m_{i-1}} u_{i-1} a_{i-1} \sinh(\sqrt{m_{i-1}} u_{i-1} 0) + \frac{1}{m_{i-1}} u_{i-1} b_{i-1} \cosh(\sqrt{m_{i-1}} u_{i-1} 0) = \frac{1}{m_i} u_i a_i \sinh(\sqrt{m_i} u_i D_i) + \frac{1}{m_i} u_i b_i \cosh(\sqrt{m_i} u_i D_i), \]

namely,

\[ a_{i-1} = a_i \cosh(\sqrt{m_i} u_i D_i) + b_i \sinh(\sqrt{m_i} u_i D_i), \]

\[ b_{i-1} = a_i \frac{m_{i-1} u_i}{m_i u_i} \sinh(\sqrt{m_i} u_i D_i) + b_i \frac{m_{i-1} u_i}{m_i u_i} \cosh(\sqrt{m_i} u_i D_i). \]

As a result, the coefficient \((a_{i-1}, b_{i-1})^T\) is related to \((a_i, b_i)^T\) through a transfer matrix \(T_{i-1,i}\), as follows:

\[ \begin{pmatrix} a_{i-1} \\ b_{i-1} \end{pmatrix} = T_{i-1,i} \begin{pmatrix} a_i \\ b_i \end{pmatrix}, \]

where

\[ T_{i-1,i} = \begin{pmatrix} \cosh(\sqrt{m_i} u_i D_i) & \sinh(\sqrt{m_i} u_i D_i) \\ \frac{m_{i-1} u_i}{m_i u_i} \sinh(\sqrt{m_i} u_i D_i) & \frac{m_{i-1} u_i}{m_i u_i} \cosh(\sqrt{m_i} u_i D_i) \end{pmatrix}. \]
2.2. Layer with a source

In the case that a source is located at \( z_m = D \) in the \( m \)th layer, we divide the \( m \)th layer into an upper (U) and lower (L) layer based on the source location as in Fig. 2. In the upper and lower layer, the solutions can be written as

\[
\begin{align*}
\hat{G}^U_m &= a_m^U \cosh(\sqrt{m_m C_m} u_m(z_m - D)) + b_m^U \sinh(\sqrt{m_m C_m} u_m(z_m - D)), \quad D < z_m < D_m, \\
\hat{G}^L_m &= a_m^L \cosh(\sqrt{m_m C_m} u_m z_m) + b_m^L \sinh(\sqrt{m_m C_m} u_m z_m), \quad 0 < z_m < D,
\end{align*}
\]

respectively. At the interface \( z_m = D \), we have the following conditions,

\[
\begin{align*}
\alpha_m \hat{G}^U_m - \alpha_m \hat{G}^L_m &= 0, \\
\beta_m \frac{d}{dz} \hat{G}^U_m - \beta_m \frac{d}{dz} \hat{G}^L_m &= -\frac{1}{2\pi},
\end{align*}
\]

where \( \alpha_m = 1 \) and \( \beta_m = \frac{1}{m_m} \). The second condition Eq. (20) can be inferred from the differential equation (4), which requires that the first derivative of \( \beta_m G \) has a step jump of \(-1/2\pi \) to produce the Dirac delta function on the right hand side of Eq. (4). These interface conditions imply

\[
\begin{align*}
a_m^U \cosh(\sqrt{m_m C_m} u_m 0) + b_m^U \sinh(\sqrt{m_m C_m} u_m 0) &= a_m^L \cosh(\sqrt{m_m C_m} u_m D) + b_m^L \sinh(\sqrt{m_m C_m} u_m D), \\
u_m a_m^U \sinh(\sqrt{m_m C_m} u_m 0) + u_m b_m^U \cosh(\sqrt{m_m C_m} u_m 0) &= u_m a_m^L \sinh(\sqrt{m_m C_m} u_m D) + u_m b_m^L \cosh(\sqrt{m_m C_m} u_m D) - \frac{m_m}{2\pi},
\end{align*}
\]

resulting in the equations

\[
\begin{align*}
a_m^U &= a_m^L \cosh(\sqrt{m_m C_m} u_m D) + b_m^L \sinh(\sqrt{m_m C_m} u_m D), \\
b_m^U &= a_m^L \sinh(\sqrt{m_m C_m} u_m D) + b_m^L \cosh(\sqrt{m_m C_m} u_m D) - \frac{m_m}{2\pi U_m}.
\end{align*}
\]

Therefore, one can have the coefficients in the upper layer expressed in matrix form in terms of the coefficients in the lower layer by

\[
\begin{pmatrix}
a_m^U \\
b_m^U
\end{pmatrix}
= C_m
\begin{pmatrix}
a_m^L \\
b_m^L
\end{pmatrix}
+ S_m,
\]

where

\[
C_m =
\begin{pmatrix}
\cosh(\sqrt{m_m C_m} u_m D) & \sinh(\sqrt{m_m C_m} u_m D) \\
\sinh(\sqrt{m_m C_m} u_m D) & \cosh(\sqrt{m_m C_m} u_m D)
\end{pmatrix}
\]

and

\[
S_m =
\begin{pmatrix}
0 \\
\frac{m_m}{2\pi U_m}
\end{pmatrix}
= \begin{pmatrix}
0 \\
0
\end{pmatrix}.
\]

2.3. Multilayers with a source

In general, the coefficient in the \( j \)th layer \( V_j = (a_j, b_j)^T \) can be found recursively using Eqs. (15) and (25) as follows:

\[
V_j = T_{j, j+1} T_{j+1, j+2} T_{j+2, j+3} \cdots T_{N-1, N} V_N = \prod_{k=j+1}^{N} T_{k-1,k} V_N, \quad j > m, \tag{28}
\]

\[
V_j = T_{j, j+1} T_{j+1, j+2} \cdots T_{m-1,m} \left(C_m \left(T_{m,m+1} T_{m+1,m+2} \cdots T_{N-1,N} V_N\right) + S_m \right) \prod_{k=j+1}^{m} T_{k-1,k} V_N + S_m, \quad j < m. \tag{29}
\]

The top- and bottom-layer coefficients \( (V_0 \text{ and } V_N) \) can be obtained using the solutions \( \hat{G}_0 \) and \( \hat{G}_N \) at both ends of the structure, namely,

\[
\begin{align*}
\hat{G}_0 &= a_0 \cosh(\sqrt{m_0 u_0 z}) + b_0 \sinh(\sqrt{m_0 u_0 z}) = \frac{a_0 + b_0}{2} \exp(\sqrt{m_0 u_0 z}) + \frac{a_0 - b_0}{2} \exp(-\sqrt{m_0 u_0 z}), \tag{30}
\end{align*}
\]

\[
\begin{align*}
\hat{G}_N &= a_N \cosh(\sqrt{m_N u_N z}) + b_N \sinh(\sqrt{m_N u_N z}) = \frac{a_N + b_N}{2} \exp(\sqrt{m_N u_N z}) + \frac{a_N - b_N}{2} \exp(-\sqrt{m_N u_N z}). \tag{31}
\end{align*}
\]

In order to have proper decay of the solution at both ends of the layered structure, the first term in Eq. (30) and the second term in Eq. (31) should be zero. Thus,
\[ a_0 = -b_0, \]  
\[ a_N = b_N. \]  
Finally, the system of algebraic equations between \( V_0 \) and \( V_N \) can be found from Eq. (29) as
\[
\left( \begin{array}{c} a_0 \\ -a_0 \end{array} \right) = \prod_{k=1}^{m} T_{k-1,k}^N \prod_{m=1}^{N} T_{k-1,k} V_N + \prod_{k=1}^{m} T_{k-1,k}^S = \left( \begin{array}{cc} x_{11} & x_{12} \\ x_{21} & x_{22} \end{array} \right) \left( \begin{array}{c} a_N \\ b_N \end{array} \right) + \left( \begin{array}{cc} y_{11} & y_{12} \\ y_{21} & y_{22} \end{array} \right) \left( \begin{array}{c} 0 \\ c \end{array} \right),
\]  
which can be solved for \( a_0 \) and \( a_N \) as follows:
\[
a_N = \frac{-c(y_{12} + y_{22})}{x_{11} + x_{12} + x_{21} + x_{22}},
\]
\[
a_0 = \frac{-c(x_{11} + x_{12})/y_{12} + y_{22})}{x_{11} + x_{12} + x_{21} + x_{22}} + cy_{12}.
\]
Additionally, the initial and terminal conditions for the half-space with perfect electric conductor \( z < 0 \) are derived to provide verification of the formula. In the terminal layer, the solution can be written as
\[
\hat{G}_N = a_N \cosh(\sqrt{m_N u_N}z_N) + b_N \sinh(\sqrt{m_N u_N}z_N),
\]
where \( 0 \leq z_N \leq D_N \), and it must be zero at the lower boundary \( z_N = 0 \). Therefore,
\[
a_N \cosh(\sqrt{m_N u_N}0) + b_N \sinh(\sqrt{m_N u_N}0) = a_N = 0
\]
and the conditions for \( a_0 \) and \( b_0 \) are the same as before. Again from Eq. (29), the system of algebraic equations for \( a_0 \) and \( b_N \) is found to be
\[
\left( \begin{array}{c} a_0 \\ -a_0 \end{array} \right) = \prod_{k=1}^{m} T_{k-1,k}^N \prod_{m=1}^{N} T_{k-1,k} V_N + \prod_{k=1}^{m} T_{k-1,k}^S = \left( \begin{array}{cc} x_{11} & x_{12} \\ x_{21} & x_{22} \end{array} \right) \left( \begin{array}{c} a_N \\ 0 \end{array} \right) + \left( \begin{array}{cc} y_{11} & y_{12} \\ y_{21} & y_{22} \end{array} \right) \left( \begin{array}{c} 0 \\ c \end{array} \right)
\]
and it is solved for \( a_0 \) and \( b_N \) in the half-space as
\[
a_0 = \frac{-c(x_{12}(y_{12} + y_{22})}{x_{12} + x_{22}} + cy_{12},
\]
\[
b_N = \frac{-c(y_{12} + y_{22})}{x_{12} + x_{22}}.
\]
Finally, once the correct unknown coefficient \( V_j \) is found from Eq. (28) and the terminal condition \( V_N \), the spectral domain Green’s function \( \hat{G}_j(k_j, z; z') \) can be written as
\[
\hat{G}_j = V_j \cdot (\cosh(\sqrt{m_j u_j}(z - d_j)), \sinh(\sqrt{m_j u_j}(z - d_j)))
\]
if \( j \neq m \) or \( j = m \) and \( z < z' \) and
\[
\hat{G}_j = V_j \cdot (\cosh(\sqrt{m_j u_j}(z - z'))), \sinh(\sqrt{m_j u_j}(z - z'))
\]
if \( j = m \) and \( z > z' \).

2.4. Validation of the spectral Green’s function

In order to validate the spectral form of Green’s function for a layered structure, we conduct a consistence check with the free-space case, the half-space case, and the multilayered case with symmetry. The free-space spectral Green’s function is given by
\[
\hat{G} = \frac{1}{4\pi u_0} \exp(-\sqrt{m_0 u_0}|z - z'|),
\]
where \( m_0 = 1 \) and
\[
u_0^2 = k_0^2 - k_0^2.
\]
Meanwhile the half-space Green’s function with zero boundary conditions on the half plane \( z = 0 \) is given by an image formula [21] as
\[
\hat{G}_{\text{half}} = \frac{1}{4\pi u_0} \exp(-\sqrt{m_0 u_0}|z - z'|) - \frac{1}{4\pi u_0} \exp(-\sqrt{m_0 u_0}|z + z'|).
\]
We apply formulas (42) and (43) with 5 fictitious layers (with layer locations \( d_0 = 0, \ d_1 = -2, \ d_2 = -4, \ d_3 = -6, \ d_4 = -8, \) and \( k_i = m_i = 1 \) for all layers) and \( z = -3.0 \) and \( z' = 5.0, \) and the result is compared with the exact free-space Green’s function calculated using Eq. (44) along the positive real \( k_p \) axis in Fig. 3(a). In Fig. 3(b), for the fixed points \( z = 5.0 \) and \( z' = 3.0, \)
the half-space spectral Green’s function consisting of 5 fictitious layers (with layer locations \( d_0 = 8 \), \( d_1 = 6 \), \( d_2 = 4 \), \( d_3 = 2 \), \( d_4 = 0 \), and \( k_i = m_i = 1 \) for all layers) and with perfect electric conductor (\( z < 0 \)) is calculated with the terminal condition given by Eq. (41). In both free- and half-spaces, numerical solutions agree well with the exact solutions. For a general multilayer medium, we can verify the formulas (42) and (43) by checking the symmetries of the Green’s function for a symmetric profile of layer parameters. Fig. 4 (a) and (b) show the spectral Green’s functions for \( k_0 = 1 \), \( k_1 = 1 \), \( k_2 = 1 \), \( k_3 = 3 \), \( k_4 = 3 \), \( k_5 = 3 \) and \( k_0 = 1 \), \( k_1 = 3 \), \( k_2 = 3 \), \( k_3 = 3 \), \( k_4 = 3 \), \( k_5 = 1 \) for \( z = −3 \) and \( z' = −5 \) pair (marker) and \( z = −5 \) and \( z' = −3 \) pair (solid line).

3. Green’s function in real space

The Green’s function in real space can be obtained from the Green’s function in the spectral domain with the 0th order Hankel transform, namely,
where the integral is truncated at the length $L$ (see Fig. 5(a)), and $k_s^q$ and $w_s$ are quadrature points and weights, respectively. This numerical integration Eq.(47) has two difficulties. The first difficulty comes from the surface poles in the real axis of the complex $k_p$ plane [25]. In order to remove the pole effects, a discrete wavelet transform (DWT) [5,6] is used to identify the pole locations and then adaptive generalized Gaussian quadratures are used. The second difficulty is the size of the contour length $L$ due to the slow decay of the spectral Green’s function when $z \sim z'$ or $z = z'$. For this matter, the window function is used to enhance the decay of the spectral Green’s function or to shorten the truncation of the integral domain $L$. In the following subsections, the adaptive generalized Gaussian quadratures and window function techniques are presented.

3.1. Adaptive quadrature points with wavelets

The spectral Green’s function has an oscillatory and $x^{-1/2}$ singular behavior (near the surface poles) along the real axis of the complex $k_p$ space, therefore an adaptive and specially designed quadrature formula will be used to reduce the number of quadrature points $N_q$ in Eq.(47); this plays an important role in the overall cost of the fast algorithm. At the same time, an accurate quadrature rule for $x^{-1/2}$ is necessary. First, surface pole locations are found using DWT (see Refs. [5,6] for details) as can be seen in Fig. 6, which shows the real part of the integrand of the Hankel transform (right axis) and wavelet collocation points (left axis) calculated with layer locations $d_0 = 0$, $d_1 = -2$, $d_2 = -4$, $d_3 = -6$, $d_4 = -8$ and $k_0 = 5$, $k_1 = 2$, $k_2 = 4$, $k_3 = 3$, $k_4 = 4$, $k_5 = 5$, and $z = -3$ and $z' = -5$ as a function of $k_p$ from 0 to 5. The locations of singular behaviors are well detected with wavelet collocation points, and these can be used to construct an adaptive quadrature rule for the Hankel transform with generalized Gaussian quadrature for $x^{-1/2}$ by Ma et al. [23]. In detail, denote the singular

![Fig. 5. (a) Integral contour for the Hankel transform for the tree-code version solver and (b) the wFMM version solver (mirror image of the contour in (a)).](image)

![Fig. 6. Singular behavior detection with wavelets. The red solid line (right axis) represents the integrand of the Hankel transform and blue asterisks (left axis) are wavelet collocation points at each level with layer locations $d_0 = 0$, $d_1 = -2$, $d_2 = -4$, $d_3 = -6$, $d_4 = -8$ and $k_0 = 5$, $k_1 = 2$, $k_2 = 4$, $k_3 = 3$, $k_4 = 4$, $k_5 = 5$, and $z = -3$ and $z' = -5$.](image)
points by $s_i$, then the generalized Gaussian quadratures (3–5 quadrature points are used) can be used in the intervals $[s_i - \delta, s_i]$ and $[s_i, s_i + \delta]$ for a given positive $\delta$, and normal Gaussian quadrature is used in all other intervals (see Fig. 7).

Fig. 8 is drawn to show how the performance of the adaptive quadrature combined with the generalized Gaussian quadratures compares with the uniform quadrature, which consists of equally spaced intervals with several Gaussian quadrature points in them. First, in Fig. 8(a), the real part of the free-space Green’s functions are calculated with 200 uniform (triangle) and 20 (circle), 35 (square), and 65 (cross) adaptive quadrature points. The solution calculated with the uniform quadratures visibly differs from the exact solution (solid line) calculated with Eq. (52). Fig. 8(b) displays the relative errors between the exact and numerical solution calculated with 200 uniform (triangle) and 20 (circle), 35 (square), and 65 (cross) adaptive quadrature points. Only one tenth of the quadrature points are necessary to outperform the uniform quadratures. Fig. 8(c) shows the multilayer Green’s functions calculated with 109 (circle) and 205 (square) adaptive and 1000 uniform (triangle) quadrature points compared with the reference solution (solid line), which is calculated with 1309 adaptive quadrature points (see the figure caption for the computation parameters). Again, 109 adaptive quadrature points performs better than
1,000 uniform quadrature points. In order to see the difference more quantitatively, relative errors between the reference solution and the numerical solution calculated with adaptive and uniform quadrature points are displayed in Fig. 8(d).

In conclusion, the adaptive quadrature combined with generalized Gaussian quadrature can achieve almost 10 times better accuracy with 10% quadrature points compared with the uniform one. This reduction of quadrature points will play a key role in optimizing the fast solver in a later section since it determines the number of tree-code or wFMM calls.

3.2. Hankel transform with window functions

When \( z \) is very close or equal to \( z' \), the spectral Green’s function decays slowly (for example, the free- and half-space spectral Green’s functions in Eqs. (44) and (46) decay as \( (k^2 - k_0^2)^{-1/2} \) along the real \( k_ρ \) axis). As a consequence, the truncation of the integral domain \( L \) has to be large, and the number of quadrature points will be increased as a result. In order to overcome this problem, a window function method is proposed in Ref. [8]. According to the Algorithm 1 in this reference, the Hankel transform (Eq. (47)) is modified and approximated for \( k_ρ > a \) by

\[
G(\rho; z, z') = \frac{1}{M_0} \int_{a}^{\infty} \hat{G}(k_ρ; z, z') \hat{\psi}(k_ρ) k_ρ d k_ρ + O(a^2),
\]

where the window function \( \psi(k_ρ) \) in real space and \( \hat{\psi}(k_ρ) \) in the spectral domain are defined by

Fig. 9. Real part of the spectral Green’s function calculated without (blue dashed line) and with (red solid line) window functions (window support \( a = 0.6 \) and order \( m = 6 \)) for (a) free-space with 5 fictitious layers with \( k_l = m_l = 1 \) and (b) multilayered structure with \( k_0 = 1, k_1 = 3, k_2 = 3, k_3 = 3, k_4 = 3, \) and \( k_5 = 1 \) when \( z = z' = -4 \).

Fig. 10. Half-space Green’s function calculated without (black solid line) and with (dash-dot red line) window functions and exact solution (dashed blue line) when \( z = z' = 4 \).
\[
\psi_a(\rho) = \begin{cases} 
(1 - \left(\frac{q}{a}\right)^2)^m, & \text{if } \rho \leq a \\
0, & \text{otherwise}
\end{cases},
\]
(49)

\[
\hat{\psi}_a(k, \rho) = \int_0^\infty \psi_a(\rho) J_0(k \rho) k \rho dk.
\]
(50)

and

\[
M_0 = \frac{1}{2\pi} \int_{\rho < a} \psi_a(\rho) d\rho = \frac{a^2}{2(m+1)}.
\]
(51)

The closed forms and their approximations of window functions in the spectral domain (Eq. (50)) are given in Ref. [33], and included here for completeness in Appendix A.

The free-space and multilayer spectral Green’s function calculated with (solid line) and without (dashed line) window functions when \(z = z’\) are plotted in Fig. 9(a) and (b), respectively. Both figures show the slow decay of the spectral Green’s function and the enhancement of decay from the window functions. Fig. 10 depicts the half-space Green’s function when \(z = z’ = 4\) calculated with (dash-dotted line) and without (solid line) window functions compared with the exact solution (dashed line). \(L = 20\) is used for all of the following calculations.

Now, Green’s functions in layered structures will be calculated with the techniques presented in previous subsections. For fixed points \(z = -3\) and \(z’ = -5\), the numerical solution from Eq. (47) with 5 fictitious layers (with layer locations \(d_0 = 0, d_1 = -2, d_2 = -4, d_3 = -6, d_4 = -8\) and \(k_i = m_i = 1, i = 0, 1, \ldots, 5\)) and the exact free-space Green’s function,

\[
G(\rho; z, z’) = \frac{e^{-jk\sqrt{|\rho^2 + (z-z’)^2|}}}{4\pi \sqrt{\rho^2 + (z-z’)^2}},
\]
(52)

are plotted as a function of \(\rho\) in Fig. 11(a). In addition, for the fixed points \(z = 5.0\) and \(z’ = 3.0\), the half-space Green’s function with 5 fictitious layers (\(d_0 = 8, d_1 = 6, d_2 = 4, d_3 = 2, d_4 = 0\) and \(k_i = m_i = 1, i = 0, 1, \ldots, 4\)) with perfect conductor \((z < 0)\) is calculated in Fig. 11(b). The numerical solution agrees very well with the exact half-space Green’s function obtained with the image method, namely,

\[
G_{\text{half}} = \frac{e^{-jk\sqrt{|\rho^2 + (z-z’)^2|}}}{4\pi \sqrt{\rho^2 + (z-z’)^2}} = \frac{e^{-jk\sqrt{|\rho^2 + (z-z’)^2|}}}{4\pi \sqrt{\rho^2 + (z-z’)^2}}.
\]
(53)

Finally, in Fig. 11(c) and (d), two different multilayered structures are considered with \(k_0 = 1, k_1 = 3, k_2 = 1, k_3 = 3, k_4 = 3, k_5 = 3, k_6 = 1, k_7 = 3, k_8 = 3, k_9 = 3\), and \(k_0 = 1, k_1 = 3, k_2 = 3, k_3 = 3, k_4 = 3, k_5 = 1, k_6 = 3\), respectively, and with \(m_i\) assumed to be 1 in all layers for both figures. The symmetry is tested by switching \(z\) and \(z’\) in both figures. In both cases, the two solutions obtained with \(z = -3\) and \(z’ = -5\) and \(z = -5\) and \(z’ = -3\) are found to be the same because of the symmetry properties of the structure.

4. Fast algorithms for computing the Helmholtz integral operator in layered media

With the layered media Green’s function developed in the previous section, parallel fast algorithms for the Helmholtz integral operator in layered media are proposed in this section.

The computation of the integral in Eq. (47) is accelerated with the 2-D wFMM for the complex Helmholtz equation [10,11] as suggested in [7]. In order to use wFMM, the integral domain in Eq. (47) is extended with its mirror image using symmetry as in Fig. 5(b), and the Bessel function is replaced by the Hankel function as follows:

\[
G(\rho; z, z’) = \frac{1}{2M_0} \int_{-\infty}^{\infty} \tilde{G}(k, z, z’) H_0^{(1)}(k \rho) \hat{\psi}_a(k) dk.
\]
(54)

Next, the integral in Eq. (54) is discretized with adaptive quadrature points \((k_p^i)\) and weights \((w_i)\) proposed in Section 3.1 as

\[
G(\rho; z, z’) = \frac{1}{2M_0} \sum_{i=1}^{2N_0} w_i \tilde{G}(k_p^i, z, z’) H_0^{(1)}(k_p^i \rho) \hat{\psi}_a(k_p^i) k_p^i = \frac{1}{2M_0} \sum_{i=1}^{2N_0} \Gamma_i \tilde{G}(z, z’, k_p^i) H_0^{(1)}(k_p^i \rho),
\]
(55)

where

\[
\Gamma_i = w_i \hat{\psi}_a(k_p^i) k_p^i
\]
(56)

and \(N_0\) is the number of quadrature points in the positive real axis. Assume that there are \(N_2 = O(1)\) layers and \(|\Delta| = O(N_x N_y)\) points in each layer. Then, for each point \((x_k^i, y_k^i, z_k)\) \in \Delta, 1 \leq k \leq N_x, the Helmholtz integral operator in layered media \(\phi(x_k^i, y_k^i, z_k)\) can be calculated as follows using Eq. (55):
\[
\phi(x_i^k, y_j^k, z_k) = \sum_{m,n,l} G(x_i^k, y_j^k, z_k; x_m^l, y_n^l, z_l) f(x_m^l, y_n^l, z_l)
\]
\[
= \frac{1}{2M_0} \sum_{m,n,l=1}^{2N_0} J_m \tilde{G}(z_k, z_l) H_0^{(1)}(k_p \rho) f(x_m^l, y_n^l, z_l)
\]
\[
= \frac{1}{2M_0} \sum_{m,n,l=1}^{2N_0} J_m \tilde{G}(z_k, z_l) H_0^{(1)}(k_p \rho) f(x_m^l, y_n^l, z_l)
\]
\[
= \frac{1}{2M_0} \sum_{m,n,l=1}^{2N_0} J_m \tilde{G}(z_k, z_l) H_0^{(1)}(k_p \rho) f(x_m^l, y_n^l, z_l)
\]
\[
\text{where}
\]
\[
\rho_{ij,m,n} = \sqrt{(x_i^k - x_m^l)^2 + (y_j^k - y_n^l)^2}
\]
\[
\psi_i(x_i^k, y_j^k, z_i) = \sum_{m,n} H_0^{(1)}(k_p \rho_{ij,m,n}) f(x_m^l, y_n^l, z_l)
\]

Now, for each fixed pair \(z_k\) and \(z_l\), \(\psi_i\) is the cylindrical wave integral operator, and it can be evaluated for all positive \(k^l_p\) with wFMM in a fast manner, and can be reused for negative \(k^l_p\) (i.e. the mirror-imaged integral contour) due to the symmetry of the Hankel function.

![Fig. 11](image-url)

**Fig. 11.** The Green’s function for (a) free-space \((k_i = m_i = 1\) with \(z = -3\) and \(z' = -5\)) calculated with numerical method (marker) and the exact solution (solid line), (b) half-space \((k_i = m_i = 1\) with \(z = 5\) and \(z' = 3\)) calculated with the numerical method (marker) and the exact solution (solid line), (c) the multilayered structure with \(k_0 = 1, k_1 = 1, k_2 = 1, k_3 = 3, k_4 = 3,\) and \(k_5 = 3\) with the pair \(z = -3\) and \(z' = -5\) (marker) and the pair \(z = -5\) and \(z' = -3\) (solid line), and (d) the multilayered structure with \(k_0 = 1, k_1 = 3, k_2 = 3, k_3 = 3, k_4 = 3,\) and \(k_5 = 1\) with the pair \(z = -3\) and \(z' = -5\) (marker) and the pair \(z = -5\) and \(z' = -3\) (solid line).
In the derivation of the fast algorithm, the integral domain is extended to the negative axis in order to utilize wFMM. However, a 2-D local expansion tree-code for the 0th order Bessel function can be easily developed and implemented based on our wFMM code (see Appendix B) with the same computational complexity and with a smaller prefactor. Moreover, two big advantages of the tree-code over the wFMM are that it does not require time-consuming multipole-to-local conversion operations and wideband treatment in wFMM. As a consequence, the tree-code is expected to be more efficient and faster than wFMM. Now we derive the tree-code version fast solver directly from Eq. (47) for the positive integral contour only. First, the Helmholtz integral operator in layered media (Eq. (58)) can be rewritten as

\[
\phi(x^0_i, y^0_j, z_0) = \frac{1}{M_0} \sum_{s=1}^{N_s} \sum_{l} \hat{G}(z_k, z_l, k_s) \psi_s(x^0_i, y^0_j, z_0),
\]

where

\[
\psi_s(x^0_i, y^0_j, z_0) = \sum_{m,n} J_0(k_s \rho_{i,j,m,n}) y(x_m^0, y_n^0, z_0).
\]

The \(\psi_s\) can be calculated with the 2-D tree-code for the 0th order Bessel function only for positive quadrature points.

The operation counts for both solvers are \(O(N_s N_x^2 N_y N_z \log N_x N_y N_z)\), where \(N_x\), \(N_y\), and \(N_z\) are the number of unknown points in the \(x\)-, \(y\)-, and \(z\)-axis, respectively. \(N_z\) is much smaller than \(N_x\) and \(N_y\) for thin layers which is the case for many applications. Therefore, the choice of \(N_z\) plays an important role in the overall cost of the fast algorithm.

Fig. 13 shows the CPU time of the tree-code version (Eq. (61)) and wFMM version (Eq. (58)) fast algorithms compared with direct calculation (Eq. (57)) for the half-space. Fig. 13(a) is redrawn as (b) (with \(x\)-axis truncated at 100,000) to show the break-even point. Five fictitious layers \(d_0 = 8, d_1 = 6, d_2 = 4, d_3 = 2, d_4 = 0\) with \(k_i = m_i = 1, i = 0, 1, \ldots, 5\), and the
perfect conductor ($z < 0$) are used to calculate the Green’s function at two layers ($N_z = 2$) of random unknown points $N_x \times N_y$ in boxes $[2.5,3.5] \times [-0.5,0.5]$ located at $z = 5$ and $[-0.5,0.5] \times [-0.5,0.5]$ located at $z = 3$ (see Fig. 12 for an illustration).

The number of unknown points $N_x \times N_y \times N_z$ are varied from $50 \times 50 \times 2$ to $800 \times 800 \times 2$, and the outer sum on $k_q^i$ (15 quadrature points for 1% relative error) is parallelized with OpenMP for multicore shared memory systems [26]. Computations are conducted with a machine consisting of two quad-core Intel Xeon 3.00 GHz processors, 32 GB memory, and a gcc version 4.5.1 running on Fedora release 11. The CPU time increases quadratically for the direct calculation and $O(N_x N_y N_x \log N_x N_z)$ with $N_x = 2$ for the fast solvers as predicted. The break-even point between the direct and wFMM version fast solver is around 60,000 points and the tree-code version is 35,000 (see Fig. 13(b)). The tree-code version solver is found to be faster than the wFMM version fast algorithm as expected. Note that any numerical tools can be used to accelerate the calculation of $\psi_i$ depending on user’s preference.

In addition to the half-space, a multilayer (with layer locations $d_0 = 0$, $d_1 = -2$, $d_3 = -4$, $d_4 = -8$ and $k_0 = 5$, $k_1 = 2$, $k_3 = 4$, $k_4 = 8$ and $m_i = 1$, $i = 0,1,...,5$), with two layers ($N_z = 2$) of random unknown points $N_x \times N_y$ in boxes $[2.5,3.5] \times [-0.5,0.5]$ located at $z = -3$ and $[-0.5,0.5] \times [-0.5,0.5]$ located at $z = -5$, is considered in Fig. 14. The number of unknown points in the thin layer $N = N_x \times N_y \times N_z$ is varied from $50 \times 50 \times 2$ to $800 \times 800 \times 2$. The tree-code version fast solver with $N_q = 40$ (red triangle) quadrature points, which yields 7% relative error, is compared with the direct calculation (blue circle) in terms of the CPU time. The break-even point was found to be at approximately $N = 62,000$, and it will increase as the number of quadrature points increases (or if a higher accuracy is required) because
the tree-code must be called at each additional quadrature point. Fig. 14(a) is redrawn with the log–log scale in Fig. 14(b) to show the computational complexity. The \( O(N) \) complexity line (cross) is added only for a comparison purpose.

Finally, in Fig. 15, the parallel efficiency for the half-space is plotted with \( N = 225,000 \) as the number of CPU cores increases from 1 to 8. When 8 cores are used, it is 7 times faster than the single core CPU time.

5. Conclusions

Parallel fast algorithms for the Helmholtz integral operator in 3-D layered media based on the tree-code for the Bessel function and the wFMM for the Hankel function are implemented, and their efficiencies are investigated. In computing the Hankel transform for the layered media Green’s functions, adaptive quadrature rules with a discrete wavelet transform for pole singularity identifications are used to remove surface pole effects, and the window function is utilized to enhance the decay of the spectral Green’s function for fast convergence. The proposed algorithms can be implemented with a high degree of parallelism with respect to the quadrature point \( N_q \), namely, the computation for each of the quadrature points can be done independently using separate processors. More rigorous optimization and massive parallelization are under study and will be reported elsewhere. Finally, the fast algorithm will be applied to many possible applications in electromagnetics and quantum mechanics such as quantum dots [34] and surface plasmons in thin structured films [17].

Acknowledgments

This work is supported in part by the US army office of research (Grant No. W911NF-11-1-0364), NSF grant DMS-1005441 and the US Department of Energy (Grant No. DEFG0205ER25678). The authors would like to acknowledge Prof. Jingfang Huang at UNC Chapel Hill, Prof. Robert Krasny at U. of Michigan, Dr. Hengliang Zhu at Fudan University, and Mr. Mark Hamrick at UNC Charlotte for valuable comments and help regarding FMM, tree-code, DWT, and computing support, respectively.

Appendix A. Closed form of the window functions

Here the closed forms and their approximations of the window functions are provided up to the 7th order.

- The case \( k_qa < 1 \):

Using the taylor expansion of the Bessel function [1],

\[
J_n(z) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \frac{1}{(n+k)!} \left( \frac{z}{2} \right)^{2k+n},
\]

(A.1)

the window function in the spectral domain can be written as

\[
\hat{\psi}_a(k_q) = \int_0^a \left( 1 - \left( \frac{r}{a} \right)^2 \right)^m J_0(k_qr) \rho d\rho
\]

(A.2)

\[
= a^m \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{r_{ij} \neq 0} a_{j+1} y_{ij} y_{ij} \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{r_{ij} \neq 0} a_{j+1} y_{ij} y_{ij}
\]

(A.3)

where \( y = k_qa \) and the coefficients \( H_{m_j} \). \( j = 0, 1, \ldots, 6 \) are given in Table A.1.

- The case \( k_qa \geq 1 \):

Using the Lommel function and Bessel functions, the window function in the spectral domain can be expressed as follows:

\[
\hat{\psi}_a(k_q) = g_m,
\]

(A.5)
where
\[
g_1 = \frac{a^2}{y^2} \left[ -2J_0(y) + \frac{4}{y} J_1(y) \right], \tag{A.6}
\]
\[
g_2 = \frac{a^2}{y^2} \left[ 8 \left( \frac{4}{y} J_0(y) + \left( \frac{8}{y^2} - 1 \right) J_1(y) \right) \right], \tag{A.7}
\]
\[
g_3 = \frac{a^2}{y^2} \left[ \left( 1 - \frac{24}{y^2} \right) J_0(y) - \frac{8}{y} \left( 1 - \frac{6}{y^2} \right) J_1(y) \right], \tag{A.8}
\]
\[
g_4 = \frac{a^2}{y^2} \left[ 182 \left( 8 \left( 1 - \frac{16}{y^2} \right) J_0(y) + 2 \left( 1 - \frac{24}{y^2} \right) \left( \frac{3}{y^2} - 16 \right) J_1(y) \right) \right], \tag{A.9}
\]
\[
g_5 = \frac{a^2}{y^2} \left[ 1920 \left( 96 \left( \frac{3}{y^2} - 40 \right) J_0(y) + \frac{1}{y} \left( 36 - \frac{1536}{y^2} \right) \left( 1 - \frac{5}{y^2} \right) J_1(y) \right) \right], \tag{A.10}
\]
\[
g_6 = \frac{a^2}{y^2} \left[ 46080 \left( \frac{12}{y^2} - 1 \right) J_0(y) + \left( -1 + \frac{96}{y^2} \left( \frac{3}{y^2} - 20 \right) \left( 5 - \frac{24}{y^2} \right) \right) J_1(y) \right], \tag{A.11}
\]
\[
g_7 = \frac{a^2}{y^2} \left[ 645120 \left( 480 \left( 1 - \frac{4}{y^2} \left( 15 - \frac{168}{y^2} \right) \right) J_0(y) + \frac{32}{y} \left( 30 - \frac{144}{y^2} \left( 6 - \frac{28}{y^2} \right) \right) J_1(y) \right) \right]. \tag{A.12}
\]

Appendix B. Local expansion tree-code for Bessel function in 2-D

The downward pass of the 2-D FMM algorithm [18] is used to implement local expansion tree-code. In this tree-code, far-field signature of the \(J\)-expansion (local expansion) is directly constructed from the interaction list rather than converted from the multipole expansion in FMM. Then, it is shifted to its children boxes with a diagonalized operator and added with the far-field signature constructed from the interaction list of the children box. This procedure continues until the bottom of the tree data structure is reached. At the childless boxes, the far-field signature is evaluated at each particle location and added with direct interaction between particles in the close-neighbor boxes and itself. All the necessary formulas for the implementation of tree-code are given below. All the analysis and formula follow exactly from the [27].

- \(J\)-expansion:
  Assume \(N\) particles \(q_k = (x_k, y_k), k = 1, 2, \ldots, N\), outside of a disc \(D_1\) centered at \(c_1\), then for any point \(r\) inside of \(D_1\), the radiation field due to \(N\) particles can be written as
  \[
  \psi_{c_1}(r) = \sum_{k=1}^{N} J_0(||r - r_k||) q_k = \sum_{m=-\infty}^{\infty} a_m J_m(\rho_1) e^{-im\theta}, \tag{B.1}
  \]
  where
  \[
  a_m = \sum_{k=1}^{N} q_k J_0(k ||r_k||) e^{im\theta_k}, \tag{B.2}
  \]
  \(\rho_1 = ||r - c_1||\), and \(\theta\) and \(\theta_k\) are angles between \(r - c_1\) and the x-axis and \(r_k - c_1\) and the x-axis, respectively. This formula is a direct consequence of the addition theorem for Bessel functions [1].

- \(J\)-2-\(J\) operation:
  Assume
  \[
  \psi_{c_1}(r) = \sum_{m=-\infty}^{\infty} a_m J_m(\rho_1) e^{-im\theta_1}, \tag{B.3}
  \]
  is analytic for any \(r \in D_1\) centered at \(c_1\), where \(\rho_1 = ||r - c_1||\) and \(\theta_1\) is the angle between \(\rho_1\) and the x-axis. Then, in a disc \(D_2\) centered at \(c_2\), the potential can be written as
  \[
  \psi_{c_2}(r) = \sum_{m=-\infty}^{\infty} a_m J_m(\rho_2) e^{-im\theta_2}, \tag{B.4}
  \]
  where
  \[
  a_m = \sum_{J=-\infty}^{\infty} e^{ij \theta_{12} - \pi} a_m J_{J}(k \rho_{12}), \tag{B.5}
  \]
  \(\rho_2 = ||r - c_2||\), \(\rho_{12} = ||r_2 - r_1||\), \(\theta_2\) and \(\theta_{12}\) are the angles between \(\rho_2\) and the x-axis and \(\rho_{12}\) and the x-axis, respectively. Again, this operation is a direct application of the addition theorem.
Far-field signature $G$ of the $J$-expansion:

$$G(\theta) = \sum_{m=-\infty}^{\infty} a_m e^{-i(m\pi/2)} e^{i m \theta},$$

where $\{a_m\}_{m=-\infty}^{\infty}$ are the coefficients of the $J$-expansion.

Diagonalized $G$-2-$G$ operation:

$$h(w) = e^{-iK_{12} \sin(w - h_2 - \pi)},$$

Then, the far field expansion $G_{c1}$ in $D_1$ centered at $c_1$ can be translated to the far field expansion $G_{c2}$ in $D_2$ centered at $c_2$ by

$$G_{c2}(w_j) = h(w_j) G_{c1}(w_j),$$

where

$$w_j = \frac{2\pi j}{N}, \quad j = 0, 1, 2, \ldots, N - 1.$$