We present an adaptive fast multipole method for inverting the square root of the Laplacian in two dimensions. Solving this problem is the dominant computational cost in many applications arising in electrical engineering, geophysical fluid dynamics, and the study of thin films. It corresponds to the evaluation of the field induced by a planar distribution of charge or vorticity. Our algorithm is direct and assumes only that the source distribution is discretized using an adaptive quad-tree. The amount of work grows linearly with the number of mesh points.

Coulomb interactions on planar structures: inverting the square root of the Laplacian

Zydrunas Gimbutas, Leslie Greengard, and Michael Minion
Research Report YALEU/DCS/RR-1185
July 27, 1999

The first author was supported in part by AFOSR under Grant F49620-97-1-0011; the second author was supported by the Applied Mathematical Sciences Program of the U.S. Department of Energy under Contract DEFGO288ER25053; the third author was supported by the Applied Mathematical Sciences Program of the U.S. Department of Energy under Contract DEFGO288ER25053.
Approved for public release: distribution is unlimited.
Keywords: *Fast Multipole Method, N-body Problems, Potential Theory, Fast Algorithms.*

1 Introduction

In this paper, we present a fast, adaptive, numerical method for solving the pseudodifferential equation

$$(-\Delta)^{1/2}\psi = \omega \tag{1}$$

in the plane, where Δ denotes the Laplacian operator. This equation appears in a variety of different mathematical models for problems in surface physics, some of which are described below. We assume that $\omega(x) \in L^2(\mathbf{R}^2)$, with Fourier transform

$$\hat{\omega}(\mathbf{k}) = \int_{\mathbf{R}^2} \omega(\mathbf{x}) e^{-2\pi i\,\mathbf{k}\cdot\mathbf{x}} d\mathbf{x}.$$

The symbol of $-\Delta$ is $1/(2\pi|\mathbf{k}|)^2$, so that the solution to (1) can be written as

$$\psi(\mathbf{x}) = \int_{\mathbf{R}^2} \frac{\widehat{\omega(\mathbf{k})}}{2\pi|\mathbf{k}|} e^{2\pi i\,\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}. \tag{2}$$

Alternatively, we can seek a Green's function for this operator, by solving the equation

$$(-\Delta)^{1/2}G(\mathbf{x}) = \delta(\mathbf{x}).$$

A straightforward calculation yields

$$G(\mathbf{x}) = \int_{\mathbf{R}^2} \frac{1}{2\pi|\mathbf{k}|} e^{2\pi i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k} = \frac{1}{|\mathbf{x}|}, \tag{3}$$

from which the inversion formula

$$\psi(\mathbf{x}) = \int_{\mathbf{R}^2} \frac{\omega(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|} d\mathbf{y} \tag{4}$$

follows.

Since the function $G(\mathbf{x})$ obtained in (3) is the Green's function for the three-dimensional Laplacian, we can think of equations (1) and (4) as describing a three-dimensional Poisson equation for which $\omega(\mathbf{x})$ is a singular density lying entirely on the plane $z = 0$. Applications involving such surface interactions include the study of planar circuits in electrical engineering, a variety of problems in thin films [5, 19, 20], and certain problems in tomography [7]. In quasigeostrophic fluid dynamic models [17], one encounters an equation analogous to the incompressible Navier-Stokes equations which take the form

$$\begin{aligned} \omega_t + (U\cdot\nabla)\omega &= \nu\Delta\omega \\ U &= \nabla^{\perp}\psi \\ (-\Delta^{1/2})(-\psi) &= \omega, \end{aligned} \tag{5}$$

where U is a velocity field, ω is a vorticity-like variable, and ψ is a stream function. In the past three years, these equations have seen a new wave of interest stemming, in part, from observations made in [6] which draw interesting analogies between the quasigeostrophic equations and the three-dimensional incompressible Euler equations. Another interesting application is a two-fluid model system which supports internal waves and which can be applied to a variety of geophysical model systems [3]. One specific example is a vertically stratified fluid in which one layer is much thinner than the other, and in which the square root of the Laplacian plays a fundamental role.

All of the preceding application areas would benefit from adaptive numerical simulation tools in order to resolve complex solution features. Most existing numerical methods for solving (1), however, rely on the spectral form of the solution given in (2) and are implemented via the fast Fourier transform. This approach precludes the use of adaptivity and constrains the computational domain to be periodic.

Direct evaluation of the convolution integral (4) involves interactions between all pairs of grid points. Hence, with N points in the discretization, the work involved would require $O(N^2)$ operations. In this paper, we describe two special purpose fast multipole methods (FMMs) for the computation of the integral transform (4) for which the work required grows linearly with the number of grids points. This approach allows for adaptive mesh refinement and the imposition of either free-space or periodic boundary conditions. Related methods have been developed in [2, 14, 16].

2 Data structures and the Fast Multipole Method

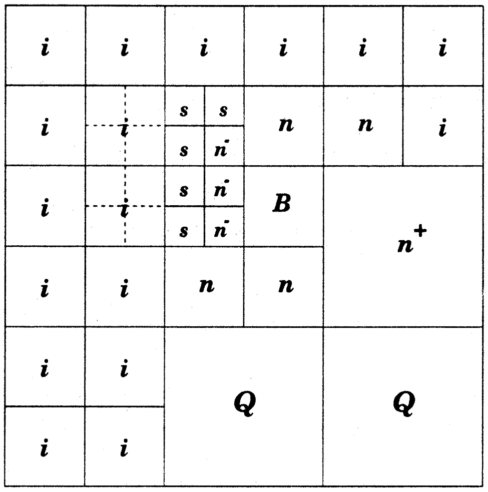
We assume that the source distribution ω in (1) is supported inside the unit square D, centered at the origin, on which is superimposed a hierarchy of refinements (a quad-tree). Grid level 0 is defined to be D itself, and grid level $l+1$ is obtained recursively by subdividing each square at level l into four equal parts. Using standard terminology, if d 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. In order to allow for adaptivity, we do not use the same number of levels in all regions of D. We do, however, assume that the quad-tree satisfies one fairly standard restriction, namely that two leaf nodes which share a boundary point must be no more than one refinement level apart (Fig. 1).

The leaf nodes on which the source distribution is given will be denote by D_i. Thus, $D = \cup_{i=1}^{M} D_i$ and we rewrite (4) in the form

$$\psi(\mathbf{x}) = -\sum_{i=1}^{M} \int_{D_i} \frac{\omega(\mathbf{y})}{|\mathbf{x}-\mathbf{y}|}\, d\mathbf{y}. \tag{6}$$

Definition 2.1. *The* colleagues *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.) The*

Figure 1: For the childless node $\mathbf{B}$, colleagues are labelled by $\mathbf{n}$, coarse neighbors are labelled $\mathbf{n}^+$, and fine neighbors are labelled $\mathbf{n}^-$. The interaction list for B consists of the boxes marked by $\mathbf{i}$ as well as those marked by $\mathbf{Q}$. The boxes marked by s are children of B's colleagues which are separated from B, so they are not fine neighbors. They constitute the *s-list* for B [see Def. 2.1].



coarse neighbors *of B are leaf nodes at the level of B's parent which share a boundary point with B. The* fine neighbors *of B are leaf nodes one level finer than B which share a boundary point with B. Together, the union of the colleagues, coarse neighbors and fine neighbors of B will be referred to as B's* neighbors. *The* s-list *of a box B consists of those children of B's colleagues which are not fine neighbors of B.*

The interaction region *for B consists of the area covered by the neighbors of B's parent, excluding the area covered by B's colleagues and coarse neighbors. The* interaction list *for B consists of those squares in the interaction region which are at the same refinement level, as well as leaf nodes in the interaction region which are at coarser levels. When the distinction is important, the squares at the same refinement level will be referred to as the* standard interaction list, *while the squares at coarser levels will be referred to as the* coarse interaction list.

In our fast multipole method, following [4, 11, 12], terms in the convolution integral (6) from neighbor leaf nodes are computed directly. More distant interactions are accounted for on coarser levels, through the use of a hierarchy of far-field and local multipole expansions. We consider the local interactions first.

2.1 Local interactions

We assume that we are given ω on a cell-centered 4×4 grid for each leaf node B. We can, therefore, take these 16 data points and construct a fourth-order polynomial approximation to ω of the form

$$\omega_B(x,y) \approx \sum_{j=1}^{10} c_B(j)\, b_j(x - x_B, y - y_B),$$

where (x_B, y_B) denotes the center of B. The basis functions $b_j(x,y)$ are given by

$$1,\ x,\ y,\ x^2,\ xy,\ y^2,\ x^3,\ x^2y,\ xy^2,\ y^3,$$

for $j = 1, \ldots, 10$, respectively. If we let $\vec{\omega_B} \in \mathbf{R}^{16}$ denote the discrete function values in standard ordering, then the calculation of the coefficient vector $\vec{c}_B$ is clearly overdetermined. We obtain it through a least squares fit based on the singular value decomposition. The pseudoinverse matrix $\mathcal{P} \in \mathbf{R}^{10\times16}$, such that

$$\vec{c}_B = \mathcal{P}\,\vec{\omega_B},$$

can be precomputed and stored.

Consider now a target point Q, which lies in a neighbor of B (Fig. 2). Then, the field induced by ω_B is approximated by

$$\psi_B(Q) = \sum_{j=0}^{10} c_B(j) w(Q,j), \tag{7}$$

where

$$w(Q,j) = \int_B \frac{b_j(x - x_B, y - y_B)}{|Q - (x,y)|}\, dxdy. \tag{8}$$

Since the target points Q are regularly spaced in each neighboring square, we can precompute the weights (8) for each of the sixteen possible locations at each of 9 possible colleagues, 12 possible fine neighbors, and 8 possible coarse neighbors. To be more precise, we can precompute the weights assuming that B is the unit square $[-0.5, 0.5]^2$, because of the following straightforward lemma.

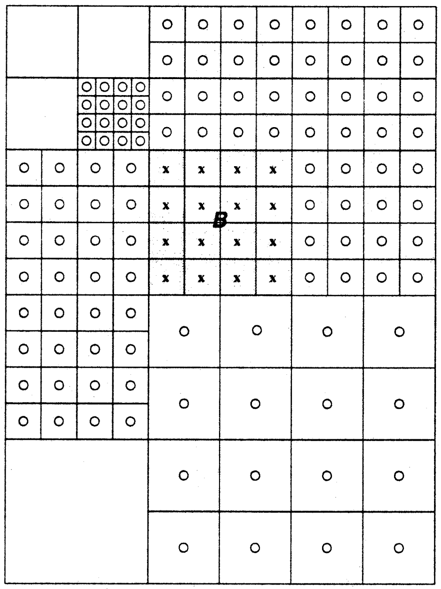
Lemma 2.1. *Let B be a leaf node at level l and let Q denote a target point in one of B's neighbors. Let Q^* denote the scaled target point for the unit cell centered at the origin*

$$Q^* = 2^l \cdot (Q - (x_B, y_B)),$$

and let

$$w^*(Q^*,j) = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \frac{b_j(x,y)}{|Q^* - (x,y)|}\, dxdy. \tag{9}$$

Figure 2: The source distribution ω is given on a cell-centered 4×4 grid in the central square B. The field induced by the distribution on B's neighbors can be tabulated and stored. In the adaptive grid, neighbors can be at the same refinement level as B, one coarser, or one finer.



Then the integral $w(Q,j)$ defined in (8) is given by

$$w(Q,j) = w^*(Q^*,j) \cdot 2^{-[d+1]l},$$

where d is the degree of the polynomial basis function b_j.

Thus, we need only obtain weights for a box of unit area. Elementary counting arguments show that the storage required for this precomputation is

$16 \cdot 10 \cdot 9$ real numbers	for colleagues,
$16 \cdot 10 \cdot 12$ real numbers	for fine neighbors,
$16 \cdot 10 \cdot 8$ real numbers	for coarse neighbors,

for a total of 4640 real numbers.

2.2 Far-field interactions

We turn now to the calculation of far-field interactions, which are computed by means of multipole expansions. We refer the reader to [9, 15, 18] for more detailed discussions of potential theory. Our starting point is the usual multipole expansion for a charge distribution, which we state formally as a theorem.

Theorem 2.1. (Multipole Expansion). *Let $\sigma(T)$ be a charge distribution contained within S, a sphere of radius a centered at the origin, and let $Q = (r, \theta, \phi) \in \mathbb{R}^3$ with $r > a$. Then the field at Q induced by the charge distribution*

$$\Phi(Q) = \int_S \frac{\sigma(T)}{|T - Q|}\, dT$$

can be described by the multipole expansion

$$\Phi(Q) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{M_n^m}{r^{n+1}} \cdot P_n^m(\cos\theta) e^{im\phi} \ , \tag{10}$$

where P_n^m denotes the standard associated Legendre function and

$$M_n^m = \frac{(n - |m|)!}{(n + |m|)!} \int_S \sigma(T) \cdot \rho^n \cdot P_n^{|m|}(\cos\alpha) e^{im\beta} dT. \tag{11}$$

In the preceding expression, (ρ, α, β) are the spherical coordinates of T. Furthermore, for any $p \geq 1$,

$$\left| \Phi(Q) - \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{M_n^m}{r^{n+1}} \cdot P_n^m(\cos\theta) e^{im\phi} \right| \leq \left(\frac{\int_S |\sigma(T)|\, dT}{r - a} \right) \left(\frac{a}{r} \right)^{p+1} . \tag{12}$$

In the setting of the present paper, the sources and targets are restricted to the plane $z = 0$, so that in the preceding formulas, $Q = (r, \pi/2, \phi)$ and $T = (\rho, \pi/2, \beta)$ when expressed in spherical coordinates. Thus, for a charge distribution σ supported in a square D centered at the origin and a target point Q lying in the interaction list for B, $\Phi(Q)$ can be expressed as a multipole expansion of the form

$$\Phi(Q) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \cdot \frac{e^{im\phi}}{r^{n+1}}, \tag{13}$$

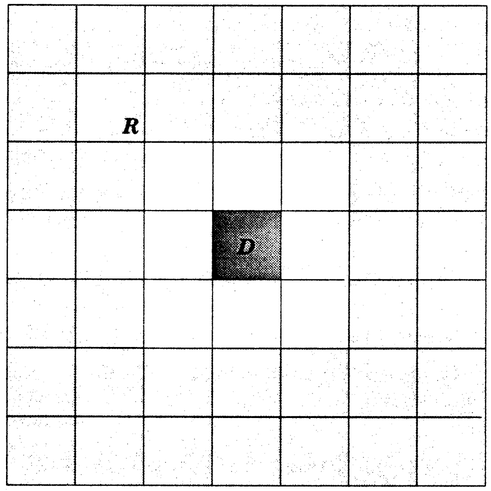
with

$$M_n^m = \frac{(n - |m|)!}{(n + |m|)!} [P_n^m(0)]^2 \int_B \sigma(T) \rho^n e^{im\beta}\, dT, \tag{14}$$

where (ρ, β) are the polar coordinates of T (Fig. 3). The error estimate (12) takes the special form

$$\left| \Phi(Q) - \sum_{n=0}^{p} \sum_{m=-n}^{n} M_n^m \cdot \frac{e^{im\phi}}{r^{n+1}} \right| \leq \left(\frac{\int_D |\sigma(T)|\, dT}{2a} \right) \left(\frac{\sqrt{2}}{3} \right)^{p+1} \tag{15}$$

Figure 3: For the box D, the interaction region is contained within the outer shaded region.



where a^2 is the area of D.

Within the FMM, it is convenient to be able to describe the field within a region due to sources which are far away. For this, suppose that Q lies in D and that

$$\Psi(Q) = \int_S \frac{\sigma(P)}{|P-Q|}\, dP,$$

where the region S lies outside the nine colleagues of D (Fig. 3). Then

$$\Psi(Q) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} L_n^m \cdot r^n\, e^{im\phi}, \tag{16}$$

with

$$L_n^m = \frac{(n-|m|)!}{(n+|m|)!}[P_n^m(0)]^2 \int_S \sigma(P) \frac{e^{im\theta}}{\rho^{n+1}}\, dP, \tag{17}$$

where (ρ, θ) are the polar coordinates of P. Furthermore,

$$\left| \Psi(Q) - \sum_{n=0}^{p} \sum_{m=-n}^{n} L_n^m \cdot r^n\, e^{im\phi} \right| \leq \left(\frac{\|\sigma(P)\|_{L_1}}{a} \right) \left(\frac{\sqrt{2}}{3} \right)^{p+1} . \tag{18}$$

The FMM relies on the ability to manipulate multipole and local expansions for every box in the tree hierarchy. We omit the technical details, and refer the reader the original papers [4, 9, 11, 12].

Definition 2.2. *We denote by $S_{l,k}$ the* k*th square at refinement level* l.

We denote by $\Phi_{l,k}$ the multipole expansion describing the far field due to the source distribution supported inside $S_{l,k}$.

We denote by $\Psi_{l,k}$ the local expansion describing the field due to the source distribution outside the neighbors of $S_{l,k}$.

We denote by $\tilde{\Psi}_{l,k}$ the local expansion describing the field due to the source distribution outside the neighbors of the parent *of $S_{l,k}$.*

Remark 2.1. Let $S_{l,k}$ be a square in the quad-tree hierarchy and let $S_{l',k'}$ be a square in its interaction list. Then there is a linear operator $\mathcal{T}_{MM}$ for which

$$\Phi_{l,k} = \mathcal{T}_{MM}[\Phi(C_1), \Phi(C_2), \Phi(C_3), \Phi(C_4)], \tag{19}$$

where $\Phi(C_j)$ denotes the multipole expansion for the j*th* child of $S_{l,k}$. In other words, we can merge the expansions for four children into a single expansion for the parent. Similarly, there is a linear operator $\mathcal{T}_{LL}$ for which

$$[\tilde{\Psi}(C_1), \tilde{\Psi}(C_2), \tilde{\Psi}(C_3), \tilde{\Psi}(C_4)] = \mathcal{T}_{LL}\,\Psi_{l,k}, \tag{20}$$

where C_j denotes the j*th* child of $S_{l,k}$. In other words, we can shift the local expansion Ψ for a box to the corresponding expansion $\tilde{\Psi}$ for each of its children. Finally, there is a linear operator $\mathcal{T}_{ML}$ for which the field in $S_{l,k}$ due to the source distribution in $S_{l',k'}$ is described by $\Psi = T_{ML}\Phi_{l',k'}$. It is easy to verify that

$$\Psi_{l,k} = \tilde{\Psi}_{l,k} + \sum_{i \in \mathcal{IL}} T_{ML}\Phi_i, \tag{21}$$

where $\mathcal{IL}$ denotes the interaction list for square $S_{l,k}$.

Remark 2.2. One slight complication in the adaptive algorithm concerns the interaction between boxes of different sizes. Referring to Fig. 1, we need to account for the influence of a childless square B on each box marked s and vice-versa. (This interaction clearly doesn't arise if B undergoes further refinement.) For the box marked s, its multipole expansion is rapidly convergent at each of the sixteen target points in B. Thus, its influence can be computed by direct evaluation of the truncated series. For the reverse, however, note that B's multipole expansion is not so rapidly convergent. In this case, we directly compute the coefficients of the local expansion in s from the formula (17). A more precise statement than (21) is

$$\Psi_{l,k} = \tilde{\Psi}_{l,k} + \sum_{i \in SIL} T_{ML}\Phi_i, + \sum_{i \in CIL} L_{direct}(\vec{\omega}_i), \tag{22}$$

where SIL denotes the standard interaction list and CIL denotes the coarse interaction list. The operator L_{direct}, which maps the coefficients of the polynomial approximation of the density in the coarse box onto the p^2 coefficients of the local expansion can be precomputed and stored.

The bulk of the work in the FMM consists in applying the operators $\mathcal{T}_{MM}, \mathcal{T}_{ML}, \mathcal{T}_{LL}$ in a systematic fashion. Unfortunately, these operators are dense. Using multipole and local expansions truncated after p^2 terms, the naive cost of application is proportional to p^4. Modern versions of the FMM have reduced this cost to $O(p^3)$ or $O(p^2)$ [12].

Algorithm

Initialization

Comment [We assume we are given a square domain $D = S_{0,0}$, on which is superimposed an adaptive hierarchical quad-tree structure. We let M be the number of leaf nodes and denote them by D_i, $i = 1, \ldots, M$. The number of grid points is, therefore, $N = 16M$. We let p denote the order of the multipole expansion, ($p \approx \log_2 \epsilon$, where ϵ is the desired accuracy). We let l_{max} denote the maximum refinement level.]

Step I: Multipole Sweep

Upward pass

```
for $l = levmax, \ldots, 0$
    for all boxes $j$ on level $l$
        if $j$ is childless then
            form the multipole expansion $\Phi_{l,j}$ from (14)
        else
            form the multipole expansion $\Phi_{l,j}$ by merging the expansions of
            its children using the operator $\mathcal{T}_{MM}$ (see (19))
    end
end
```

Downward pass

```
Initialize the local expansion $\Psi_{0,0} = 0$.
for $l = 1, \ldots, levmax$
    for all squares $j$ on level $l$
        Compute $\tilde{\Psi}_{l,j}$ by shifting its parent's $\Psi$ expansion using the operator $\mathcal{T}_{LL}$
        Compute $\Psi_{l,j}$ by adding in the contributions from all squares in $j$'s
```

```
                interaction list according to (22).
            if $j$ is childless then
                for all boxes $k$ in the s-list of $j$:
                    evaluate the multipole expansion $\Phi_k$ at each
                    target in square $j$.
                end
                Evaluate the local expansion $\Psi_{l,j}$ at each
                target in square $j$.
            endif
        end
    end
```

Cost [The upward pass requires approximately Mp^2 work, where M is the number of leaf nodes. The downward pass requires approximately $12Mp^2 + 4Mp^3$ work (see Remark 3.3 below).]

Step II: Local interactions

Comment [At this point, for each leaf node D_i, we have computed the influence of the source distribution ω over all leaf nodes D_j outside the neighbors of D_i.]

```
    do $i = 1, ..., M$
        For each target point in $D_i$, evaluate the influence of each
            neighbor according to (7) using the precomputed
            tables of coefficients (9).
    end
```

Cost [The maximum number of neighbors a square can have is thirteen (twelve fine neighbors and itself). Thus the local work is bounded by is $12 \cdot 10 \cdot N$ operations.]

Remark 2.3. It is somewhat difficult to determine the cost of Step I precisely, since it depends on the actual structure of the adaptive quad-tree. A reasonable estimate for the work is

$$N(120 + \frac{12}{16}p^2 + \frac{4}{16}p^3).$$

2.3 The generalized FMM

One drawback of the preceding scheme is that it relies on spherical harmonic expansions. These are particularly efficient tools for fully three-dimensional calculations, but they make no use of the fact that we are restricted to a two-dimensional domain. In fact, a two-dimensional Taylor series would have served equally well. In the last few years, methods related to the FMM have been developed which are based only on the fact that the far-field is smooth [1, 2, 8, 14]. The paper [8] presents a generalized FMM based on "compressing" operators with the singular value decomposition (SVD). To understand this approach, let us reconsider the situation depicted in Fig. 3.

Definition 2.3. *Let $\mathcal{C}$ denote the space of real analytic charge distributions defined on a square D and let $\mathcal{P}$ denote the space of real analytic functions defined on the outer shaded region R depicted in Fig. 3. We define the operator $\mathcal{O} : \mathcal{C} \to \mathcal{P}$ according to $\mathcal{O}(\sigma) = \Phi_\sigma$, where*

$$\Phi_\sigma(Q) = \int_D \frac{\sigma(P)}{|P-Q|}\, dP.$$

We also define the dual operator $\mathcal{S} : \mathcal{P} \to \mathcal{C}$ according to $\mathcal{S}(\sigma) = \Psi_\sigma$, where σ is a charge distribution defined on R, $P \in D$, and

$$\Psi_\sigma(P) = \int_R \frac{\sigma(Q)}{|P-Q|}\, dQ.$$

Remark 2.4. *The spaces $\mathcal{C}$ and $\mathcal{P}$ are infinite-dimensional, but both $\sigma \in \mathcal{C}$ and $\Phi_\sigma \in \mathcal{P}$ can be approximated arbitrarily well by a finite tensor-product Legendre series*

$$\sigma(x,y) \approx \sum_{n=0}^{\mathcal{N}} \sum_{m=0}^{\mathcal{N}} \alpha_n^m P_n(x) P_m(y), \tag{23}$$

$$\Phi_\sigma(\xi,\eta) \approx \sum_{n=0}^{\mathcal{N}} \sum_{m=0}^{\mathcal{N}} \beta_n^m P_n(\xi) P_m(\eta), \tag{24}$$

where P_n is the Legendre polynomial of degree n scaled to the dimensions of D. More precisely, we suppose that there is an expansion of the form (24) for each of the forty squares comprising the region R.

For $\mathcal{N}$ sufficiently large, we can approximate the operator $\mathcal{O}$ by the finite-dimensional matrix

$$\mathcal{O}_\mathcal{N} : I\!R^{\mathcal{N}^2} \to I\!R^{40\mathcal{N}^2},$$

mapping the coefficients $\{\alpha_n^m\}$ in the expansion of the charge density σ to the 40 sets of coefficients $\{\beta_n^m\}$ in the expansions of Φ_σ. If we let the singular value decomposition of $\mathcal{O}_\mathcal{N}$ be given by

$$\mathcal{O}_\mathcal{N} = U_\mathcal{N} \Sigma_\mathcal{N} V_\mathcal{N}^T,$$

then $\mathcal{O}_{\mathcal{N}}$ can be *compressed* by retaining only the first k terms in the decomposition:

$$\mathcal{O}_{\mathcal{N}} \approx U_{\mathcal{N}}(k)\,\Sigma_{\mathcal{N}}(k)\,V_{\mathcal{N}}(k)^T, \tag{25}$$

where $V_{\mathcal{N}}(k) \in I\!R^{\mathcal{N}^2 \times k}$, where $U_{\mathcal{N}}(k) \in I\!R^{40\mathcal{N}^2 \times k}$ and $\Sigma_{\mathcal{N}}(k) \in I\!R^{k \times k}$. This leaves one open question: for a given precision ϵ, how large must $\mathcal{N}$ and k be so that

$$\|\mathcal{O}(\sigma) - \; U_{\mathcal{N}}(k)\,\Sigma_{\mathcal{N}}(k)\,V_{\mathcal{N}}(k)^T\,P_{\mathcal{N}}\sigma\| < \epsilon, \tag{26}$$

where $P_{\mathcal{N}}$ is the operator projecting σ onto its truncated Legendre series? Similarly, $\mathcal{S}$ has an approximate singular value decomposition

$$\mathcal{S}_{\mathcal{N}} = W_{\mathcal{N}}(k)\Omega_{\mathcal{N}}(k)Y_{\mathcal{N}}(k)^T,$$

where $Y_{\mathcal{N}}(k) \in I\!R^{40\mathcal{N}^2 \times k}$, where $W_{\mathcal{N}}(k) \in I\!R^{\mathcal{N}^2 \times k}$ and $\Omega_{\mathcal{N}}(k) \in I\!R^{k \times k}$. It remains also to determine $\mathcal{N}$ and k so that

$$\|\mathcal{S}(\sigma) - \; W_{\mathcal{N}}(k)\,\Omega_{\mathcal{N}}(k)\,Y_{\mathcal{N}}(k)^T\,P_{\mathcal{N}}\sigma\| < \epsilon. \tag{27}$$

This is a rather complicated matter to handle analytically [8, 13], but straightforward to determine computationally. One can simply increase $\mathcal{N}$ and k until the desired level of accuracy is achieved, and we summarize the results in Table 1. The generalized FMM then proceeds as above with the following changes.

1. $\Phi_{l,j}$ is used to denote the projection of the charge density in box j at level l onto the first k right-singular vectors $V_{\mathcal{N}}(k)$ of $\mathcal{O}$ scaled to that level. $\Phi_{l,j}$ is stored as a k-vector, which we refer to as the "outgoing" coefficients. The right-singular vectors correspond, in some sense, to the multipole terms $e^{im\phi}/r^{n+1}$.

2. $\Psi_{l,j}$ and $\tilde{\Psi}_{l,j}$ are defined as before, except that we describe the field due to distant sources in terms of its projections onto the left singular vectors $W_{\mathcal{N}}(k)$ of $\mathcal{S}$. These fields are stored as k-vectors of "incoming" coefficients.

3. In Step I: *Upward Pass*, if box j is childless, we compute $\Phi_{l,j} = V_{\mathcal{N}}(k)^T\sigma$.
   Since σ is described by ten polynomial coefficients, this requires $10k$ operations; the inner product of each of the k singular vectors with each of the ten basis functions can be precomputed and stored.

4. In Step I: *Upward Pass*, if box j has children, we project the "outgoing coefficients" of the four children onto the first k right singular vectors at j's level.
   This merging operator requires k^2 operations per square, since each matrix entry in the transformation can be precomputed and stored.

Table 1: Degree of Legendre expansions $\mathcal{N}$ and number of terms in the singular value decomposition k required to approximate the operators $\mathcal{O}$ and $\mathcal{S}$ to the indicated precision.

ϵ	$k(\mathcal{O})$	$\mathcal{N}(\mathcal{O})$	$k(\mathcal{S})$	$\mathcal{N}(\mathcal{S})$
10^{-3}	9	4	9	4
10^{-6}	36	8	36	8
10^{-8}	49	10	49	10
10^{-11}	144	14	144	14

5. In Step I: *Downward Pass*, where we had made use of the operator $\mathcal{T}_{LL}$, we now project the "incoming" field of the parent's Ψ expansion to form $\tilde{\Psi}_{l,j}$ using left singular vectors.
   This requires k^2 operations, since each matrix entry in the transformation can be precomputed and stored.

6. In Step I: *Downward Pass*, we compute $\Psi_{l,j}$ by analogy with (22). We need to convert the "outgoing" coefficients for a box in the interaction list of square j to the corresponding "incoming" coefficients. Each entry in the conversion matrix is rather complicated, coupling the right singular vectors of $\mathcal{O}$ to the left singular vectors of $\mathcal{S}$. This matrix, however, can be precomputed and stored, so that the total cost is at most $27k^2$ operations per square.

7. In Step I: *Downward Pass*, we also need to evaluate the "local expansion." This information is now encoded in the left singular vector coefficients and a naive method would require access of each left singular vector (of dimension $\mathcal{N}^2$).
   Fortunately, we can precompute the influence of each singular vector at each of the sixteen target points, so that only $16k$ operations are required at this stage.

Remark 2.5. A reasonable estimate for the total work of the generalized FMM is

$$N(120 + \frac{27}{16}k^2 + 2k).$$

The first term corresponds to the local work, the second term to the expansion shifting work, and the third term to the work involved in forming and evaluating expansions at leaf nodes.

2.4 Periodic Boundary Conditions

The inversion formula (4) and the fast algorithm described above assume that the right-hand side ω is supported within a unit square. In certain applications, however, one would like to consider ω to be periodically extended to cover the entire $x-y$ plane. This requires a modest modification of the fast multipole method, following the ideas presented in [11]. At the end of the upward pass of the algorithm, we have a net multipole expansion

$$\Phi(Q) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} M_n^m \cdot \frac{e^{im\phi}}{r^{n+1}}, \tag{28}$$

for the whole unit square. This is also the expansion for each periodic image of the square (with respect to its own center). If we imagine that D represents the unit square in Fig. 3, then all such images except for the nearest (unshaded) neighbors are separated from D. Thus, the totality of the field they induce inside D is accurately representable as a single local expansion of the form

$$\Psi(Q) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} L_n^m \cdot r^n \, e^{im\phi} \tag{29}$$

It remains only to obtain the operator mapping the coefficients $\{M_n^m\}$ to the coefficients $\{L_n^m\}$. We refer the reader to [10] for a discussion of this operator, which is based on the precomputation of certain lattice sums. The local expansion which describes the field due to all squares outside the neighbors of D, can be denoted by $\Psi_{0,0}$ within the context of the FMM described above. In the rest of the algorithm, only two slight modifications are required; the interaction list and the local computations must be adjusted for boxes near the boundary to account for periodic images. This involves no significant increase in the amount of work.

3 Numerical Results

The algorithms described above have been implemented using a combination of Fortran 77 and C. All of the timings listed below correspond to calculations performed on an UltraSparc-I/167 with 128Mb RAM.

Example 1. In our first experiment, we consider the discrete N-body problem

$$\phi(\mathbf{x}_j) = \sum_{\substack{i=1 \\ i \neq j}}^{N} \frac{q_i}{\|\mathbf{x}_i - \mathbf{x}_j\|}.$$

with source locations $\{\mathbf{x}_i\}$ randomly but uniformly distributed in the unit square and source strengths randomly distributed in $[-1,1]$. We compare the performance of the two fast multipole methods at 3, 6, and 12-digit accuracy (Tables 2, 3, and 4).

Remark 3.1. Subsequent calculations will rely on the generalized FMM, since our numerical experiments show it to be three to four times faster than the spherical harmonic-based code.

Table 2: Timing results for the two fast multipole methods for 3-digit accuracy. The time for the generalized FMM using 9 singular functions is denoted by T_{alg}, while the L_2 error of the result is denoted by E_{alg}. The time and L_2 error for the classical FMM using 8th-order spherical harmonics are denoted by T_{sh} and E_{sh}, respectively. The time for direct calculation is denoted by T_{dir}.

N	T_{alg}	T_{sh}	T_{dir}	$E_2(alg)$	$E_2(sh)$
200	0.01	–	0.02	$.11 \cdot 10^{-3}$	$.10 \cdot 10^{-3}$
400	0.02	0.10	0.08	$.14 \cdot 10^{-3}$	$.93 \cdot 10^{-4}$
800	0.08	0.22	0.34	$.19 \cdot 10^{-3}$	$.11 \cdot 10^{-3}$
1600	0.12	0.47	1.40	$.20 \cdot 10^{-3}$	$.19 \cdot 10^{-3}$
3200	0.28	1.13	5.77	$.26 \cdot 10^{-3}$	$.21 \cdot 10^{-3}$
6400	0.51	2.14	22.98	$.28 \cdot 10^{-3}$	$.26 \cdot 10^{-3}$

Table 3: Timing results for the two fast multipole methods for 6-digit accuracy. The generalized FMM uses 36 singular functions, while the classical FMM uses 18th-order spherical harmonics. The columns are defined as in Table 2.

N	T_{alg}	T_{sh}	T_{dir}	$E_2(alg)$	$E_2(sh)$
200	0.03	–	0.02	$.77 \cdot 10^{-7}$	–
400	0.04	–	0.08	$.60 \cdot 10^{-7}$	–
800	0.19	0.67	0.34	$.72 \cdot 10^{-7}$	$.25 \cdot 10^{-7}$
1600	0.32	1.42	1.50	$.79 \cdot 10^{-7}$	$.24 \cdot 10^{-7}$
3200	1.10	3.21	6.04	$.15 \cdot 10^{-6}$	$.30 \cdot 10^{-7}$
6400	1.72	6.63	24.21	$.15 \cdot 10^{-6}$	$.33 \cdot 10^{-7}$

Example 2. We consider a smooth distribution of charge:

$$f(x,y) = x\ \exp(-500x^2).$$

In Table 5, we show the results of inverting the square root of the Laplacian on a uniform $N_1 \times N_1$ grid using the generalized FMM with 24 singular functions, the generalized FMM

Table 4: Timing results for the two fast multipole methods for 12-digit accuracy. The generalized FMM uses 144 singular functions, while the spherical harmonic FMM uses 32nd-order spherical harmonics. The columns are defined as in Table 2.

N	T_{alg}	T_{sh}	T_{dir}	$E_2(alg)$	$E_2(sh)$
1600	0.85	4.32	1.41	$.81 \cdot 10^{-12}$	$.37 \cdot 10^{-12}$
3200	3.36	6.43	5.69	$.98 \cdot 10^{-12}$	$.49 \cdot 10^{-12}$
6400	5.30	14.61	22.81	$.78 \cdot 10^{-12}$	$.46 \cdot 10^{-12}$

Table 5: Timing results for the uniform mesh of Example 2. The first column shows the value of N_1 defining the grid. T_{24} denotes the time required by the FMM using 24 singular functions, and E_{24} denotes the L_2 error incurred by the method. T_{36} and E_{36} are similarly defined. T_{FFT} denotes the time required by the FFT, and E_{FFT} denotes the corresponding error.

N_1	$T_{FMM(24)}$	$E_{FMM(24)}$	$T_{FMM(36)}$	$E_{FMM(36)}$	T_{FFT}	E_{FFT}
32	0.034	$.59 \cdot 10^{-1}$	0.057	$.59 \cdot 10^{-1}$	0.002	$.41 \cdot 10^{-3}$
64	0.132	$.21 \cdot 10^{-2}$	0.232	$.21 \cdot 10^{-2}$	0.006	$.17 \cdot 10^{-10}$
128	0.548	$.12 \cdot 10^{-3}$	0.889	$.12 \cdot 10^{-3}$	0.025	$.28 \cdot 10^{-13}$
256	1.912	$.12 \cdot 10^{-4}$	3.599	$.10 \cdot 10^{-4}$	0.133	$.28 \cdot 10^{-13}$
512	7.527	$.73 \cdot 10^{-5}$	14.240	$.10 \cdot 10^{-5}$	0.704	$.28 \cdot 10^{-13}$

with 36 singular functions, and the FFT. To compare the performance of the FMM with the FTT, we also compute the number of points processed per second by both schemes and a ratio of their timings (Table 6).

The following observations can be made from the data:

1. The generalized FMMs converge (more or less) as expected for a fourth-order accurate method. The 24 singular function and 36 singular function versions yield the same accuracy until $N_1 = 512$, at which point the 24 singular function FMM is dominated by the FMM precision rather than the error in polynomial approximation. The FFT is, of course, rapidly convergent for this right-hand side, since it is effectively smooth and periodic.

2. The timings for the FMMs grow linearly with the number of unknowns, while the timings for the FFT grow like $N_1^2 \log N_1$. By the time there are 250,000 unknowns, the 5-digit FMM (24 singular functions) is about 10 times as costly as the FFT. The 7-digit FMM (36 singular functions) is about 20 times as costly.

Table 6: Performance measures for the generalized FMM. The first column shows the value of N_1 defining the grid. P_{24} and P_{36} denote the number of grid points processed per second by the generalized FMM using 24 and 36 singular functions, respectively. P_{FFT} denotes the number of grid points processed per second by the FFT.

N_1	P_{24}	P_{36}	P_{FFT}	P_{24}/P_{FFT}	P_{36}/P_{FFT}
32	30118	17965	512000	17.0	28.5
64	31030	17655	682667	22.0	38.7
128	29898	18430	655360	21.9	35.6
256	34276	18210	492752	14.4	27.1
512	34827	18409	372364	10.7	20.2

Example 3. We next consider a more complex distribution of charge

$$f(x,y) = g(x-0.25, y-0.25) + g(x+.15, y-.15) + g(x-.05, y+.25),$$

where

$$g(x,y) = xy\,\exp(-2000x^2)\exp(-2000y^2).$$

There are three sharply peaked contributions to the total charge and the right-hand side is discretized adaptively. Our refinement strategy is straightforward. Let B be a leaf node with 16 grid points, as discussed in Section 2.1 and let $f_B(x,y)$ denote the fourth-order polynomial used to approximate the right-hand side on B. We then evaluate $f_B(x,y)$ on an 8×8 grid covering B and compute the discrete L_2 error $E_2 = \|f(x,y) - f_B(x,y)\|_2$ over these target points. If $E_2 > tol$, the leaf node B is subdivided. Our results are summarized in Table 7.

Example 4. In our last example, we consider a discontinuous right-hand side

$$f(x,y) = \begin{cases} 3, & |x+y| > 1/2, \\ -1, & \text{otherwise.} \end{cases}$$

Figure 4 shows the function $f(x,y)$ and the adaptive mesh generated by the discretization technique outlined in Example 3. Figure 5 shows the computed solution. Timing results are summarized in Table 8.

4 Conclusions

We have developed an adaptive direct solver for inverting the square root of the Laplacian in two dimensions. While our first implementation relied on a classical spherical-harmonic

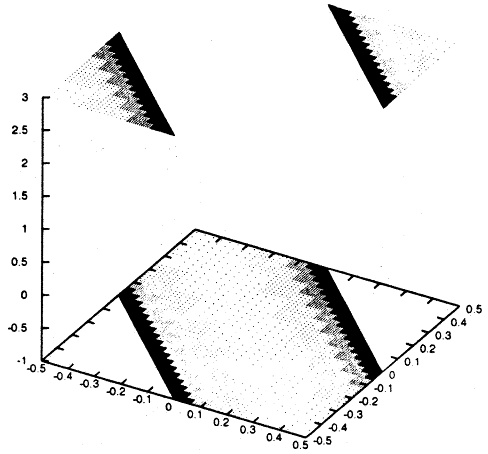
Table 7: Performance of the adaptive FMM using 24 singular functions for Example 3. The first column shows the tolerance in discretizing the right-hand side, and the second column shows the maximum number of levels used in the FMM. The third and fourth columns show the total number of boxes and total number of points in the discretization at that tolerance. The fifth column shows the number of points that would be required by a uniform grid at the finest level using an FFT-based scheme. The sixth column shows the time required (secs.) and the last column shows the number of points processed per second.

Tol	$Level$	N_{boxes}	N_{pts}	N_{uni}	T_{alg}	P_{24}
10^{-3}	7	889	10672	(262144)	0.328	32537
10^{-4}	7	1489	17872	(262144)	0.523	34172
10^{-5}	8	3265	39184	(1048576)	1.128	34736
10^{-6}	9	5369	64432	(4194304)	1.868	34493
10^{-7}	9	12977	155728	(4194304)	4.414	35280

Table 8: Performance of the adaptive FMM using 24 singular functions for Example 4. The columns are defined as in Table 7.

Tol	$Level$	N_{boxes}	N_{pts}	N_{uni}	T_{alg}	P_{24}
10^{-3}	7	1205	14464	(262144)	0.434	33327
10^{-4}	9	5045	60544	(4194304)	1.725	35098
10^{-5}	10	10165	121984	(16777216)	3.473	35124

Figure 4: The discontinuous right side f for Example 4.



based FMM, a faster scheme uses a generalized FMM [8], which constructs optimal representations for the far field using a singular value decomposition. The method can be used in free space or with periodic boundary conditions and the amount of work scales linearly with the number of grid points in the computational domain. The method is an order of magnitude slower than an FFT-based scheme on uniform grids, but quickly surpasses such schemes once adaptive refinement is required. Applications of the method to some problems in fluid dynamics will be reported at a later date.

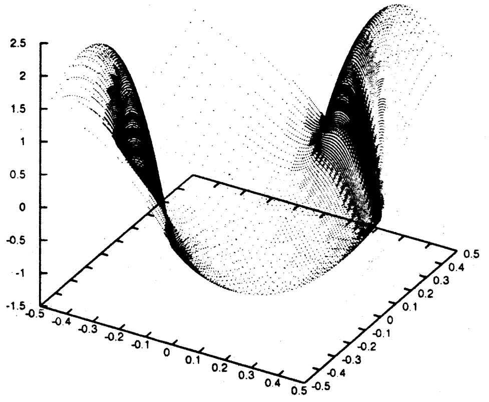
References

[1] G. Beylkin, R. Coifman and V. Rokhlin (1991). "Fast wavelet transforms and numerical algorithms I", *Comm. Pure and Appl. Math.* **44**, 141–183.

[2] A. Brandt and A. A. Lubrecht (1990), "Multilevel matrix multiplication and fast solution of integral equations", *J. Comput. Phys.* **90**, 348–370.

[3] W. Choi and R. Camassa (1996), "Weakly nonlinear internal waves in a two-fluid system", *J. Fluid Mech.* **313**, 83-103.

Figure 5: The potential induced by inverting the square root of the Laplacian using the right-side f in Figure 4.



[4] J. Carrier, L. Greengard, and V. Rokhlin (1988), "A fast adaptive multipole algorithm for particle simulations", *SIAM J. Sci. Statist. Comput.* **9**, 669–686.

[5] B. Cichocki, B. U. Felderhof (1989), "Electrostatic interactions in thin-film Coulomb systems with periodic boundary conditions", *Mol. Phys.* **67**, 1373–1384.

[6] P. Constantin, A. J. Majda, and E. Tabak (1994), "Formation of strong fronts in the 2-D quasi-geostrophic thermal active scalar", *Nonlinearity* **7**, 1495–1533.

[7] A. Faridani, D. V. Finch, E. L. Ritman, K. T. Smith (1997), "Local Tomography II", *SIAM J. Appl. Math.* **57**, 1095–1127.

[8] Z. Gimbutas (1999), *A Generalized Fast Multipole Method for Non-Oscillatory Kernels* Ph.D. Dissertation, Yale University, in preparation.

[9] L. Greengard (1988), *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, Cambridge, Mass.

[10] C. L. Berman and L. Greengard (1994), "A Renormalization Method for the Evaluation of Lattice Sums", *J. Math. Phys.* **35**, 6036–6048.

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

[12] L. Greengard and V. Rokhlin (1997). "A new version of the fast multipole method for the Laplace equation in three dimensions", *Acta Numerica* 6, 229-269.

[13] T. Hrycak and V. Rokhlin (1998). "An improved fast multipole algorithm for potential fields", *SIAM J. Sci. Statist. Comput.* **19**, 1804–1826.

[14] S. Kapur and D. E. Long (1998). "IES3: Efficient electrostatic and electromagnetic simulation", *IEEE Comput. Sci. Eng.* **5**, 60–67.

[15] J. D. Jackson (1975), *Classical Electrodynamics*, Wiley, New York.

[16] J. R. Phillips and J. K. White (1997), "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures" *IEEE Trans. Comput. Aid. D.* **16**, 1059–1072.

[17] N.R. Smith (1993), "Ocean modeling in a global ocean observing system", *Reviews of Geophysics* **31**, 281–317.

[18] P. R. Wallace (1984), *Mathematical Analysis of Physical Problems*, Dover, New York.

[19] H. Weber, M. Wallin, H. J. Jensen (1996), "Monte Carlo calculation of the current-voltage characteristics of a two-dimensional lattice Coulomb gas", *Phys. Rev. B* **53**, 8566–8574.

[20] A. H. Widmann and D. B. Adolf (1997), "A comparison of Ewald summation techniques for planar surfaces", *Comp. Phys. Comm.* **107**, 167–186.