Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 211 (2006) 616–637
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An adaptive fast solver for the modified Helmholtz equation in two dimensions

Hongwei Cheng [a], Jingfang Huang [b,*,1], Terry Jo Leiterman [b]

[a] *Madmax Optics Inc., 3035 Whitney Ave., Hamden, CT 06518, USA*
[b] *Department of Mathematics, CB# 3250, University of North Carolina, Chapel Hill, NC 27599, USA*

Received 19 December 2003; received in revised form 24 May 2005; accepted 6 June 2005
Available online 2 August 2005

Abstract

In this paper, we present a fast multipole-accelerated integral equation method for solving the modified Helmholtz equation $\Delta u(\vec{x}) - \beta^2 u(\vec{x}) = f(\vec{x})$ in two dimensions. The method is direct, and unlike classical FFT based fast solvers, it allows for adaptive mesh refinement but with comparable amount of work per grid point. When the computational domain is rectangular, Dirichlet, Neumann, periodic, and free-space boundary conditions can be imposed analytically without the need to solve a system of linear equations. Several important features of the algorithm are discussed, including the use of precomputed tables, diagonal translation operators, and lattice sums to impose periodic boundary conditions. Numerical experiments show that, for a wide range of the parameter β, the algorithm is stable and high-order accurate.

Keywords: Fast multipole method; Generalized Gaussian quadrature; Modified Helmholtz equation

1. Introduction

A variety of problems in scientific computing require the efficient solution of the partial differential equation

$$\Delta u(\vec{x}) - \beta^2 u(\vec{x}) = f(\vec{x}). \tag{1}$$

* Corresponding author. Fax: +1 919 962 9345.
E-mail address: huang@amath.unc.edu (J. Huang).
[1] The work of this author was partially supported by NSF Grant DMS-0411920 and DMS-0327896.

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.06.006

This equation is sometimes referred to as the modified Helmholtz equation or the Yukawa equation. It appears, for example, in implicit marching schemes for the heat equation, in Debye–Huckel theory, and in the linearization of the Poisson–Boltzmann equation [37,39,46]. The underlying free-space Green's function is usually referred to as the Yukawa potential in nuclear physics. In physics, chemistry, and biology, when Coulomb forces are damped by screening effects, this Green's function is also known as the screened Coulomb potential. In Debye–Huckel theory, the constant β represents the inverse of the electron Debye length which indicates the length scale over which an individual charged particle exerts a notable effect.

Traditional fast solvers, when based on the fast Fourier transform (FFT) [6,7], only allow for uniform grids and simple geometry. Iterative methods such as the multigrid method and domain decomposition techniques handle unstructured grids and complex geometry [5,9,10,14,36,40,41]. However, despite significant progress, the available solvers tend to compare unfavorably with the FFT based direct solvers in terms of work per grid point.

In this paper, we introduce a new fast direct solver for the modified Helmholtz equation in two dimensions. The algorithm is based on an integral equation formulation and is accelerated using the new version of fast multipole method [11,26,28]. The solver allows for adaptive mesh refinement and is comparable in speed with FFT based methods. For the sake of simplicity, we restrict our attention here to the unit square, but allow different kinds of boundary conditions. For this simple geometry, using results from classical potential theory, the method is explicit. That is, no linear system is solved for either volume or boundary unknowns. For general complex geometry, an integral equation needs to be solved for unknowns restricted to the boundary alone.

One feature of the present paper is a generalization of the technique first introduced in [11,26,28] to accelerate the fast multipole method (FMM) for the Laplace and Helmholtz equations. It is based on the use of exponential expansions rather than classical multipole expansions to represent far-field interactions. Due perhaps to the complexity of the new version of the FMM, we are not aware of any implementation for computing volume integrals with the Yukawa potential in either two or three dimensions, although particle interactions in three dimensions were discussed in [25]. Our new solver, described below, shows great potential to compete with the FFT in work per grid point, while still allowing for adaptive mesh refinement and accuracy control.

It is worth mentioning that in marching schemes for the time dependent heat and Navier–Stokes equations, the modified Helmholtz equation usually appears in the form $u(\vec{x}) - \epsilon\Delta u(\vec{x}) = \tilde{f}(\vec{x})$, where $\epsilon \ll 1$. Rescaling this equation to the standard form, we have $\Delta u(\vec{x}) - \beta^2 u(\vec{x}) = -\beta^2 \tilde{f}(\vec{x})$ with $\beta \gg 1$.

The paper is organized as follows: In Section 2, we outline the integral equation formulation and the fast multipole method. The current method shares many features with the recently developed fast solvers for the Poisson equation $\Delta u = f$ and the pseudo-differential equation $(-\Delta)^{1/2}\psi = \omega$ (see [20,22]). Therefore, we discuss the generic structure of the FMM algorithm only briefly and refer the reader to the earlier papers for further details. In Section 3, we discuss several important features of the current algorithm, including the use of precomputed tables, and the use of lattice sums and the method of images to impose different boundary conditions. In Section 4, we provide a brief error analysis, which supports our claims for accuracy, and finally in Section 5, we present several numerical examples for different values of β and compare the efficiency and accuracy of our solver to the FFT based uniform grid solvers.

2. Integral equation methods and the fast multipole method

In recent years, there has been a great deal of success in implementing highly accurate and efficient adaptive solvers for a wide range of linear partial differential equations based on the integral form of the solution. The integral approach is naturally adaptive, allows for higher order approximations easily, and can handle arbitrarily complex boundaries. In the absence of fast algorithms, these approaches are not

competitive. However, with modern techniques such as the particle mesh Ewald method (PME), the precorrected FFT method (pFFT), and the fast multipole method (FMM) [16,23,27,44], the integral equation approach is extremely efficient for large scale simulations and may become the method of choice in many science and engineering applications.

2.1. Potential theory and the integral equation method

In this paper, we consider an integral equation approach for the modified Helmholtz equation in the unit square D centered at the origin. For the sake of concreteness, we will consider, in some detail, the Dirichlet boundary conditions on D given in Fig. 1.

Define $\vec{x} = (x_1, x_2)$ and $\vec{y} = (y_1, y_2)$. The free-space Green's function $G(\vec{x} - \vec{y}) = G(x_1 - y_1, x_2 - y_2)$ for the operator $\Delta - \beta^2$ is given by the zeroth-order modified Bessel's function of second kind,

$$G(x_1, x_2) = -\frac{1}{2\pi} K_0\left(\beta\sqrt{x_1^2 + x_2^2}\right). \tag{2}$$

Therefore, a particular solution to the modified Helmholtz equation in (1) is simply

$$u(\vec{x}) = \int\int_D G(\vec{x} - \vec{y}) f(\vec{y})\, \mathrm{d}\vec{y}.$$

This function, however, does not satisfy the desired Dirichlet boundary conditions. For this, a more elaborate solution procedure is required, based on the method of images [35,45].

Consider first the problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} \Delta u_1(\vec{x}) - \beta^2 u_1(\vec{x}) = f(\vec{x}) & \text{in } D = \left[-\frac{1}{2}, \frac{1}{2}\right] \times \left[-\frac{1}{2}, \frac{1}{2}\right], \\ u_1 = 0 & \text{on the boundary.} \end{cases} \tag{3}$$

The classical method of images can be applied in a straightforward manner and tiles the plane with the pattern of images depicted in Fig. 2. Here, the shaded box is the computational domain D containing the source distribution f. The odd reflection of f across the top boundary is denoted by $-f_{\mathrm{T}}$ while the odd reflection of the function f across the right boundary is denoted by $-f_{\mathrm{R}}$. The reflection of $-f_{\mathrm{R}}$ across the line $x_2 = +\frac{1}{2}$ is denoted by f_{RT}. It is straightforward to verify that the 2×2 super-cell defined by the solid lines in Fig. 2 tiles the plane periodically and that the convolution of the free-space Green's function with the entire plane of sources in Fig. 2 solves (3).

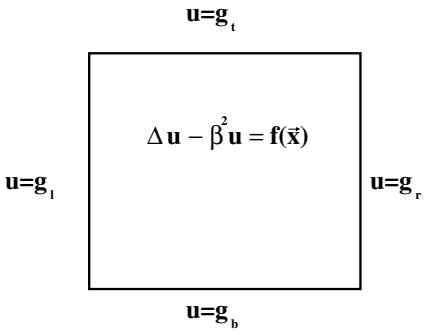

Fig. 1. Modified Helmholtz equation with Dirichlet boundary conditions.

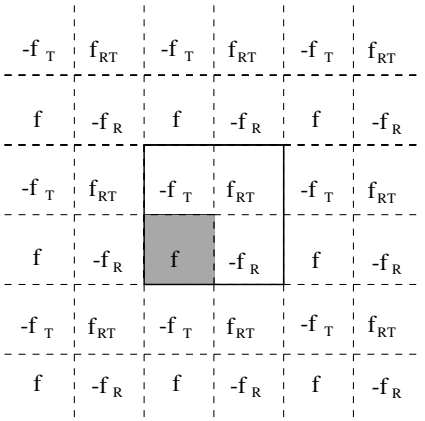

Fig. 2. A source distribution tiling the plane which solves the modified Helmholtz equation in the shaded box with homogeneous Dirichlet boundary conditions.

The second step of the algorithm solves the homogeneous modified Helmholtz equation with the prescribed boundary conditions. That is, it solves:

$$\begin{cases} \Delta u_2(\vec{x}) - \beta^2 u_2(\vec{x}) = 0 \quad \text{in } D, \\ u_2 = g_t \quad \text{on the top boundary}, \\ u_2 = g_b \quad \text{on the bottom boundary}, \\ u_2 = g_l \quad \text{on the left boundary}, \\ u_2 = g_r \quad \text{on the right boundary}. \end{cases} \tag{4}$$

Once u_1 and u_2 are available, the solution to the original problem shown in Fig. 1 is given by $u = u_1 + u_2$.

To determine u_2, we will rely heavily on the following classical result from potential theory [29,48].

Theorem 1. *Let ϕ satisfy the Yukawa equation $\Delta\phi - \beta^2\phi = 0$ in the half-space $x_2 > 0$ with Dirichlet boundary conditions $\phi(x_1,0) = f(x_1)$. Then $\phi(x_1,x_2)$ is given by the double layer potential*

$$\phi(x_1,x_2) = 2\int_{-\infty}^{\infty} \frac{\partial G}{\partial x_2}(x_1-\xi,x_2) f(\xi)\,\mathrm{d}\xi,$$

where

$$\frac{\partial G}{\partial x_2}(x_1-\xi,x_2) = \frac{\beta x_2}{2\pi\sqrt{(x_1-\xi)^2+x_2^2}} K_1\left(\beta\sqrt{(x_1-\xi)^2+x_2^2}\right),$$

with $K_1(\cdot)$ being the first-order modified Bessel function of second kind.

This result, coupled with the method of images allows us to write the solution of Eq. (4) as the potential due to a carefully laid out arrangement of double layer potentials with densities depicted in Fig. 3. Here, $D_l = 2g_l$, $D_r = 2g_r$, $D_t = 2g_t$, and $D_b = 2g_b$. $-D_{lr}$ is the odd reflection of D_l across the top boundary, $-D_{rr}$ is the odd reflection of D_r across the top boundary, $-D_{tr}$ is the odd reflection of D_t across the right boundary, and $-D_{br}$ is the odd reflection of D_b across the right boundary. It is straightforward to verify that the field induced by the set of all densities labeled D_l and $-D_{lr}$ satisfies the Dirichlet boundary condition $u = g_l$ on the left side *and zero boundary conditions on the other three sides*. Analogous results hold for the pairs $\{D_r,-D_{rr}\}$, $\{D_t,-D_{tr}\}$, and $\{D_b,-D_{br}\}$, and the principle of superposition implies that the net effect of all densities is to satisfy the desired conditions in (4).

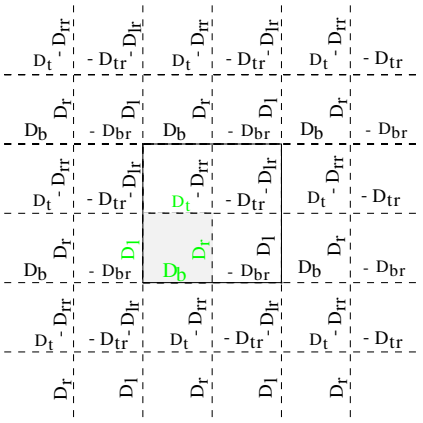

Fig. 3. Double layers for inhomogeneous Dirichlet boundary conditions.

In summary, by using an appropriate integral equation formulation, we can explicitly write the solution as a collection of volume and layer potentials whose density functions extend throughout the plane and are given explicitly by the method of images. One advantage of this formulation is that the error depends only on the discretization error of the source distribution and the accuracy with which the various integrals involved in the exact solution are carried out. Another advantage is that discontinuities and corner singularities can be handled easily in this formulation, which is not the case for finite difference and finite element based formulations.

2.2. Fast multipole acceleration

The fast multipole method (FMM) was first introduced by Greengard and Rokhlin as an efficient way to evaluate the Coulomb potential due to a collection of charged particles [8,23,27]. As shown in [20,24], the technique can be extended to volume potentials and allows for the evaluation of the field in $O(N)$ time, where N denotes the number of grid points in the discretization. Moreover, the work per grid point was shown to be comparable to that for FFT-based solvers, despite the added flexibility of adaptive mesh refinement.

The basic structure of the fast multipole method follows and we refer the reader to [8,23,27] for a more detailed discussion. Here, we focus on the fast multipole method for the free-space problem whose solution is given by the volume integral

$$u(\vec{x}) = \int \int_D G(\vec{x} - \vec{y}) f(\vec{y})\, \mathrm{d}\vec{y}. \tag{5}$$

We assume that the source distribution f is supported inside the unit square D, centered at the origin. On this square, we superimpose a hierarchy of refinements (a quad-tree). Grid level 0 is defined to be D itself, and grid level $l+1$ is obtained recursively by subdividing each square at level l into four equal parts. Using standard terminology, if s is a fixed square at level l, the four squares at level $l+1$ obtained by its subdivision will be referred to as its children. Allowing for adaptivity is simple since one does not need to use the same number of levels of subdivision in all regions of D. However, we do assume that the quad-tree satisfies one fairly standard restriction. We require that two leaf nodes which share a boundary point must be no more than one refinement level apart (see [2,20,36]).

Denoting the childless leaf nodes in the quad-tree structure by D_i for $i = 1, \ldots, M$ where M is the total number of such node cells, we assume that we are given f on a cell-centered 4×4 grid for each D_i. Then

$N = 16 \times M$ is the total number of grid points in D. To obtain fourth-order accuracy, these 16 data points are used to construct a fourth-order polynomial approximation to f of the form

$$f_{D_i}(y_1, y_2) \approx \sum_{j=1}^{10} c_{D_i}(j) b_j(y_1 - y_{1_{D_i}}, y_2 - y_{2_{D_i}}), \tag{6}$$

where $(y_{1_{D_i}}, y_{2_{D_i}})$ denotes the center of D_i and the basis functions $\{b_j(y_1,y_2)|j = 1,\ldots,10\}$ are given by $\{y_1^{j_1} y_2^{j_2} | j_1, j_2 \geqslant 0, j_1 + j_2 \leqslant 3\}$. Therefore, the solution in (5) can be approximated by

$$u(\vec{x}) \approx \sum_{i=1}^{M} \int\int_{D_i} G(x_1 - y_1, x_2 - y_2) \sum_{j=1}^{10} c_{D_i}(j) b_j(y_1 - y_{1_{D_i}}, y_2 - y_{2_{D_i}}) \, dy_1 \, dy_2. \tag{7}$$

Further, to evaluate this potential at the N grid points, the $O(N^2)$ work required using direct summation can be reduced to $O(N)$ using the fast multipole method. The basic notion underlying the FMM is that for each grid point, contributions from "nearby" cells (neighbors) to the potential field can be handled directly, while "far-field" (non-neighbor) interactions are handled using multipole and related types of expansions.

Definition 1. Consider a square B in the adaptive quad-tree structure shown in Fig. 4 (see also [20]):

- The *colleagues* (labeled "c") of a square B are squares at the same refinement level which share a boundary point with B. B is considered to be a colleague of itself.
- When B is a leaf node:
  1. The *coarse neighbors* (labeled "n^+") of B are leaf nodes at the level of B's parent which share a boundary point with B.
  2. The *fine neighbors* (labeled "n^-") of B are leaf nodes one level finer than B which share a boundary point with B.
  3. The *s-list* (labeled "s") of B consists of those children of B's colleagues which are not fine neighbors of B.
- Together, the union of the colleagues, coarse neighbors, fine neighbors, and s-list members of a node B will be referred to as B's *neighbors*.

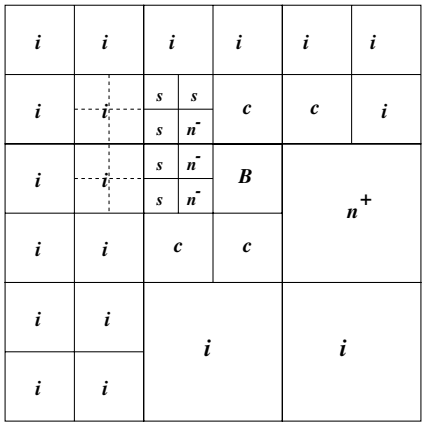

Fig. 4. Neighbors and interaction list of a box B in an adaptive tree structure.

The first step in the fast multipole method is to form the multipole expansions for all the nodes in the quad-tree structure. This is done through the use of Graf's addition theorem [1].

Theorem 2. (Graf's addition theorem) *Assume the polar coordinates for points* $\vec{x} = (x_1, x_2)$ *and* $\vec{y} = (y_1, y_2)$ *are given by* $(|\vec{x}|, \theta_{\vec{x}})$ *and* $(|\vec{y}|, \theta_{\vec{y}})$, *respectively*, *with* $|\vec{x}| > |\vec{y}|$, *then*

$$K_0(\beta|\vec{x} - \vec{y}|) = \sum_{l=-\infty}^{\infty} K_l(\beta|\vec{x}|) I_l(\beta|\vec{y}|) \mathrm{e}^{\mathrm{i}l(\theta_{\vec{y}} - \theta_{\vec{x}})}.$$

By applying Theorem 2, the "far-field" multipole expansion for a leaf node is then given by (see also [23,25,27]).

Theorem 3. *Let s be a node in the quad-tree centered at* $S = (s_1, s_2)$. *Assume s is not a neighbor of B*, *then*, *the potential* $\Phi(\vec{x})$ *due to s for* $\vec{x} \in B$ *is given by the multipole expansion*

$$\Phi(\vec{x}) = \int\int_s \frac{-1}{2\pi} K_0(\beta|\vec{x} - \vec{y}|) \sum_{j=1}^{10} c_s(j)\ b_j(y_1 - s_1, y_2 - s_2)\ \mathrm{d}y_1\ \mathrm{d}y_2 = \sum_{l=-\infty}^{l=\infty} M_l K_l(\beta|\vec{x} - S|) \mathrm{e}^{-\mathrm{i}l\theta_{\vec{x}}}, \tag{8}$$

where the multipole coefficients are given by

$$M_l = \frac{-1}{2\pi} \sum_{j=1}^{10} c_s(j) \int\int_s I_l(\beta|\vec{y} - S|) \mathrm{e}^{\mathrm{i}l\theta_{\vec{y}}} b_j(y_1 - s_1, y_2 - s_2)\ \mathrm{d}y_1\ \mathrm{d}y_2. \tag{9}$$

Here, I_l *and* K_l *are the lth order modified Bessel's function of the first and second kind respectively*, *and* $\theta_{\vec{x}}$ *and* $\theta_{\vec{y}}$ *denote the angular polar coordinates for* $\vec{x}$ and $\vec{y}$ *with respect to the expansion center S.*

The multipole expansion of a parent node is derived by merging the shifted multipole expansions of its four children. The "multipole to multipole" translation operator which shifts the center of the expansion is also a direct result of Graf's addition theorem:

Theorem 4. *Suppose the multipole expansion centered at S is given by*

$$\Phi(\vec{x}) = \sum_{l=-\infty}^{\infty} a_l K_l(\beta\rho) \mathrm{e}^{\mathrm{i}l\theta},$$

where (ρ,θ) *denote the polar coordinates of the evaluation point* $\vec{x}$ *with respect to S. Then the corresponding multipole expansion centered at* $S_{new} = (\rho_0,\theta_0)$ *is given by*

$$\Phi(\vec{x}) = \sum_{l=-\infty}^{\infty} b_l K_l(\beta\rho_{\mathrm{new}}) \mathrm{e}^{\mathrm{i}l\theta_{\mathrm{new}}},$$

where ρ_{new} *and* θ_{new} *are the new polar coordinates of* $\vec{x}$ *with respect to* S_{new}, *and the new multipole coefficients are expressed using the translation operator* $\mathcal{T}_{\mathrm{MM}}$ *defined by*

$$b_l = \sum_{m=-\infty}^{\infty} a_m I_{l-m}(\beta\rho_0) \mathrm{e}^{-\mathrm{i}(l-m)\theta_0}. \tag{10}$$

In the second step of the fast multipole method, it is observed that for each node B, the contribution from far-field nodes is locally "smooth". Hence we can associate to each box a *local expansion* which collects information from the far-field multipole expansions. The local expansion is given by

$$\Phi(\vec{x}) = \sum_{l=-\infty}^{\infty} L_l I_l(\beta\rho)\mathrm{e}^{\mathrm{i}l\theta}, \tag{11}$$

where (ρ,θ) denote the polar coordinates of $\vec{x}$ with respect to the center of B.

For each node B, its local expansion consists of two different contributions, one from distant boxes which make up the *interaction list* described in Definition 2, and the other inherited from its parent which contains the far-field contributions outside the interaction list.

Definition 2. In the adaptive quad-tree structure shown in Fig. 4, the *interaction list* (labeled "*i*") for a square B consists of those children boxes from the colleagues of B's parent, excluding B's colleagues; and those childless colleagues of B's parent, excluding B's coarse neighbors [20,23,27].

In the original FMM, to translate multipole expansions from nodes in the interaction list to local ones, we apply the following result:

Theorem 5. *Suppose a multipole expansion associated with node s centered at S is given by*

$$\Phi(\vec{x}) = \sum_{n=-\infty}^{\infty} M_n K_n(\beta|\vec{x} - S|)\mathrm{e}^{-\mathrm{i}n\theta_{\vec{x}}}. \tag{12}$$

Further assume s is in the interaction list of node B. Then for any $\vec{x} \in B$, $\Phi(\vec{x})$ can be represented as a local expansion

$$\Phi(\vec{x}) = \sum_{l=-\infty}^{\infty} L_l I_l(\beta\rho)\mathrm{e}^{\mathrm{i}l\theta},$$

where (ρ,θ) denote the polar coordinates of $\vec{x}$ with respect to the center of B, and the local coefficients are expressed using the "multipole to local" translation operator $\mathscr{T}_{\mathrm{ML}}$ defined by

$$L_l = \sum_{n=-\infty}^{\infty} M_n K_{l-n}(\beta\rho_0)\mathrm{e}^{-\mathrm{i}(l-n)\theta_0}. \tag{13}$$

Here (ρ_0,θ_0) are the polar coordinates of B's center with respect to S.

The process described in Theorem 5 requires approximately $27p^2$ work per node, where p denotes the length of the truncated multipole expansions and 27 is the maximal possible members in the interaction list. This is the dominating part of the operation count in the FMM. It will be shown in Appendix C that introducing the plane wave representations and "diagonal translation operator", the work amount for $\mathscr{T}_{\mathrm{ML}}$ can be reduced to approximately $3p^2$. Now, in order for the box to inherit its parent's local expansion, we apply the following theorem:

Theorem 6. *For the parent's local expansion centered at S given by*

$$\Phi = \sum_{k=-\infty}^{\infty} a_k I_k(\beta\rho)\mathrm{e}^{\mathrm{i}k\theta},$$

the corresponding local expansion for the child centered at (ρ_0,θ_0) is given by

$$\Phi = \sum_{m=-\infty}^{\infty} b_m I_m(\beta\rho_{\mathrm{new}})\mathrm{e}^{\mathrm{i}m\theta_{\mathrm{new}}},$$

where the new coefficients are computed using the "local to local" translation operator $\mathscr{T}_{\mathrm{LL}}$ defined by

$$b_m = \sum_{k=-\infty}^{\infty} a_k I_{m-k}(\beta\rho_0)\mathrm{e}^{-\mathrm{i}(m-k)\theta_0}. \tag{14}$$

Finally, in the last step of the fast multipole method, if $\vec{x}$ is a grid point in a leaf node B, we evaluate the local expansion which contains all the contribution from the "far-field", and calculate the "nearby" local interactions

$$\sum_{j=1}^{10} c_{D_i}(j)\left(\int\int_{D_i} G(x_1 - y_1, x_2 - y_2) b_j(y_1 - y_{1_{D_i}}, y_2 - y_{2_{D_i}})\,\mathrm{d}y_1\,\mathrm{d}y_2\right) \tag{15}$$

directly by mapping the coefficients $c_{D_i}(j)$ of those childless leaf nodes found in (6) to the function value at $\vec{x}$ using precomputed tables, and by evaluating the multipole expansions for those s-list members with children. The readers are referred to Section 3.2 for further discussions.

2.3. The pseudocode

We summarize the description of the fast multipole method given in Section 2.2 using the following pseudocode:

Initialization
Comment [p denotes the order of the multipole expansion determined by the desired accuracy ϵ_{FMM}. $l_{\max}$ denotes the maximum refinement level in the quad-tree structure determined by the prescribed precision ϵ_{rhs}.]
Generate the "level-restricted" adaptive quad-tree and the precomputed table of coefficients.
Step I: Far-field Interaction – Upward Pass
Comment [In upward pass, multipole expansions for all nodes in the tree structure will be formed.]

for $l = l_{\max},\ldots,0$
  for all boxes j on level l
    if j is a leaf node
      form the multipole expansion $\Phi_{l,j}$ using Eq. (8).
    else
      form the multipole expansion $\Phi_{l,j}$ by merging the expansions of its children using the operator $\mathscr{T}_{\mathrm{MM}}$ from Eq. (10).
    endif
  end
end

Cost [The upward pass requires approximately Mp^2 work, where M is the number of leaf nodes.]
Step II: Far-field Interaction – Downward Pass
Comment [In downward pass, local expansion will be formed for each box to collect far-field contributions.]
initialize the local expansion $\Psi_{0,0} = 0$.

for $l = 1,\ldots,l_{\max}$
  for all boxes j on level l
    Compute $\widetilde{\Psi}_{l,j}$ by shifting its parent's Ψ expansion using the operator $\mathscr{T}_{\mathrm{LL}}$ from Eq. (14).

Compute $\Psi_{l,j}$ by adding in the contributions from all boxes in j's interaction list using $\mathscr{T}_{\mathrm{ML}}$ from Eq. (13). (alternatively, see Eqs. (27)–(29) in Appendix C.)

end

end

Cost [The downward pass requires approximately $28Mp^2$ work using Eq. (13), or $3Mp^2$ work using Eqs. (27)–(29).]

Step III: Local interactions

Comment [At this point, for each leaf node D_i, its local expansion contains the influence of the source distribution f over all leaf nodes D_j outside the neighbors of D_i.]

for $i = 1,\ldots,M$

for each target point $\vec{x}$ in D_i,

evaluate D_i's local expansion at $\vec{x}$,

evaluate the influence of each neighbor using the precomputed table of coefficients. (see Section 3.2)

end

end

Cost [The maximum number of neighbors a box can have is thirty-three (twelve fine neighbors, twenty s-list members, and itself). Thus, the local work is approximately $33 \cdot N$ operations.]

3. Algorithm details

Compared with the Poisson equation solver in [20], one important difference in the present case is that the modified Helmholtz equation depends on an extra parameter β. Thus, many of the building blocks of the FMM for the Poisson context have to be changed accordingly. In this section, we discuss these implementation details.

3.1. Scaling

Consider the multipole and local expansions given in Eqs. (8) and (11). For small z, $I_l(z) \sim (\frac{1}{2}z)^l/\Gamma(l+1)$ and $K_l(z) \sim \frac{1}{2}\Gamma(l)(\frac{1}{2}z)^{-l}$. Thus, one can easily encounter underflow and overflow in numerical calculations. To avoid loss of accuracy, the Bessel functions, as well as the coefficients, have to be scaled appropriately. For this, we introduce the scaled multipole and local coefficients $\widetilde{M}_l$ and $\widetilde{L}_l$ defined by:

$$\widetilde{M}_l = M_l \cdot \left[\Gamma(|l|+1)\left(\frac{1}{2}\beta h\right)^{-|l|}\right],$$

$$\widetilde{L}_l = L_l \Big/ \left[\Gamma(|l|+1)\left(\frac{1}{2}\beta h\right)^{-|l|}\right].$$

We define the scaled Bessel functions by:

$$\widetilde{K}_l = K_l \Big/ \left[\Gamma(|l|+1)\left(\frac{1}{2}\beta h\right)^{-|l|} \right],$$

$$\widetilde{I}_l = I_l \cdot \left[\Gamma(|l|+1)\left(\frac{1}{2}\beta h\right)^{-|l|} \right].$$

Clearly $\sum M_l K_l = \sum \widetilde{M}_l \widetilde{K}_l$ and $\sum L_l I_l = \sum \widetilde{L}_l \widetilde{I}_l$. In practice, instead of (9), we compute the scaled multipole moments from the formula

$$\widetilde{M}_l = \sum_{j=1}^{10} c_s(j) \cdot \left(\frac{h}{2}\right)^{J(j)} \cdot W(l,j), \tag{16}$$

where

$$W(l,j) = \frac{-1}{2\pi} \int_{-1}^{1} \int_{-1}^{1} \widetilde{I}_l\left(\frac{\beta h}{2}\sqrt{y_1^2+y_2^2}\right) \mathrm{e}^{-\mathrm{i}l\theta} y_1^{j_1} y_2^{j_2} \, \mathrm{d}y_1 \, \mathrm{d}y_2. \tag{17}$$

The weights $W(l,j)$ in (17) used in computing the multipole coefficients are functions of βh only. Thus, we can precompute and tabulate these weights at selected values of the variable βh and recover them for arbitrary values using high-order interpolation. Details of this step are discussed in the following section.

3.2. Local interactions and the precomputed coefficients

The efficiency of our algorithm is improved dramatically by precomputing the linear mappings used in the evaluation of the local interactions in (15). For a given node B, the local contributions include those from its colleagues, its fine neighbors (which will be referred to as "small to big" interactions), its coarse neighbors ("big to small"), and its s-list members ("small to big far").

As an example, consider the "small to big" local interactions, and let us index the 16 uniform grid points in the large center cell of Fig. 5 by p_m for $m = 1,\ldots,16$. Further, let the 12 small boxes in the figure be indexed by s_n for $n = 1,\ldots,12$. For simplicity, we assume the coordinate system is translated so that box s_1 is centered at the origin with side length h. Then upon approximating the source function f in s_1 by a 4th degree polynomial as in (6), the potential due to s_1 at the point $p_m = (x_1(m),x_2(m))$ is given by

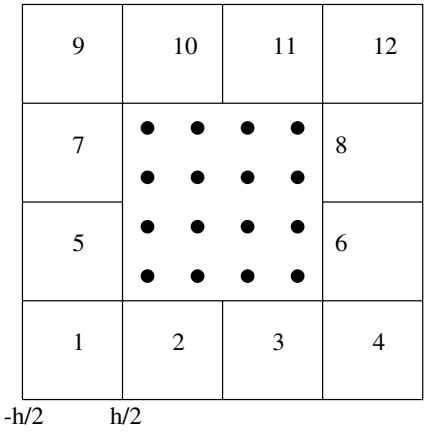

Fig. 5. Potential at 16 uniform grid points in large cell due to sources in 12 small boxes.

$$\Phi(x_1,x_2) = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{-\frac{h}{2}}^{\frac{h}{2}} \sum_{j=1}^{10} c_s(j) G(x_1(m)-y_1, x_2(m)-y_2) b_j(y_1,y_2)\, \mathrm{d}y_1\, \mathrm{d}y_2$$
$$= \sum_{j=1}^{10} c_s(j) h^{j_1+j_2+2} \cdot W(s_1,p_m,j). \tag{18}$$

After introducing the scaled variables $\tilde{x}_i = \frac{x_i}{h}$ and $\tilde{y}_i = \frac{y_i}{h}$ for $i = 1$ and 2,

$$W(s_1,p_m,j) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{-1}{2\pi} K_0\left(\beta h \sqrt{(x_1(\tilde{m})-\tilde{y}_1)^2 + (x_2(\tilde{m})-\tilde{y}_2)^2}\right) \cdot \tilde{y}_1^{\,j_1} \cdot \tilde{y}_2^{\,j_2}\, \mathrm{d}\tilde{y}_1\, \mathrm{d}\tilde{y}_2.$$

Notice that $W(s_1,p_m,j)$ is a function of βh only. Thus, it can be approximated by a Chebyshev expansion over a large range of βh. For instance, to obtain a relative approximation error less than 10^{-13}, numerical experiments indicate that for s_1, p_1 and $j = 1$, 21-point Chebyshev on a total of 9 intervals $\{I_i = [a_i,a_{i+1}], i = 0,\ldots,8\}$ is sufficient. For this, the interval endpoints a_i are given by {0, 1, 3, 8, 20, 50, 110, 175, 290, 10^{15}}. Thus, 9×21 values need to be precomputed for the function $W(s_1,p_1,1)$. Similar results apply for each source box and each target point. In our code, for a given β, the linear mapping from the coefficients $c_s(j)$ to the values $\Phi(x_1(m),x_2(m))$ is precomputed for all levels in the initialization step by evaluating the Chebyshev approximations of $W(s_n,p_m,j)$. The "small to big" interactions are then calculated using (18).

As for storage, there are at most 12 source boxes s_n, 16 target points p_m, and 10 basis functions $c(j)$. With 9 intervals and 21 Chebyshev nodes per interval for each function, the size of the table is approximately $12 \times 16 \times 10 \times 9 \times 21$ which consumes approximately 400 kb of memory. It is possible to further reduce the size of the table by symmetry considerations with the consequence of a slightly more complicated code and a negligible sacrifice of speed. We omit the details and refer the interested readers to [20].

As indicated earlier, precomputed tables are used in other steps of our algorithm as well. In (6), the approximating polynomials are computed by mapping the source function values at the 16 grid points in a cell to 10 coefficients, where the linear mapping matrix is derived using a precomputed table based on a least squares fit. In (17), the coefficients are also precomputed and stored as Chebyshev polynomial coefficients.

3.3. Boundary conditions and the lattice sums

As noted in Section 2.1, problems with Dirichlet boundary conditions for the unit square can be solved using double layer and volume potentials based on the method of images. Using classical results similar to Theorem 1, this technique can be generalized to problems with Neumann or mixed boundary conditions. That is, the solution can be represented as a combination of volume, single, and/or double layer potentials for rectangular structures where the density functions are explicitly given by the right hand side of the governing equation and the boundary conditions as well as their images. Again, these images tile the plane periodically. In this section, we leave the derivation of the periodic "super-cell" structure for other boundary conditions to [20] or classical textbooks on potential theory and consider only the periodic cells shown in Figs. 2 and 3. We impose periodic boundary conditions on these structures using what are known as lattice sums. For simplicity of notation, we assume the fundamental unit super-cell (outlined with solid lines) in these figures is centered at the origin.

Since the plane is tiled by identical copies of the super-cell, they all have the same multipole coefficients, i.e., an image cell denoted by S_{k_1,k_2} has a multipole expansion of the form

$$\Phi(\vec{x}) = \sum_{n=-p}^{p} M_n K_n(\beta|\vec{x}-\vec{c}_{k_1,k_2}|) \mathrm{e}^{-\mathrm{i}n\theta_{\vec{x}-\vec{c}_{k_1,k_2}}},$$

where $\theta_{\vec{x}-\vec{c}_{k_1,k_2}}$ is the angle of $\vec{x}$ with respect to the cell center denoted by $\vec{c}_{k_1,k_2}$. We are only interested in the field within the cell centered at the origin. Thus, the contributions from all distant image cells can be collected as a single local expansion using the multipole to local translation operator $\mathscr{T}_{\mathrm{ML}}$ in (13). Define Λ to be the set of all integer lattice points (k_1,k_2) in the plane, except the origin and its eight nearest neighbors, and denote the polar coordinates of $\vec{c}_{k_1,k_2}$ with respect to the origin by (ρ_S,θ_S). The local coefficients are given by

$$L_l = \sum_{n=-p}^{p} M_n \sum_{(k_1,k_2)\in\Lambda} K_{l-n}(\beta\rho_S)\mathrm{e}^{-\mathrm{i}(l-n)\theta_S},$$

where $S = S_{k_1,k_2}$ is the far-field cell. Further, introduce the concept of *lattice sums* defined as

$$S_n(\beta) = \sum_{S\in\Lambda} K_n(\beta\rho_S)\mathrm{e}^{\mathrm{i}n\theta_S}, \quad n = 0,\ldots,\infty, \tag{19}$$

the linear mapping from the multipole to local coefficients then becomes

$$L_l = \sum_{n=-p}^{p} S_{l-n}M_n. \tag{20}$$

Once the local expansion for the fundamental super-cell has been created, what remains can be embedded in a fast multipole code, which is only slightly more involved than the one described above for free-space problems. The details are technical and involve essentially bookkeeping considerations.

In order for the method to be efficient, however, it is convenient to have a fast algorithm to evaluate the lattice sums themselves. There are a number of such schemes. Ewald-like methods work by splitting the sum into two pieces [4,21,43,47], where the first piece decays rapidly in the "physical domain", while the second part converges rapidly in the Fourier domain. Other recent approaches include Berman and Greengard's renormalization method [3], Helsing and Lambert's direct method using fast multipole acceleration [30,38], and Chin, Nicorovici and McPhedran's method using absolutely converging series [13]. In [18,34], new integral formulas were developed for Coulombic and Helmholtz lattice sums based on exponential ("plane wave") expansions similar to those in Theorem 9 of Appendix A. The lattice sums become one-dimensional integrals in the two-dimensional case and are easily computed using Gaussian quadrature [18,34].

4. Error analysis

As indicated earlier, an important feature of a direct integral-transform based solution procedure is that error analysis takes a particularly simple form. Unlike finite difference and finite element methods, there is no need for convergence estimates based on smoothness properties of the (unknown) solution. There are precisely three sources of error: the discretization/approximation of the data (the source distributions and boundary conditions); the truncation of the infinite series (multipole, local, and exponential expansions); and the calculation of local interactions. The following error control strategies are applied:

1. The discretization error is handled by adaptive refinement of the data. The quad-tree structure is generated so that the relative discretization error is less than the desired tolerance ϵ_{rhs}.
2. For extremely large β in a unit box (or βh in scaled boxes), contributions from far-field boxes are exponentially small and easily neglected.

3. The number of terms in the truncated expansions is determined by the prescribed accuracy ϵ_{FMM}. Under fairly reasonable assumptions, the truncated multipole and local expansions with 21 terms guarantee single precision, while 42 terms give double precision.
4. The relative accuracy in direct local interactions is set to 13 digits using the precomputed tables discussed in Section 3.2.

Of these strategies, control of the discretization and local interaction errors (items (1) and (4) above) are easily understood. The error due to truncation of the infinite series is very complicated and we refer the readers to [17,19] for more detailed discussion. It is, perhaps, not so surprising that the number of terms in the expansion for screened Coulomb interactions is (at most) the number of terms needed for the unscreened case. The number of terms cited above (21 for single precision and 42 for double precision) are essentially the same as for the Poisson equation.

5. Numerical examples

The fast modified Helmholtz solver has been implemented in Fortran77. The following numerical examples are performed on a 450 MHZ SUN Ultra-80 with 1024 Mb memory.

As a first test, we consider the free-space problem

$$\Delta\psi(\vec{x}) - 0.01\psi(\vec{x}) = f(\vec{x}), \tag{21}$$

whose exact solution is given by

$$\psi(\vec{x}) = \sum_{i=1}^{3} \mathrm{e}^{-\alpha|\vec{x}-\vec{x}_i|^2}, \tag{22}$$

where $\alpha = 50$, $\vec{x}_1 = (0.1, 0.1)$, $\vec{x}_2 = (0, 0)$, and $\vec{x}_3 = (-0.15, 0.1)$. As the source distribution $f(\vec{x})$ decays rapidly for large $\vec{x}$, we may therefore assume $f(\vec{x})$ is compactly supported in the unit square D so that the solution is given by $\int\int_D G(\vec{x}-\vec{y})f(\vec{y})\,\mathrm{d}s_{\vec{y}}$. The adaptive mesh structure based on AMR [2,36] is shown in Fig. 6, which resolves $f(\vec{x})$ to 3 digits. (Three levels of refinement are present.) The exact solution is shown in Fig. 7.

To test efficiency, we first compare our solver with existing FFT-based algorithms. Since our solver is adaptive and because traditional FFT-based solvers only allow for uniform grids and simple geometry, we compare the speed by measuring how many grid points each solver can process in one second. In Table 1, we list the averaged results for the FFT based second-order HWSCRT [49,50] which, compared with other FFT-based solvers (the fourth-order Mehrstellen methods and the sixth-order FFT9 [15,32]), is readily available and allows for different boundary conditions. In the table, N denotes the number of grid points, E_∞ denotes the "relative" L_∞ error of the computed solution $(\frac{\|\psi_{\mathrm{num}}-\psi_{\mathrm{anal}}\|_\infty}{\|\psi_{\mathrm{anal}}\|_\infty})$ where ψ_{num} is the numerical approximation to the exact solution ψ_{anal} in (22), T_{hwscrt} denotes the required solution time in seconds, and *Rate* denotes the number of grid points "processed" per second (N/T_{hwscrt}).

The timing results for the new fourth-order solver are given in Table 2. Here, ϵ_{FMM} denotes the requested precision from far-field interactions within the fast multipole method, which determines the number of terms in different truncated expansions, ϵ_{rhs} denotes the requested precision in discretizing $f(\vec{x})$, N_{lev} denotes the number of levels used in the FMM hierarchy, E_∞ denotes the relative L_∞ error, N_{free} denotes the number of grid points used, T_{fmm} denotes the required solution time in seconds, and *Rate* denotes the number of grid points "processed" per second. In all cases, the prescribed accuracy is obtained by our new solver. Compared with Table 1, the FMM-based solver is approximately 3–6 times slower than the FFT based HWSCRT. We want to mention that we have not carried out low-level optimization of the current code, which we expect would reduce the cost by a significant factor.

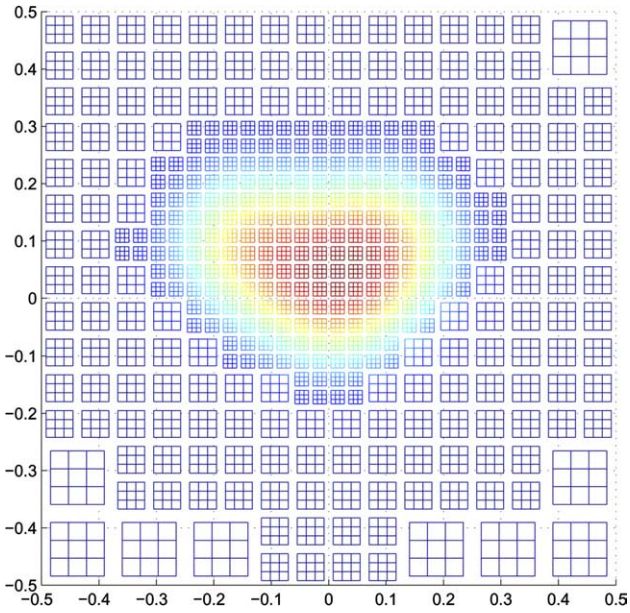

Fig. 6. The adaptive grid structure for test problem 1.

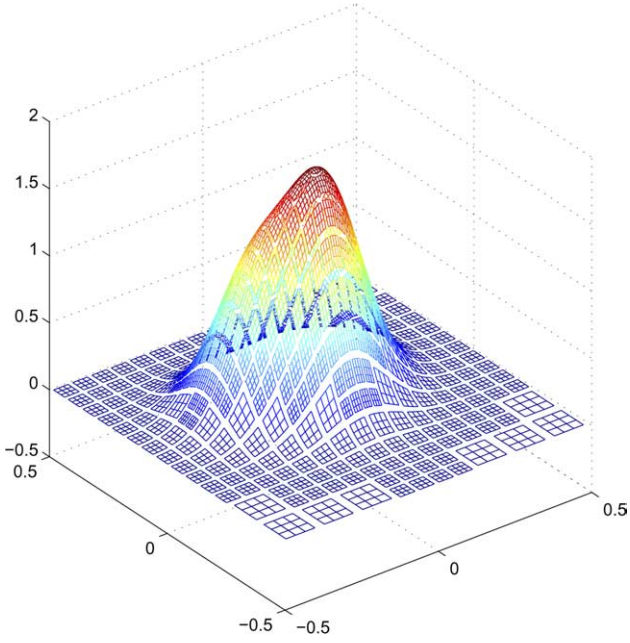

Fig. 7. The exact solution of test problem 1.

In example 2, we consider the same equation as in test problem (21), but with periodic boundary conditions. In Table 3, we can conclude that the additional time required for treatment of the boundary conditions is less than 20% extra of that of the free-space solver. The reader may note that the number of grid

Table 1
Timing results for HWSCRT

N	E_∞	T_{hwscrt}	Rate
$128 \times 128 = 16{,}384$	4.8×10^{-4}	0.19	3.4×10^5
$256 \times 256 = 65{,}536$	1.1×10^{-4}	0.19	3.4×10^5
$512 \times 512 = 262{,}144$	3.0×10^{-5}	0.85	3.1×10^5
$1024 \times 1024 = 1{,}048{,}576$	7.4×10^{-6}	4.5	2.3×10^5
$2048 \times 2048 = 4{,}194{,}304$	1.9×10^{-6}	19.9	2.1×10^5

Table 2
Timing results for fourth-order FMM accelerated Yukawa solver

ϵ_{FMM}	ϵ_{rhs}	N_{lev}	E_∞	N_{free}	T_{fmm}	Rate
10^{-3}	10^{-3}	5	7.0×10^{-4}	8128	0.09	9.0×10^4
10^{-3}	10^{-6}	7	7.8×10^{-5}	51,952	0.45	1.2×10^5
10^{-6}	10^{-6}	7	1.6×10^{-7}	51,952	0.74	7.0×10^4
10^{-6}	10^{-9}	8	5.5×10^{-8}	714,688	9.00	8.0×10^4
10^{-9}	10^{-9}	8	9.0×10^{-10}	714,688	12.41	5.8×10^4

Table 3
Timing comparison for free-space and periodic boundary conditions

ϵ_{FMM}	ϵ_{rhs}	N_{free}	N_{periodic}	T_{free}	T_{periodic}	$\text{Rate}_{\text{free}}$	$\text{Rate}_{\text{periodic}}$
10^{-3}	10^{-3}	8812	8812	0.09	0.09	9.0×10^4	9.0×10^4
10^{-3}	10^{-6}	51,952	52,096	0.45	0.51	1.2×10^5	1.0×10^5
10^{-6}	10^{-6}	51,952	52,096	0.74	0.79	7.0×10^4	6.6×10^4
10^{-6}	10^{-9}	714,688	716,176	9.00	9.10	8.0×10^4	7.9×10^4
10^{-9}	10^{-9}	714,688	716,176	12.41	12.58	5.8×10^4	5.7×10^4

points in the periodic case denoted by N_{periodic} is slightly increased compared with N_{free}. This is due to the requirement we impose on the quad-tree that no two neighboring boxes can be more than one refinement level apart.

The fact that our solver guarantees the number of digits of accuracy so long as the source distribution and boundary conditions are discretized to the same precision is extremely useful when considering discontinuous boundary conditions or source distributions.

To further test the accuracy of the algorithm, in example 3, we consider the same equation as in test problem (21), but specify the Dirichlet boundary conditions using the exact solution. The error distribution is shown in Fig. 8. Note that we obtained the prescribed accuracy. Note also that the error is not smooth – due in part to the truncation of the various expansions used in the algorithm.

As the last example, we consider the equation

$$u(x_1, x_2) - \epsilon \Delta u(x_1, x_2) = f(x_1, x_2)$$

with $\epsilon = 10^{-5}$. We use the same exact solution (22) and the exact Dirichlet boundary conditions. We set the required accuracy to 6 digits. The numerical result is shown in Fig. 9. Although the error is highly oscillatory, the prescribed accuracy is met. We have tested numerous problems with ϵ as small as 10^{-20} and the prescribed accuracy is always obtained.

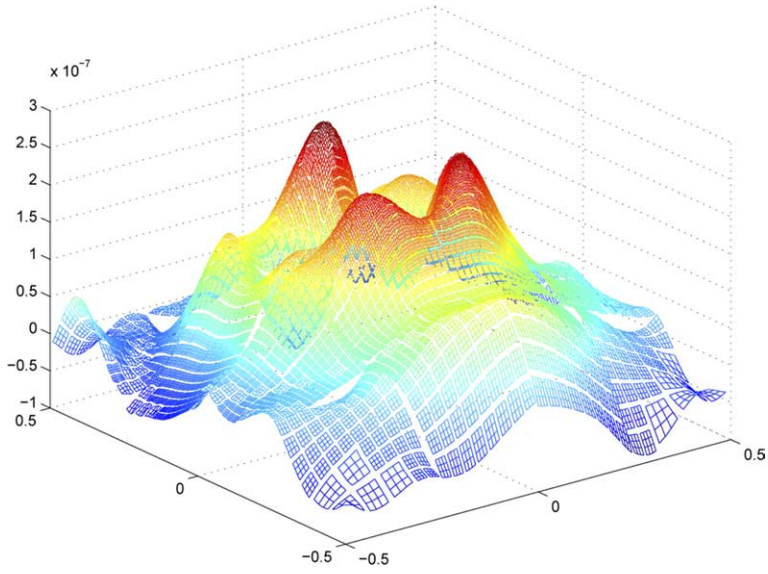

Fig. 8. Error distribution for the Dirichlet boundary conditions.

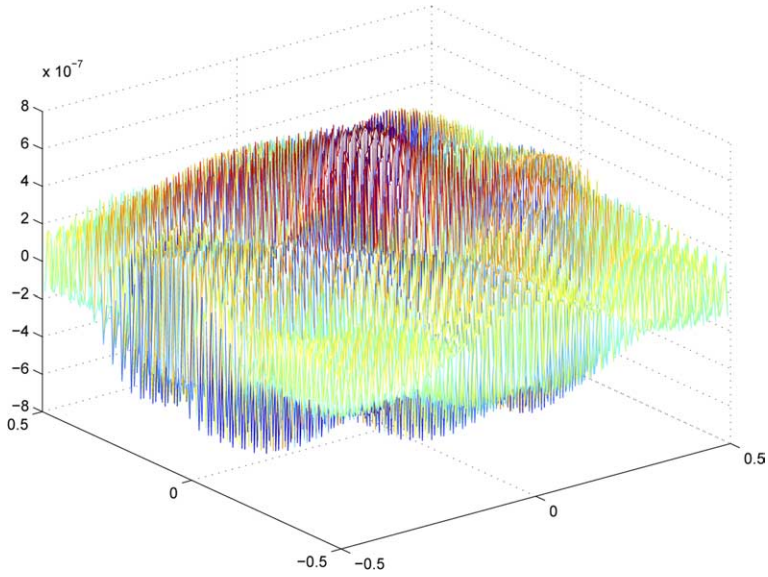

Fig. 9. Error distribution for small ϵ.

6. Conclusion

In this paper, we have developed a new fast direct solver for the modified Helmholtz equation on the unit box. The solver is based on an integral equation formulation, accelerated by the new version of fast multipole method, and handles different kinds of boundary conditions. The solver allows for adaptive mesh refinement where the amount of work scales linearly with the total number of grid points, and its efficiency

is very competitive with FFT based methods. The code has been tested for a large variety of boundary conditions and, in all cases, achieved the prescribed accuracy.

Currently, the code is being optimized for performance and implemented on a variety of parallel architectures. Generalization of the solver to three dimensions is being considered. Except for several technical details including the optimal quadrature nodes for plane wave expansions and the more complicated adaptive tree structure, such generalization is straightforward. Complex geometry is also being considered through the coupling of the volume integral scheme described here with a boundary integral representation using an unknown surface "charge" density. The techniques in this solver can be applied to other equations, including the low frequency Helmholtz equation which is widely used in computational electromagnetics. Results along these directions will be reported at a later date.

Acknowledgments

We express our gratitude to Professor Leslie Greengard of Courant Institute and Dr. Frank Ethridge of Madmax Optics Inc. for many helpful discussions. Part of the work was finished while JH was a visiting scholar at the Institute of Mathematical Sciences at the Chinese University of Hong Kong. J.H. is very thankful for their support.

Appendix A. Plane wave expansions

For the two-dimensional Green's function of the modified Helmholtz equation (zeroth-order term in the multipole expansion), the following plane wave expansion formula is well known ([42], p. 823).

Theorem 7. *For a source point* $s = (x,y)$, *with* $x > 0$, *whose polar coordinates are given by* (r,θ),

$$K_0(\beta r) = \frac{1}{2}\int_0^\infty \frac{e^{-\sqrt{\lambda^2+\beta^2}x}}{\sqrt{\lambda^2+\beta^2}}\left[e^{i\lambda y} + e^{-i\lambda y}\right] d\lambda. \tag{23}$$

For higher order terms, the following theorem ([31], p. 127) is applied.

Theorem 8. *Suppose* $f_n(x_1,x_2,\ldots,x_p)$ *is a rational algebraic homogeneous function of degree* n, *and* $F(r^2)$ *is any function of* $r^2 = x_1^2 + x_2^2 + \cdots + x_p^2$, *then*

$$f_n\left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_p}\right) F(r^2) = \left\{2^n \frac{d^n F}{d(r^2)^n} + \frac{2^{n-2}}{1!}\frac{d^{n-1}F}{d(r^2)^{n-1}}\Delta + \cdots + \frac{2^{n-2k}}{k!}\frac{d^{n-k}F}{d(r^2)^{n-k}}\Delta^{n-k} + \cdots\right\} \times f_n(x_1, x_2, \ldots, x_p). \tag{24}$$

Plane wave expansions for different directions can be derived.

Theorem 9. *For a point* (x,y), *with* $x > 0$, *whose polar coordinates are given by* (r,θ),

$$K_n(\beta r)e^{in\theta} = \frac{1}{2}\int_0^\infty \frac{e^{-x\sqrt{\lambda^2+\beta^2}}}{\sqrt{\lambda^2+\beta^2}}\left[e^{i\lambda y}\left(\frac{\sqrt{\lambda^2+\beta^2}+\lambda}{\beta}\right)^n + e^{-i\lambda y}\left(\frac{\sqrt{\lambda^2+\beta^2}-\lambda}{\beta}\right)^n\right] d\lambda.$$

For $x < 0$,

$$K_n(\beta r)e^{in\theta} = \frac{(-1)^n}{2}\int_0^\infty \frac{e^{x\sqrt{\lambda^2+\beta^2}}}{\sqrt{\lambda^2+\beta^2}}\left[e^{i\lambda y}\left(\frac{\sqrt{\lambda^2+\beta^2}-\lambda}{\beta}\right)^n + e^{-i\lambda y}\left(\frac{\sqrt{\lambda^2+\beta^2}+\lambda}{\beta}\right)^n\right] d\lambda.$$

For $y > 0$,

$$K_n(\beta r)e^{in\theta} = \frac{i^n}{2}\int_0^\infty \frac{e^{-y\sqrt{\lambda^2+\beta^2}}}{\sqrt{\lambda^2+\beta^2}}\left[e^{i\lambda x}\left(\frac{\sqrt{\lambda^2+\beta^2}-\lambda}{\beta}\right)^n + e^{-i\lambda x}\left(\frac{\sqrt{\lambda^2+\beta^2}+\lambda}{\beta}\right)^n\right] d\lambda.$$

For $y < 0$,

$$K_n(\beta r)e^{in\theta} = \frac{(-i)^n}{2}\int_0^\infty \frac{e^{y\sqrt{\lambda^2+\beta^2}}}{\sqrt{\lambda^2+\beta^2}}\left[e^{i\lambda x}\left(\frac{\sqrt{\lambda^2+\beta^2}+\lambda}{\beta}\right)^n + e^{-i\lambda x}\left(\frac{\sqrt{\lambda^2+\beta^2}-\lambda}{\beta}\right)^n\right] d\lambda.$$

Appendix B. Generalized gaussian quadrature

The new version of fast multipole method translates the multipole expansions to exponential ones using the plane wave expansions as discussed in Appendix A. In the numerical implementation, a very natural question arises: What are the optimal weights and nodes for approximating $K_n(\beta r)e^{in\theta}$? In this section, we discuss the term $n = 0$, and ask how to approximate (23) using

$$K_0(\beta r) \approx \sum_{k=1}^{N} w_k e^{-\sqrt{\lambda_k^2+\beta^2}x}(e^{i\lambda_k y} + e^{-i\lambda_k y}). \tag{25}$$

Note that the approximation must hold for all points (x,y) in the interaction list, and that the optimal nodes are unknown. For this non-linear optimization problem, Cheng et al. [12] constructed a generalized Gaussian quadrature scheme. In traditional Gaussian quadrature techniques, the nodes and weights of the quadratures satisfy systems of non-linear equations where polynomials are used as basis test functions. The generalized Gaussian quadrature is based on the observation that for the integral in (23), polynomial test functions can be replaced by a system of functions derived using singular value decomposition. This results in a similar system of non-linear equations which can be solved to obtain the optimal nodes. The weights are then derived using least squares fit.

This technique has been successfully applied to a fairly broad classes of functions, including the diagonal translation operators for Laplace's equation. In our algorithm, we applied this technique to generate the optimal quadratures for the plane wave expansion in (25). Numerical results show that for a six digit accuracy requirement, at most sixteen node points are required for β of all range. As β gets larger, less terms are required.

Appendix C. Diagonal translation operator

The new version of the FMM [11,20,22,26,28,33,51] are based on exponential expansions discussed in Appendix A. It diagonalizes the multipole to local translation operator $\mathscr{T}_{ML}$ in (13), and reduces the constant prefactor implied by the O(N) notation substantially, especially in three dimensions. Here, we extend this new technology to the modified Helmholtz equation in two dimensions, reducing the number of operations in applying multipole-to-local translation from $27p^2$ to $p^2 + 27p + p^2$.

First, suppose a node s centered at $S = (s_1,s_2)$ is in the interaction list of a given box B and it is associated with the truncated multipole expansion

$$\Phi = \sum_{l=-p}^{p} M_l K_l(\beta\rho)\mathrm{e}^{\mathrm{i}l\theta}, \tag{26}$$

which is to be transmitted to the local expansion of B. Assume B lies to the right (x increasing), using formulas in Theorem 9 of Appendix A and the generalized Gaussian quadratures in Appendix B, the field can equally well be described, although approximately, by the Q-term plane wave (exponential) expansion

$$\Phi \approx \sum_{q=1}^{Q} \left[M_q^1 \mathrm{e}^{-\sqrt{\lambda_q^2+\beta^2}(x-s_1)}\mathrm{e}^{\mathrm{i}\lambda_q(y-s_2)} + M_q^2 \mathrm{e}^{-\sqrt{\lambda_q^2+\beta^2}(x-s_1)}\mathrm{e}^{-\mathrm{i}\lambda_q(y-s_2)} \right]. \tag{27}$$

The advantage of this plane wave representation is that the operator which shifts the expansion center to a new location $S' = (s_1', s_2')$ is carried out using $\mathrm{O}(Q)$ multiplications rather than $\mathrm{O}(p^2)$. More precisely, the translated expansion is simply

$$\Phi \approx \sum_{q=1}^{Q} \left[N_q^1 \mathrm{e}^{-\sqrt{\lambda_q^2+\beta^2}(x-s_1')}\mathrm{e}^{\mathrm{i}\lambda_q(y-s_2')} + N_q^2 \mathrm{e}^{-\sqrt{\lambda_q^2+\beta^2}(x-s_1')}\mathrm{e}^{-\mathrm{i}\lambda_q(y-s_2')} \right], \tag{28}$$

where:

$$N_q^1 = M_q^1 \cdot \mathrm{e}^{-\sqrt{\lambda_q^2+\beta^2}(s_1'-s_1)}\mathrm{e}^{\mathrm{i}\lambda_q(s_2'-s_2)},$$
$$N_q^2 = M_q^2 \cdot \mathrm{e}^{-\sqrt{\lambda_q^2+\beta^2}(s_1'-s_1)}\mathrm{e}^{-\mathrm{i}\lambda_q(s_2'-s_2)}.$$

As the coefficients $\{M_q\}$ are mapped to $\{N_q\}$ using only Q operations and the linear mapping matrix is diagonal, this operator is called a diagonal translation operator. It remains only to control the number of terms in the exponential representation, denoted by Q above. This is discussed briefly in Appendix B, and it turns out that Q is, in fact, slightly smaller than p.

The shifted exponential expansion is then converted to a classical local expansion. This can be done using the following formula (see [1], p. 376):

$$\mathrm{e}^{z\cos\theta} = I_0(z) + 2\sum_{k=1}^{\infty} I_k(z)\cos(k\theta). \tag{29}$$

We refer the reader to [20] for a more thorough discussion of plane wave based schemes in the context of the Poisson equation.

References

[1] M. Abramowitz, I. Stegun (Eds.), Handbook of Mathematical Functions, Dover, New York, 1965.
[2] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 53 (1989) 484–512.
[3] C.L. Berman, L. Greengard, A renormalization method for the evaluation of lattice sums, J. Math. Phys. 35 (1994) 6036–6048.
[4] D. Borwein, J.M. Borwein, Analysis of certain lattice sums, J. Math. Anal. Appl. 143 (1989) 126.
[5] A. Brandt, Multi-level adaptive solutions to boundary value problems, Math. Comput. 31 (1977) 330–390.
[6] B.L. Buzbee, G.H. Golub, C.W. Nielson, On direct methods for solving Poisson's equations, SIAM J. Numer. Anal. 7 (1970) 627–656.
[7] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, New York, 1988.
[8] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM J. Sci. Stat. Comput. 9 (1988) 669–686.
[9] T. Chan, R. Glowinski, J. Périaux, O. Widlund (Eds.), Domain Decomposition Methods, Society for Industrial and Applied Mathematics, 1989.

[10] T. Chan, B. Smith, Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes, Electron. Trans. Numer. Anal. 2 (1994) 171–182.
[11] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive fast multipole algorithm in three dimensions, J. Comput. Phys. 155 (1999) 468.
[12] H. Cheng, V. Rokhlin, N. Yarvin, Nonlinear optimization, quadrature, and interpolation, SIAM J. Opt. 9 (4) (1999) 901–923.
[13] S.K. Chin, N.A. Nicorovici, R.C. McPhedran, Green's function and lattice sums for electromagnetic scattering by a square array of cylinders, Phys. Rev. E 49 (5) (1994) 4590.
[14] G. Chesshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1991) 1–64.
[15] L. Collatx, The Numerical Treatment of Differential Equations, third ed., Springer-Verlag, Berlin, 1960.
[16] T. Darden, D. York, L. Pederson, Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems, J. Chem. Phys. 98 (1993).
[17] E. Darve, The fast multipole method. I. Error analysis and asymptotic complexity, SIAM J. Numer. Anal. 38 (1) (2000) 98–128.
[18] A. Dienstfrey, F.B. Hang, J.F. Huang, Lattice sums and the two-dimensional periodic Green's function for the Helmholtz equation, Proc. Roy. Soc. Lond. A 457 (2005) 67–85, 2001.
[19] M.A. Epton, B. Dembart, Multipole translation theory for the three-dimensional Laplace and Helmholtz equations, SIAM J. Sci. Comput. 16 (1995) 865–897.
[20] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM J. Sci. Comput. 23 (3) (2001) 741–760.
[21] P. Ewald, Die Berechnung optischer und elektrostatischer Gitterpotentiale, Ann. Phys. 64 (1921) 253–287.
[22] Z. Gimbutas, L. Greengard, M. Minion, Coulomb interactions on planar structures: inverting the square root of the Laplacian, SIAM J. Sci. Comput. 22 (6) (2001) 2093–2108.
[23] L. Greengard, The Rapid Evaluation of Potential Fields in Particle Systems, MIT Press, Cambridge, MA, 1988.
[24] L. Greengard, J.Y. Lee, A direct adaptive Poisson solver of arbitrary order accuracy, J. Comput. Phys. 125 (1996) 415–424.
[25] L. Greengard, J. Huang, A new version of the fast multipole method for screened Coulomb interactions in three dimensions, J. Comput. Phys. 180 (2002) 642–658.
[26] L. Greengard, J. Huang, V. Rokhlin, S. Wandzura, Accelerating fast multipole methods for low frequency scattering, IEEE Comp. Sci. Eng. 5 (1998) 32–38.
[27] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.
[28] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace's equation in three dimensions, Acta Numer. 6 (1997) 229.
[29] R.B. Guenther, J.W. Lee, Partial Differential Equations of Mathematical Physics and Integral Equations, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[30] J. Helsing, Bounds on the shear modulus of suspensions by interface methods, J. Mech. Phys. Solids 42 (1994) 1123.
[31] E.W. Hobson, The Theory of Spherical and Ellipsoidal Harmonics, Cambridge at the University Press, Cambridge, 1931.
[32] E.N. Houstis, T.S. Papatheodorou, High-order fast elliptic equation solver, ACM Trans. Math. Software 5 (1979) 431–441.
[33] T. Hrycak, V. Rokhlin, An improved fast multipole algorithm for potential fields, SIAM J. Sci. Stat. Comput. 19 (1998) 1804–1826.
[34] J.F. Huang, Integral representations of harmonic lattice sums, J. Math. Phys. 40 (10) (1999) 5240–5246.
[35] J.D. Jackson, Classical Electrodynamics, Wiley, New York, 1975.
[36] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60–85.
[37] A.H. Juffer, E.F.F. Botta, B.A.M.v. Keulen, A.v.d. Ploeg, H.J.C. Berendsen, The electric potential of a macromolecule in a solvent: a fundamental approach, J. Comput. Phys. 97 (1991) 144.
[38] C.G. Lambert, T.A. Darden, J.A. Board, A multipole-based algorithm for efficient calculation of forces and potentials in macroscopic periodic assemblies of particles, J. Comput. Phys. 126 (1996) 274.
[39] J. Liang, S. Subramaniam, Computation of molecular electrostatics with boundary element methods, Biophys. J. 73 (1997) 1830.
[40] S. McCormick, Multigrid Methods, Society for Industrial and Applied Mathematics, 1987.
[41] M.L. Minion, Two methods for the study of vortex patch evolution on locally refined grids, Ph.D. Thesis, University of California, Berkeley, 1994.
[42] P.M. Morse, H. Feshbach, Methods of Theoretical Physics, McGraw-Hill, New York, 1953.
[43] B.R.A. Nijboer, F.W. De Wette, On the calculation of lattice sums, Physica 23 (1957) 309.
[44] J.R. Phillips, J.K. White, A precorrected-FFT method for electrostatic analysis of complicated 3-D structures, IEEE Trans. Comp. Aided Des. 16 (Oct) (1997) 1059–1072.
[45] L. Rayleigh, On the influence of obstacles arranged in a rectangular order upon the properties of the medium, Philos. Mag. 34 (1892) 481.
[46] W.B. Russell, D.A. Seville, W.R. Schowalter, Colloidal Dispersions, Cambridge University Press, Cambridge, 1991.

[47] K.E. Schmidt, M.A. Lee, Implementing the fast multipole method in three dimensions, J. Stat. Phys. 63 (1991) 1223.
[48] I. Stakgold, Boundary Value Problems of Mathematical Physics, MacMillan Company, New York, 1968.
[49] P.N. Swarztrauber, The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle, J. Comput. Phys. 15 (1974) 46–54.
[50] P. Swarztrauber, R. Sweet, Efficient Fortran subprograms for the solution of elliptic partial differential equations, NCAR Technical Note NCAR-TN/IA-109, 1975, pp. 135–137.
[51] N. Yarvin, V. Rokhlin, Generalized Gaussian quadratures and singular value decompositions of integral operators, SIAM J. Sci. Comput. 20 (1999) 699–718.