Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 196 (2004) 591–626

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A kernel-independent adaptive fast multipole algorithm in two and three dimensions ☆

Lexing Ying, George Biros *, Denis Zorin

Courant Institute of Mathematical Sciences, New York University, New York 10012, USA

Received 17 July 2003; received in revised form 6 November 2003; accepted 11 November 2003

Abstract

We present a new fast multipole method for particle simulations. The main feature of our algorithm is that it does not require the implementation of multipole expansions of the underlying kernel, and it is based only on kernel evaluations. Instead of using analytic expansions to represent the potential generated by sources inside a box of the hierarchical FMM tree, we use a continuous distribution of an equivalent density on a surface enclosing the box. To find this equivalent density, we match its potential to the potential of the original sources at a surface, in the far field, by solving local Dirichlet-type boundary value problems. The far-field evaluations are sparsified with singular value decomposition in 2D or fast Fourier transforms in 3D. We have tested the new method on the single and double layer operators for the Laplacian, the modified Laplacian, the Stokes, the modified Stokes, the Navier, and the modified Navier operators in two and three dimensions. Our numerical results indicate that our method compares very well with the best known implementations of the analytic FMM method for both the Laplacian and modified Laplacian kernels. Its advantage is the (relative) simplicity of the implementation and its immediate extension to more general kernels.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Fast multipole methods; Fast solvers; Integral equations; Single-layer potential; Double-layer potential; Particle methods; *N*-body problems

1. Introduction

Many methods in computational physics (e.g., vortex methods, molecular dynamics) are based on the evolution of particle systems with pairwise interactions corresponding to potentials related to the fundamental solution of elliptic partial differential equations (PDEs). The most important among these kernels is the single-layer Laplacian. Other kernels include the kernels of the Stokes and Navier operators, their modified versions, and their derivatives (double-layer and hypersingular kernels).

☆ This work is supported by the National Science Foundation's Knowledge and Distributed Intelligence (KDI) program through Grant DMS-9980069.

* Corresponding author. Tel.: +1-212-998-3358; fax: +1-212-995-4122.

E-mail addresses: lexing@cs.nyu.edu (L. Ying), gbiros@acm.org, biros@cs.nyu.edu (G. Biros), dzorin@cs.nyu.edu (D. Zorin).

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.11.021

Particle formulations result in dense linear algebraic systems because all pairwise interactions have to be computed. This is a significant bottleneck since for N particles and results in a $\mathcal{O}(N^2)$ computation. In order to make large scale problems tractable it is essential to efficiently compute these interactions. A number of algorithms have been proposed for this purpose. The fast multipole method (FMM) has been one of the most successful, especially for nonuniform particle distributions.

In this paper, we present a new kernel-independent FMM-like algorithm. Our algorithm has the structure of the adaptive FMM algorithm [12] but requires only the kernel evaluations, and it does not sacrifice the efficiency of the original algorithm. The crucial element of our approach is to replace the analytic expansions and translations with *equivalent density* representations. These representations are computed by solving local exterior and interior problems on circles (2D), spheres or cubes (3D) using the integral equation formulations. We demonstrate the efficiency of our method in both 2D and 3D for many kernels: the single and double layer potentials of the Laplacian, the modified Laplacian, the Navier, the Stokes, and their modified variants. Our method has $\mathcal{O}(N)$ asymptotic complexity, and, like analytic FMM, works well for nonuniform particle distributions.

1.1. Synopsis of the new method

The basic structure of our method follows [14], the original fast multipole method, which we briefly review in Section 2. FMM consists of the following steps:

1. generation of a hierarchical tree partitioning of the computational domain;
2. accumulation of the multipole expansions for the far field by a postorder traversal of the tree;
3. translation of the multipole moments to the local expansions;
4. construction of local expansions by a preorder traversal of the tree;
5. evaluation of the far-field action on the particles using local expansions;
6. evaluation of the near field interactions.

The same steps are used in our algorithm. However in the postorder traversal of the tree, the multipole expansion construction is replaced by solving local exterior inverse problems. To represent the potential generated by particles inside a box, we use a continuous distribution of an equivalent density on a surface enclosing the box. To find this equivalent density on the surface, we match its potential to the potential of the original sources at another surface in the far field. The translations are done by direct evaluation on the far field, sparsified with SVD or FFT. During the preorder traversal of the tree, we evaluate the far-field interaction on a surface enclosing a target box, and solve an interior Dirichlet-type integral equation to compute an equivalent density. Then we use this density to represent the potential inside a target box.

Our method does not require implementation of analytic expansions for the kernel, it only requires their existence, and exclusively uses kernel evaluations. Like FMM, our algorithm is recursive and has an $\mathcal{O}(N)$ complexity. Additional properties like scale invariance and rotational symmetries of kernels can be used to further accelerate the translation step, as in the case of the standard FMM.

1.2. Related work

There are four basic classes of fast summation algorithms: (1) tree codes like Barnes–Hut [2], (2) fast multipole methods and (3) regular grid fast convolution methods like FFT [1] Our algorithm belongs to the second category. The description of the original fast multipole algorithm can be found in [14,23]. Although the method is highly successful in two dimensions, the three-dimensional version of the original method was

[1] This method is somewhat related to particle–particle (near field interaction) with particle-mesh algorithms. Particle-mesh methods use fast PDE solvers on regular grids (multigrid) to evaluate the far field contributions. In this paper, we are not reviewing these methods since they are mostly useful for uniform particle distributions.

inefficient. Efficient extensions in three dimensions were realized only recently [7]. For these reasons, many researchers tried to devise algorithms which were hybrids of tree codes and FMM, in order to combine the high accuracy of FMM methods with the simplicity of tree codes. In addition, the extension of the FMM to more general kernels like the modified Laplacian [13], the Stokes [10], and the Navier [9,26] operators can be quite cumbersome, due to the need to implement efficient translation operators. In this paper, we only review algorithms that could be used to develop kernel-independent methods.

The idea of using a set of equivalent sources was first introduced in [1]. In that paper, the far field is represented as the solution to an exterior Dirichlet problem on a ball surrounding the particles using the exact Green's function (Poisson formula) for Laplacian. The method is somewhat easier than FMM to implement, but requires the analytic form of the Green's function for each kernel, which may not be available in the general case.

In [3] instead of using the exact Green's function, a number of equivalent densities are placed on a Cartesian grid in each source box; these densities are computed analytically by matching a number of multipole moments in the multipole expansion series of the original source densities. An important feature of this method is the fact that the Cartesian grid allows the use of FFT to accelerate the multipole to local-expansions translations. However, the method is not kernel-independent since for different kernels different expansions have to be constructed. The same idea is used in [19], and like in Andersen's method the densities are distributed over a ball containing the source box.

The idea of equivalent densities is also used in the precorrected FFT method [21]. The equivalent densities are distributed over a regular grid, so that the far-field convolutions can be computed with FFT instead of FMM. The term "precorrected" is related to the computation of the local interactions: the subtraction of the local influence of the equivalent densities and the addition of the near field interactions. The regular grid sources are computed by matching the field at selected checking points, usually located on a ball enclosing the original sources. In [6], a precorrected FFT method is applied to the Helmholtz kernel, but the equivalent sources are distributed along the faces of an enclosing cube, and three FFTs along the coordinate system planes are used to compute the far interaction. FFT-based methods are very efficient, often faster than FMM due to much smaller constants. For uniform distributions of particles FFT is likely to be preferable and it is kernel-independent. However, in the case of highly irregular particle distributions FMM is more efficient.

A hybrid method for kernel independent matrix–vector multiplication algorithm was proposed in [16,17]. Based on the fact that large blocks of the particle interaction matrix are low rank, this method uses singular value decomposition to sample and sparsify these blocks. It can be applied recursively and attains a $\mathcal{O}(N \log N)$ complexity. We have applied this method on the Stokes and Navier operators [4,5] with very satisfactory results in both accuracy and speed. One serious shortcoming of this method is the high setup cost. For problems with static particle distributions this is not a concern, but it becomes a bottleneck for problems with time evolving particles. The SVD approach was been further explored in series of papers [11,24,25] to obtain a kernel-independent method that does not require the kernel to be a solution of an elliptic PDE or a convolution. However, due to its generality, as the authors of these papers assert, the method does not achieve the efficiency of FMM for kernels that are related to fundamental solutions of PDEs.

Another method for fast matrix multiplication is based on higher-order Taylor expansions in Cartesian coordinates. This approach is not suitable for high accuracy computations because is computationally expensive (for pth-order accuracy it requires $\mathcal{O}(p^d)$ expansion terms). However, it is a kernel-independent method (the higher-order expansions can be easily obtained by differentiation). For example, it has been used to accelerate problems with the Stokes kernel [22].

Another category of kernel-independent approaches used in solving boundary integral equations is based on wavelet decompositions, combined with a Galerkin scheme. This approach is quite promising, since it has the same complexity with FMM, and allows the constructions of efficient preconditioners for the resulting systems. However, it is hard to compare directly to FMM, as different trade-offs are made: FMM is a "bottom-up" approach, and is relatively insensitive to the distribution of samples. Adaptive wavelet methods are "top-down" but require samples to be located on a surface satisfying certain assumptions, which may not hold in the general case.

1.3. Organization of the paper

In Section 2, we briefly review the classical FMM algorithm for the two-dimensional Laplacian. In Section 3, we present the new algorithm and its implementation; in Section 4, we present an error analysis for the algorithm, and in Section 5, we present numerical results for several different scalar and vector kernels in two and three dimensions.

2. Review of the fast multipole method

Given N source densities $\{\phi_i\}$ located at N points $\{\mathbf{y}_i\}$ in $\mathbb{R}^d$ $(d = 2, 3)$, we want to compute the potential $\{q_i\}$ at N points $\{\mathbf{x}_i\}$ induced by a kernel G (single layer, double layer or other kernels of a elliptic PDE) using the following relation: [2]

$$q_i = q(\mathbf{x}_i) = \sum_{j=1}^{N} G(\mathbf{x}_i, \mathbf{y}_j)\phi(\mathbf{y}_j) = \sum_{j=1}^{N} G_{ij}\phi_j, \quad i = 1, \ldots, N.$$

Direct implementation of this summation gives an $\mathcal{O}(N^2)$ algorithm. For a large class of kernels and under reasonable assumptions on the particle distribution, FMM requires $\mathcal{O}(N)$ work to compute an approximate potential with a prescribed relative error [7,20]. The constant in the complexity estimate depends on the relative error. [3]

We will use the single layer Laplacian kernel to describe FMM. In two dimensions, we have $G(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi}\log\rho$, with $\mathbf{r} = \mathbf{x} - \mathbf{y}$, and $\rho = |\mathbf{r}|$. In the FMM context, it is convenient to use $G(\mathbf{x}, \mathbf{y}) = Re(\log(z_x - z_y))$, where z_x and z_y are complex numbers corresponding to $\mathbf{x}$ (target) and $\mathbf{y}$ (source) points on the plane. The idea of FMM is to encode the potentials of a set of source densities using the *multipole expansion* and *local expansion* at places far away from these sources. Suppose the source densities are supported in a disk centered at z_{C} with radius r. Then for all z outside the disk with radius R $(R > r)$, we can represent the potential at z from the source densities using a set of coefficients $\{a_k,\ 0 \leqslant k \leqslant p\}$, where

$$q(z) = a_0 \log(z - z_{\mathrm{C}}) + \sum_{k=1}^{p} \frac{a_k}{(z - z_{\mathrm{C}})^k} + \mathcal{O}\left(\frac{r^P}{R_p}\right) \quad \text{(Multipole expansion).} \tag{1}$$

On the other hand, if the source densities are outside the disk with radius R, the potential at a point z inside the disk with radius r can be represented using a set of coefficients $\{c_k,\ 0 \leqslant k \leqslant p\}$, where

$$q(z) = \sum_{k=0}^{p} c_k (z - z_{\mathrm{C}})^k + \mathcal{O}\left(\frac{r^P}{R_p}\right) \quad \text{(Local expansion).} \tag{2}$$

In both expansions, p is usually a small constant determining from the desired accuracy of the result. The definitions of the coefficients are given in Appendix B.

[2] We use $\mathbf{x}$ to refer to target locations and $\mathbf{y}$ to refer to source locations, but in general $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_i\}$ can be the same set of points.

[3] In the classical FMM, the absolute error of the potential is bounded by the product of the relative error and the total charge.

FMM employs the above representations in a recursive way. The computational domain, a box large enough to contain all source and target points, is hierarchically partitioned into a tree structure (a quadtree in 2D or an octtree in 3D). Each node of the tree corresponds to geometric box (square or cube). The tree is constructed so that the leaves contain no more than a prespecified number of points. For each box, the potential induced by its source densities is represented using a multipole expansion, while the potential induced by the sources from non-adjacent boxes is encoded in a local expansion. For a prescribed relative error ϵ, the number of expansion terms p is chosen to be $|\log_c \epsilon|$ where c is $(4-\sqrt{2})/\sqrt{2}$ in 2D and $(4-\sqrt{3})/\sqrt{3}$ in 3D.

Not only these expansions (multipole and local) can be used for efficient evaluation, but translations between these expansions are also available which make an $\mathcal{O}(N)$ algorithm possible. In particular, the following types of translations are used:

M2M. The *multipole to multipole* translation transforms the multipole expansions of a box's children to its own multipole expansion.

M2L. The *multipole to local* translation transforms the multipole expansion of a box to the local expansion of another non-adjacent box.

L2L. Finally, *the local to local* translation of the local expansion of a box's parent to its own local expansion. See Appendix B for the equations that define these translations.

Using the tree structure, FMM consists of two basic steps. During the first step, the upward pass, the tree is traversed in postorder [4] to compute the multipole expansion for each box. At the leaves, the multipole expansions are built following Eq. (1) (this procedure is also called the *source to multipole* (S2M) translation). At each non-leaf node, the multipole expansion is shifted from its children using the M2M translation. In the second step, the *downwards pass*, the tree is traversed in a preorder [5] to compute the local expansion. For each box B, the local expansion is the sum of two parts: first, the local-to-local transformation collects the local expansion of B's parent (the result condenses the contributions from the sources in all the boxes which are not adjacent to B's parent), and second, the multipole-to-local transformation collects the multipole expansions of the boxes which are the children of the neighbors of B's parent but are not adjacent to B (these boxes compose the interaction list of B). The sum of these two parts encodes all the contribution from the sources in the boxes which are not adjacent to *B itself*. At the end, for each box, the far interaction, which is evaluated using the local expansion at this box (this step is called the *local to target* (L2T) translation), is combined with the near interaction evaluated by iterating over all the source points in the neighborhood of the target box to obtain the potential (see Fig. 1).

Instead of Laurent series, in three dimensions the far field is represented by spherical harmonics. There are several implementation details (mostly for the M2L transformation) that are required for efficient implementation (especially in 3D), but we do not mention them here. Overall, however, the organization of the computation is the same as the two dimension case. For the derivation of the expansions and a detailed discussion on error bounds and implementation details see [7,14].

3. The new algorithm

Our algorithm is designed to generalize FMM to second-order constant coefficient non-oscillatory elliptic partial differential equations. Examples of such systems are given in Appendix A, where we also list the corresponding fundamental solution kernels. Such kernels satisfy the underlying PDE everywhere but the singularity location (pole), and are smooth away from the singularity. All problems under consideration

[4] The children of a box are visited before the box itself.

[5] The children of a box are visited after the box itself.

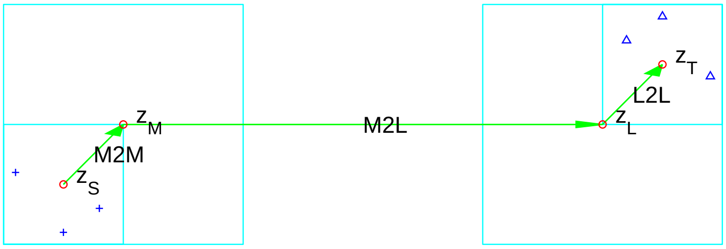


Fig. 1. The multipole expansion at z_S encodes the influence from the source densities (marked with "+") to the far field. The local expansion at z_T encodes the influence from the far field to the target points (marked with "△"). M2M translation transforms between the multipole expansions of the boxes in adjacent levels (z_S to z_M), M2L translation transforms multipole expansion of a box to the local expansion of non-adjacent boxes (z_M to z_L), and finally L2L translation transforms between local expansions between adjacent levels (z_L to z_T).

admit a unique solution for the properly posed interior/exterior Dirichlet problems. Smoothness and uniqueness are the basic properties that we use to develop our FMM approximation.

Our algorithm has the same structure with the original FMM method. The differences are how the densities are represented efficiently and how the M2M, M2L, and L2L transformations are computed. We first describe these representations and transformations, then state the complete algorithm and conclude with a discussion on efficient implementation. Below we summarize the notation we use in the description of the method; these notations are defined as follows:

B	a box in the computation tree
$\mathcal{N}^B$	the near range of the box B in $\mathbb{R}^d$
$\mathcal{F}^B$	the far range of the box B in $\mathbb{R}^d$
I_{s}^B	the set of indices of source points or densities in B
I_{t}^B	the set of indices of target points or potentials in B
$\mathbf{y}^{B,\mathrm{u}}$	the upward equivalent surface of B
$\phi^{B,\mathrm{u}}$	the upward equivalent density of B
$\mathbf{x}^{B,\mathrm{u}}$	the upward check surface of B
$q^{B,\mathrm{u}}$	the upward check potential of B
$\mathbf{y}^{B,\mathrm{d}}$	the downward equivalent surface of B
$\phi^{B,\mathrm{d}}$	the downward equivalent density of B
$\mathbf{x}^{B,\mathrm{d}}$	the downward check surface of B
$q^{B,\mathrm{d}}$	the downward check potential of B
p	the degree of discretization for equivalent densities
s	the maximum number of source (or target) points allowed in a leaf box
N	the total number of source and target points
R	the depth of the computation tree
M	the total number of boxes in the computation tree

3.1. Density translations

Given a set of N points, we define the computational domain to be a box large enough to contain all points. We construct a hierarchical tree (a quadtree in 2D and an octtree in 3D) so that each leaf of the tree contains no more than s points, where s is a prescribed number. We assume that some points are labeled as sources $\mathbf{y}_i$ and other points as targets $\mathbf{x}_i$. The source densities ϕ_i at the source locations $\mathbf{y}_i,\ i = 1, \ldots, N$ are given, and we want to evaluate the potential $\{q_i\}$ at the target locations $\{\mathbf{x}_i\}$.

We refer to the tree nodes (squares in 2D and cubes in 3D) as boxes. For a spatial region R, we use I_{s}^R and I_{t}^R to denote the index sets of the source and target points in R. Most commonly, R is a box of the computational tree.

If B is a box centered at $\mathbf{c}$ and has side length $2r$, then the box centered at $\mathbf{c}$ with side length $6r$ is called the *near range* of B and is denoted by $\mathcal{N}^B\mathbb{R}^d/\mathcal{N}^B$ is called the *far range* and is denoted by $\mathcal{F}^B$. Note that in our definition, B is a part of $\mathcal{N}^B$.

3.1.1. Equivalent densities and check potentials

We represent the potential in $\mathcal{F}^B$ from the source densities $\{\phi_i,\ i \in I_{\mathrm{s}}^B\}$ in B as the potential from a density distribution $\phi^{B,\mathrm{u}}$ supported at prescribed locations $\mathbf{y}^{B,\mathrm{u}}$ in $\mathcal{N}^B$ (Fig. 2). We call $\phi^{B,\mathrm{u}}$ the *upward equivalent density* and $\mathbf{y}^{B,\mathrm{u}}$ the *upward equivalent surface* of box B.

Results from potential theory put two restrictions on the positions of $\mathbf{y}^{B,\mathrm{u}}$ (see [18, Chapter 6]). First, to guarantee the smoothness of the potential produced by $\phi^{B,\mathrm{u}}$, its support $\mathbf{y}^{B,\mathrm{u}}$ should not overlap with $\mathcal{F}^B$. Second, to guarantee that $\phi^{B,\mathrm{u}}$ is able to represent the potential produced by any source distribution in B, $\mathbf{y}^{B,\mathrm{u}}$ needs to enclose B. Therefore, in order to ensure the existence of $\phi^{B,\mathrm{u}}$, $\mathbf{y}^{B,\mathrm{u}}$ is required to lie between B and the boundary of $\mathcal{F}^B$. We use a circle in 2D and a sphere or cube in 3D for reasons that will be explained later.

The potentials induced by the source densities and the upward equivalent density satisfy the underlying second-order linear elliptic PDE. As the solution of an exterior Dirichlet problem for such PDE is unique, these two potentials are guaranteed to be equal in all of $\mathcal{F}^B$ if they coincide at the boundary of $\mathcal{F}^B$, or any surface between $\mathcal{F}^B$ and $\mathbf{y}^{B,\mathrm{u}}$. We call such an intermediate surface the *upward check surface* and denote it by $\mathbf{x}^{B,\mathrm{u}}$. We call the potential computed on this surface the *upward check potential* and denote it by $q^{B,\mathrm{u}}$. These surfaces are also chosen to be circles in 2D, and spheres or cubes in 3D. The equality of potentials on the upward check surface can be written as follows:

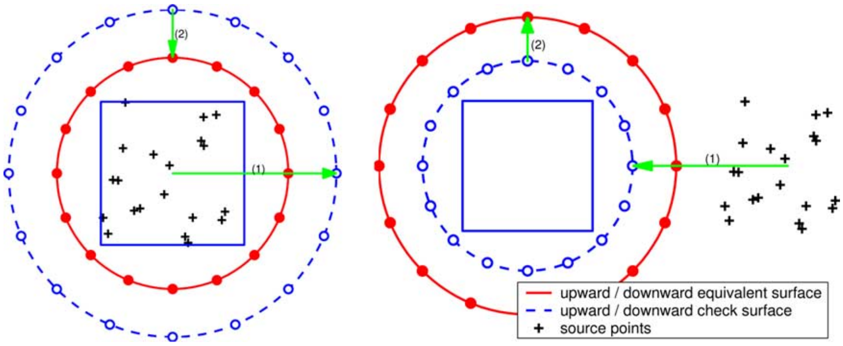

(2)
(1)
(2)
(1)
upward / downward equivalent surface
upward / downward check surface
source points

Fig. 2. The equivalent/check surfaces in 2D. Left: Given the potential generated by the source densities inside a box, located at the points marked with "+", we represent it by using the upward equivalent density located at the upward equivalent surface. The equivalent surface is shown as the solid circle enclosing the box. The upward check potentials induced by the sources and the upward equivalent density are matched at the upward check surface (the dashed circle). Right: To represent the potential in the box generated by the source in the far range, we use the downward equivalent density located at the downward equivalent surface. The downward equivalent potentials induced by both sources are matched at the upward check surface. In both plots, the discretization points of the equivalent and check surfaces are equally spaced and marked with "●" and "○", respectively. For both upward or downward steps, the computation of the equivalent density includes two steps shown by arrows in each plot: (1) the evaluation of the check potential using the original source, and (2) the inversion of the integral equation to obtain the equivalent density.

$$\int_{\mathbf{y}^{B,\mathrm{u}}} G(\mathbf{x},\mathbf{y})\phi^{B,\mathrm{u}}\,\mathrm{d}\mathbf{y} = \sum_{i\in I_s^B} G(\mathbf{x},\mathbf{y}_i)\phi_i = q^{B,\mathrm{u}} \quad \text{for any } \mathbf{x}\in\mathbf{x}^{B,\mathrm{u}}. \tag{3}$$

Similarly, we represent the potential in B from the source densities in $\mathscr{F}^B$ as the potential induced by a density distribution $\phi^{B,\mathrm{d}}$ defined at prescribed location $\mathbf{y}^{B,\mathrm{d}}$ in $\mathscr{N}^B$ (Fig. 2). We call $\phi^{B,\mathrm{d}}$ *downward equivalent density* and $\mathbf{y}^{B,\mathrm{d}}$ *downward equivalent surface.* To ensure the existence of $\phi^{B,\mathrm{d}}$, $\mathbf{y}^{B,\mathrm{d}}$ needs to be located between $\mathscr{F}^B$ and B. As the solution of the interior Dirichlet problem for the PDE we consider is also unique, we need to match the potentials only on a surface $\mathbf{x}^{B,\mathrm{d}}$ between B and $\mathbf{y}^{B,\mathrm{d}}$. We call the surface $\mathbf{x}^{B,\mathrm{d}}$ *downward check surface,* and the matched potential $q^{B,\mathrm{d}}$ the *downward check potential.*

We usually choose both $\mathbf{y}^{B,\mathrm{d}}$ and $\mathbf{x}^{B,\mathrm{d}}$ to be circles in 2D and spheres or cubes in 3D. The potential $\mathbf{y}^{B,\mathrm{d}}$ satisfies the following equation for any $\mathbf{x}\in\mathbf{x}^{B,\mathrm{d}}$:

$$\int_{\mathbf{y}^{B,\mathrm{d}}} G(\mathbf{x},\mathbf{y})\phi^{B,\mathrm{d}}\,\mathrm{d}\mathbf{y} = \sum_{i\in I_s^{\mathscr{F}^B}} G(\mathbf{x},\mathbf{y}_i)\phi_i = q^{B,\mathrm{d}}. \tag{4}$$

The integral equations (3) and (4) are the first-kind Fredholm equations. Inverting such equations for a general right-hand side is an ill-conditioned problem since it is an ill-posed infinite dimensional problem. However, the right-hand sides have a special form that guarantees the existence of the solution of the integral equation. To solve these equations numerically in a stable way, we use a regularization scheme, as discussed in Section 3.2.

3.1.2. M2M translation

For every leaf box B in the tree, the computation of the upward equivalent density $\phi^{B,\mathrm{u}}$ from the source densities uses Eq. (3). The procedure of M2M translation is similar (Fig. 3). To translate the upward equivalent density from a box A to its parent box B, we solve the following equation for $\phi^{B,\mathrm{u}}$:

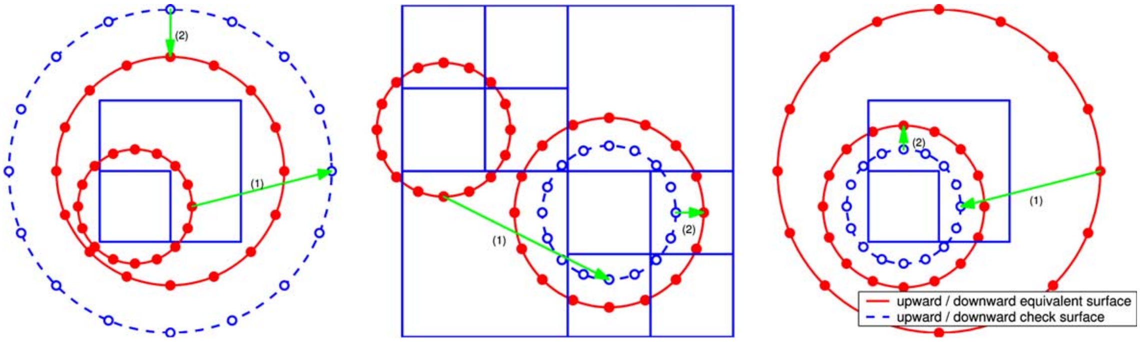

(1)
(2)
(1)
(2)
(1)
(2)
upward / downward equivalent surface
upward / downward check surface

Fig. 3. Three translations in 2D. Left: M2M translation. To compute the upward equivalent density of the large square, we evaluate the (upward check) potential at the dashed circle using its child box's upward equivalent density at the small solid circle (this operation is marked with arrow (1)), and invert the integral equation to get its upward equivalent density at the large solid circle (marked with arrow (2)). Middle: M2L translation transforms the upward equivalent density of the left box (surrounded by one circle) to the downward equivalent density of the right box (surrounded by two circles). We first evaluate the downward check potential at the dashed circle using the upward equivalent density (located at the small solid circle) (marked with (1)), and then invert the equation to obtain the downward equivalent density at the downward equivalent surface — the large solid circle (marked with (2)). Right: L2L translation transforms the downward equivalent density of the large box to its child — the the small box. In all three figures, the discretization points for the equivalent surface are marked with "●" and the ones for check surface are marked with "○".

$$\text{M2M}: \quad \int_{\mathbf{y}^{B,\mathrm{u}}} G(\mathbf{x},\mathbf{y})\phi^{B,\mathrm{u}}(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbf{y}^{A,\mathrm{u}}} G(\mathbf{x},\mathbf{y})\phi^{A,\mathrm{u}}(\mathbf{y})\,\mathrm{d}\mathbf{y} \quad \text{for all } \mathbf{x} \in \mathbf{x}^{B,\mathrm{u}}. \tag{5}$$

To ensure the existence of $\phi^{B,\mathrm{u}}$ for $B, \mathbf{y}^{B,\mathrm{u}}$ must enclose $\mathbf{y}^{A,\mathrm{u}}$ for any of its children A.

3.1.3. M2L translation

Once the upward equivalent density has been computed for each box, M2L translation computes the downward equivalent density (Fig. 3). Suppose A is a box in $\mathscr{F}^B$. The M2L translation is similar to (4), and we solve the following equation to find $\phi^{B,\mathrm{d}}$:

$$\text{M2L}: \quad \int_{\mathbf{y}^{B,\mathrm{d}}} G(\mathbf{x},\mathbf{y})\phi^{B,\mathrm{d}}(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbf{y}^{A,\mathrm{u}}} G(\mathbf{x},\mathbf{y})\phi^{A,\mathrm{u}}(\mathbf{y})\,\mathrm{d}\mathbf{y} \quad \text{for all } \mathbf{x} \in \mathbf{x}^{B,\mathrm{d}}. \tag{6}$$

To ensure the existence of $\phi^{B,\mathrm{d}}$, $\mathbf{y}^{B,\mathrm{d}}$ must be disjoint from $\mathbf{y}^{A,\mathrm{u}}$ for all A in $\mathscr{F}^B$.

3.1.4. L2L translation

The L2L translation computes the downward equivalent density of a box B at level i from that of its parent A at level $i-1$ (Fig. 3). The procedure is again similar to Eq. (4). The potential $\phi^{B,\mathrm{d}}$ satisfies

$$\text{L2L}: \quad \int_{\mathbf{y}^{B,\mathrm{d}}} G(\mathbf{x},\mathbf{y})\phi^{B,\mathrm{d}}(\mathbf{y})\,\mathrm{d}\mathbf{y} = \int_{\mathbf{y}^{A,\mathrm{d}}} G(\mathbf{x},\mathbf{y})\phi^{A,\mathrm{d}}(\mathbf{y})\,\mathrm{d}\mathbf{y} \quad \text{for all } \mathbf{x} \in \mathbf{x}^{B,\mathrm{d}}. \tag{7}$$

To ensure the existence of $\phi^{B,\mathrm{d}}$, $\mathbf{y}^{B,\mathrm{d}}$ must lie in $\mathbf{y}^{A,\mathrm{d}}$.

Eqs. (5), (6) and (7) corresponding to M2M, M2L and L2L translations are all ill-conditioned for an arbitrary right-hand side. However, similar to (3) and (4), the right-hand sides in our case are sufficiently smooth to guarantee the existence and stability of the solution of the integral equation.

3.1.5. Summary

We have described two density representations and three translations used to convert between these densities. The two equivalent densities correspond to the multi-pole and local expansions in FMM, while the three translations replace the three transformations in FMM.

In order to guarantee the existence of the equivalent densities the equivalent and check surfaces have to satisfy certain restrictions. We summarize them as follows: for each box B

- $\mathbf{y}^{B,\mathrm{u}}$ and $\mathbf{x}^{B,\mathrm{u}}$ lie between B and $\mathscr{F}^B$; $\mathbf{x}^{B,\mathrm{u}}$ encloses $\mathbf{y}^{B,\mathrm{u}}$;
- $\mathbf{y}^{B,\mathrm{d}}$ and $\mathbf{x}^{B,\mathrm{d}}$ lie between B and $\mathscr{F}^B$; $\mathbf{y}^{B,\mathrm{d}}$encloses $\mathbf{x}^{B,\mathrm{d}}$;
- $\mathbf{y}^{B,\mathrm{u}}$ encloses $\mathbf{y}^{A,\mathrm{u}}$ for any descendant box A,
- $\mathbf{y}^{B,\mathrm{u}}$ is disjoint from $\mathbf{y}^{A,\mathrm{d}}$ for all A in $\mathscr{F}^B$,
- $\mathbf{y}^{B,\mathrm{d}}$ lies inside $\mathbf{y}^{A,\mathrm{d}}$ where A is B's parent.

3.2. Discretization

3.2.1. Regularization

Eqs. (3), (5), (6) and (7) need to be discretized. Each one of them consists of two steps. First, we need to evaluate the check potential at box B using the equivalent density from box A. This step is discretized using a simple numerical quadrature. Second, we need to compute the equivalent density at B from the check potential computed in the previous step. This requires the numerical solution of a first-kind Fredholm equation. We denote this equation as

$$K\phi = q,$$

where ϕ is the equivalent density of B, q is the check potential of B and K evaluates q from the kernel and ϕ. We solve this equation using Tikhonov regularization [18]:

$$\phi = (\alpha I + K^*K)^{-1}K^*q.$$

This becomes a second-kind Fredholm integral equation, and in our implementation we solve it using the Nyström method (Galerkin or collocation methods could be used).

3.2.2. Surface geometry and discretization

The above two steps need to discretize the equivalent surfaces and check surfaces. In 2D, we choose circular equivalent and check surfaces. We use the trapezoidal rule to integrate the check potential and to discretize the integral equations; in this manner we obtain super-algebraic convergence. In 3D, this is no longer possible: to the best of our knowledge, there are no simple quadrature rules for functions defined on spheres that converge super-algebraically. Instead, we use cubes as the equivalent and check surfaces (Figs. 4 and 5), and construct quadratures of fixed order on the faces of the cubes. In Section 3.4, we explain how this approach facilitates fast M2L translations, and in Section 5 we show that the accuracy in 3D is not too different from the 2D case.

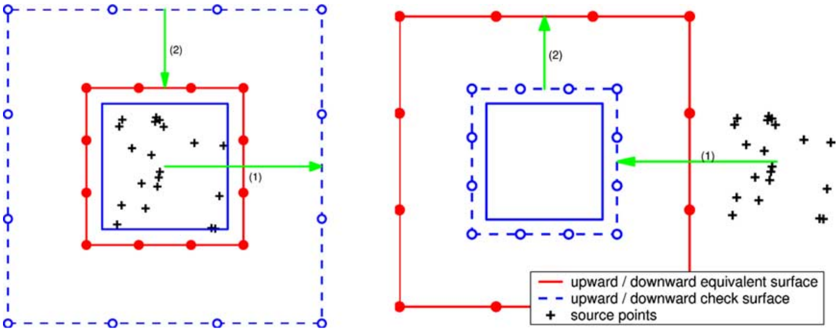

(2)
(1)
(2)
(1)
upward / downward equivalent surface
upward / downward check surface
source points

Fig. 4. The cross sections of the equivalent/check surfaces in 3D. Left: the upward equivalent density. Right: the downward equivalent density. In both plots, the innermost square is the source box. The equivalent and check surfaces are both discretized using the boundary nodes of a regular Cartesian grid. The nodes for the equivalent surfaces are marked with "●" and those for the check surfaces with "○".

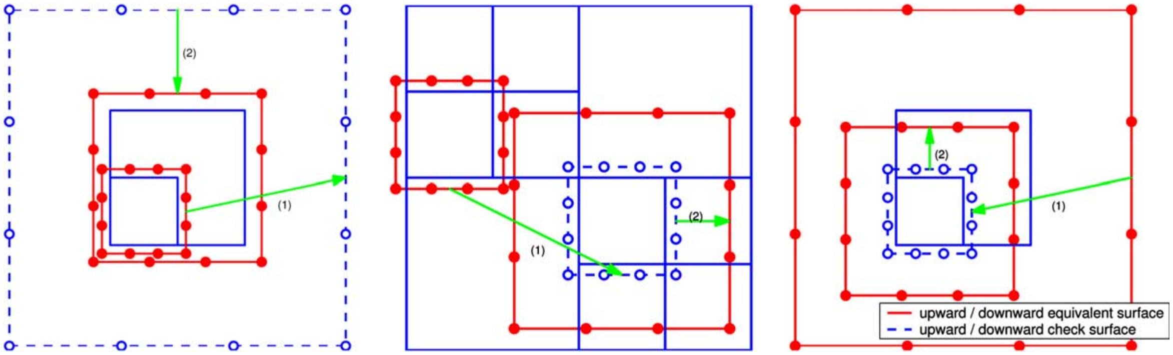

(2)
(1)
(1)
(2)
(2)
(1)
upward / downward equivalent surface
upward / downward check surface

Fig. 5. Three translations in 3D. Left: M2M translation. Middle: M2L translation. Right: L2L translation. 3D translations are similar to 2D. There are two differences: (1) equivalent/check surfaces are now cubes and (2) discretization points are the boundary nodes of a regular Cartesian grid. Note that for M2L translation the discretization points of upward equivalent surface and downward check surface are from the same Cartesian grid, therefore it can be sped up with FFT (interior nodes are padded with zero density).

3.2.2.1. 2D case. For a box B centered at $\mathbf{c}$ with side length $2r$, all related surfaces are circles centered at $\mathbf{c}$. The upward equivalent surface $\mathbf{y}^{B,\mathrm{u}}$ has radius $(\sqrt{2}+d)r$, where d a fixed number satisfying $0 \leqslant d \leqslant \frac{4-\sqrt{2}}{3}$. The upward check surface $\mathbf{x}^{B,\mathrm{u}}$ has radius $(4-\sqrt{2}-2d)r$. The downward equivalent surface $\mathbf{y}^{B,\mathrm{d}}$ has radius $(4-\sqrt{2}-2d)r$. Finally, the downward check surface $\mathbf{x}^{B,\mathrm{d}}$ has radius $(\sqrt{2}+d)r$ (Figs. 2 and 3). Note that our choice of the surfaces satisfies all restrictions at the end of Section 3.1. All circles are discretized with p equally spaced points with trapezoidal rule. The accuracy of our method is determined by the choice of p. This simple rule is known to have super-algebraic convergence for smooth functions, d is chosen to be quite small (equal to 0.1 in our implementation). By doing so, the equivalent surface and check surfaces involved in each translations are well-separated and the kernel used in the check potential integration step is very smooth. Therefore the trapezoidal rule gives good accuracy in the integration of check potential.

Remark 3.1. We could have chosen the upward/downward check surface to be identical with the upward/downward equivalent surface. However, in this case the integral equation would have a kernel-dependent form and we would need more complex quadrature rules that can be used to integrate singular kernels.

3.2.2.2. 3D case. For a box B centered at $\mathbf{c}$ with side length $2r$, all the related surfaces are the boundaries of cubes centered at $\mathbf{c}$. The upward equivalent surface $\mathbf{y}^{B,\mathrm{u}}$ is the boundary of a box with halfwidth $(1+d)r$ where $0 \leqslant d \leqslant 2/3$. The upward check surface $\mathbf{x}^{B,\mathrm{u}}$ is the boundary of a box with halfwidth $(3-2d)r$. The downward equivalent surface $\mathbf{y}^{B,\mathrm{d}}$ is the same as $\mathbf{x}^{B,\mathrm{u}}$. Finally, the downward check surface $\mathbf{x}^{B,\mathrm{d}}$ is the same as $\mathbf{y}^{B,\mathrm{u}}$ (Figs. 4 and 5). These surfaces satisfy the restrictions at the end of Section 3.1. For every surface, the quadrature points are distributed evenly on six faces, and on every face, the points are distributed on an evenly spaced 2D Cartesian grid. Under this distribution, the quadrature points at the corner of the box are shared by three faces, and those at the edge of the box are shared by two faces. We can also view these quadrature points as the boundary nodes of a 3D regular Cartesian grid. Similar to the 2D case, we use p to denote the total number of quadrature points on the surface of the box. [6] The quadrature weights are chosen in a way such that on every face the quadrature rule integrates low order 2D polynomials exactly. In our experiments, good quadrature results are observed since all the kernels are smooth away from the singularity. The parameter d is chosen to be quite small (again equal to 0.1 in our implementation) due to the reason stated in the 2D case.

3.2.3. Summary

Each one of the discretized M2M, M2L and L2L translations involves a potential evaluation and a solution of an integral equation. However, by choosing the quadrature points fixed relative to the box, both the evaluation and the solving depend only the level and the relative positions of the boxes involved in these translations. We can precompute and store these operators for each level and each relative position. Therefore, each translation invokes two matrix multiplications.

3.3. The complete algorithm

In this section, we describe our algorithm in detail. First we give some definitions related to the algorithm. Our definitions closely follow Greengard [12].

3.3.1. Definitions

The neighbors of a box are adjacent boxes in the same level. For uniform distributions of particles, a uniformly refined grid is used. In this case, the *neighbor list* L_{N}^{B} of a box B is the set of all neighbors of B and

[6] Note that, in 3D analytic FMM, p is the order of the multipole/local expansion, therefore, p^2 is the actual number of coefficients used in the expansion.

B itself. For a box away from the boundaries, the neighbor list contains nine boxes in 2D or 27 boxes in 3D. These boxes are all contained in $\mathcal{N}^B$.

The *interaction list* L_{I}^B is the set of children of the neighbors of B's parent which are not B's neighbors. Again, ignoring the boundary effects, this list contains 27 boxes in 2D and 189 boxes in 3D. These boxes are all contained in $\mathscr{F}^B$.

If the particle distribution is uniform, a regular grid can be used; however, we are primarily interested in non-uniform particle distributions. In this case, an adaptively refined grid is needed. The grid is recursively refined until the number of points in each leaf box is less than a fixed number s. Following the adaptive FMM algorithm, we give the following definitions (Fig. 6).

For a leaf box B, the *U list* L_U^B contains B itself and the leaf boxes which are adjacent to B. For a non-leaf box, the U list is empty.

The *V list* L_V^B is the set of the children of the neighbors of the parent of B which are not adjacent to B.

If B is a leaf box, the *W list* L_W^B consists of all the descendants of B's neighbors whose parents are adjacent to B, but who are not adjacent to B themselves. For a non-leaf box, the W list is empty.

The X list L_X^B consists of all boxes A such that $B \in L_W^C$.

For a leaf box B, L_U^B is similar to L_U^B in the uniform case, and L_V^B is similar to L_{I}^B. There is also a conjugate relation on these four lists. Suppose that A and B are two boxes.

- If A is in L_U^B, then B is in L_U^A.
- If A is in L_V^B, then B is in L_V^A.
- If A is in L_W^B, then B is in L_X^A.
- If A is in L_X^B, then B is in L_W^A.

For a box B, the U, V, W and X lists contain all boxes whose contribution needs to be processed by B itself. The contribution from more distant boxes are considered by B's ancestors. For a box U in L_U^B, a direct computation of the interaction of U's source points with B's target points is necessary since U and B are adjacent. For a box V in L_V^B we compute the interaction from V to B using M2L translation since two boxes

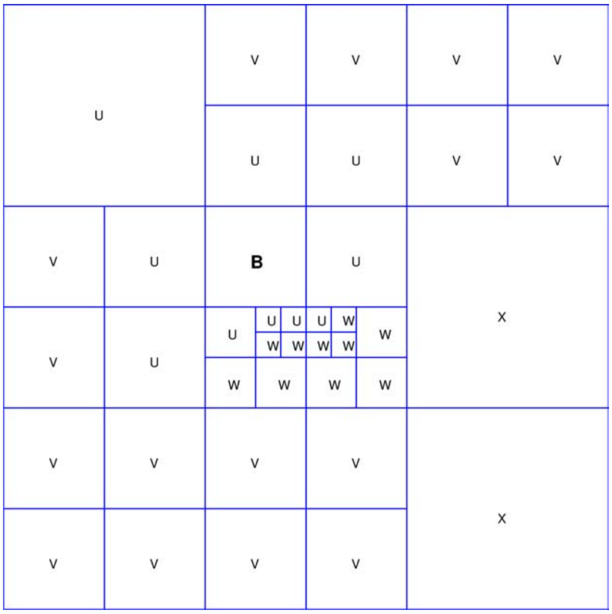


Fig. 6. Lists L_U^B, L_V^B, L_W^B and L_X^B of box B.

are well-separated. For a box W in L_W^B, we can evaluate the potential directly at B's target points using the upwards equivalent density of W, as B is in the far range of W. Finally, for a box X in L_X^B, since B is still in the near range of X, we represent the potential from X to B by first evaluating the potential at the downwards check surface at B and then invert it to the downwards equivalent density $\phi^{B,\mathrm{d}}$. The pseudocode is given in Algorithm 1.

3.4. Implementation issues

In the previous section, we described the overall structure of the algorithms with some implementation details omitted for clarity. These details, however, are very important for an efficient implementation of any FMM method. The most important issues are the efficient acceleration of the M2L computation, and the overall memory management.

Another aspect of our discussion is the distinction between the setup phase and the fast summation phase. Many times the particle distributions come from discretization of integral equations; then, given a fixed spatial particle distribution, the summation is carried many times (i.e., the matrix vector multiplication within an iterative solver such as GMRES). Many issues that we discuss here are related to efficient multiple evaluations.

3.4.1. Acceleration techniques

In our complexity analysis, we consider only the uniform particle distribution and uniform grids. While analysis of adaptive refinement is possible it requires assumptions on particle distribution. We refer the reader to [20]. The most expensive part of our algorithm are the M2L translations: the evaluation of the contribution to $q^{B,\mathrm{d}}$ of a target box B from $\phi^{A,\mathrm{u}}$, where A is a source box in the interaction list of B.

We denote the size of the interaction list by I. For a single box, the complexity of the M2L translation is $\mathcal{O}(Ip^2)$. The M2M and L2L translations are applied only once for each box and their contribution to the overall algorithm is not as important. Thus, the M2L part needs to be efficiently implemented since it is one of the two most expensive parts of the algorithm. (The other bottleneck is the computation of particle-to-particle dense interactions.)

3.4.1.1. SVD-based acceleration (2D).

Algorithm 1 (Adaptive case).

```
ASSUME
  $N$ is the total number of points
  $s$ is the maximum number of points allowed in leaf box
STEP 1 TREE CONSTRUCTION
  for each box $B$ in preorder traversal of the tree do
    subdivide $B$ if $B$ has more than s points in it
  end for
  for each box $B$ in preorder traversal of the tree do
    construct $L_U^B$, $L_V^B$, $L_W^B$ and $L_X^B$ for $B$
  end for
STEP 2 UPWARDS PASS
  for each leaf box $B$ in postorder traversal of the tree do
    evaluate $q^{B,\mathrm{u}}$ at $\mathbf{x}^{B,\mathrm{u}}$ using $\{\phi_i, i \in I_s^B\}$
    solve for $\phi^{B,\mathrm{u}}$ at $\mathbf{y}^{B,\mathrm{u}}$ that matches $q^{B,\mathrm{u}}$ at $\mathbf{x}^{B,\mathrm{u}}$ (Eq. (3))
  end for
  for each non-leaf box $B$ in postorder traversal of the tree do
```

```
        add to $q^{B,\mathrm{u}}$ at $\mathbf{x}^{B,\mathrm{u}}$ the contribution from $\phi^{C,\mathrm{u}}$ for each child $C$ of $B$
        solve for $\phi^{B,\mathrm{u}}$ at $\mathbf{y}^{B,\mathrm{u}}$ that matches $q^{B,\mathrm{u}}$ at $\mathbf{x}^{B,\mathrm{u}}$ (Eq. (5))
    end for
STEP 3 DOWNWARDS PASS
    for each non-root box $B$ in preorder traversal of the tree do
        add to $q^{B,\mathrm{d}}$ at $\mathbf{x}^{B,\mathrm{d}}$ the contribution from $\phi^{V,\mathrm{u}}$ for each box $V$ in $L_V^B$
        add to $q^{B,\mathrm{d}}$ at $\mathbf{x}^{B,\mathrm{d}}$ the contribution from $\{\phi_i, i \in I_\mathrm{s}^X\}$, for each box $X$ in $L_X^B$
        add to $q^{B,\mathrm{d}}$ at $\mathbf{x}^{B,\mathrm{d}}$ the contribution from $\phi^{P,\mathrm{d}}$, where $P$ is the parent of $B$
        solve for $\phi^{B,\mathrm{d}}$ at $\mathbf{y}^{B,\mathrm{d}}$ that matches $q^{B,\mathrm{d}}$ at $\mathbf{x}^{B,\mathrm{d}}$ (Eqs. (6) and (7))
    end for
    for each leaf box $B$ in preorder traversal of the tree do
        add to $\{q_i, i \in I_\mathrm{t}^B\}$ the contribution from $\phi^{B,\mathrm{d}}$
        add to $\{q_i, i \in I_\mathrm{t}^B\}$ the contribution from $\{\phi_i, i \in I_\mathrm{s}^U\}$ for each box $U$ in $L_U^B$
        add to $\{q_i, i \in I_\mathrm{t}^B\}$ the contribution from $\phi^{W,\mathrm{u}}$ for each box $W$ in $L_W^B$
    end for
```

In 2D, we use an SVD-based acceleration technique. We first assemble the matrix M of the interaction from $\mathbf{y}^{A,\mathrm{u}}$ to $\mathbf{x}^{B,\mathrm{d}}$. We observe that M is numerically low rank. The number of the significant singular values of M is small compared to the dimension of M, and the rest of the singular values are less than the accuracy required by the pairwise interaction evaluation. Suppose $USV^\mathrm{T} = M$ is the SVD of M. We can store only the columns of U and V which correspond to the dominant singular values of S and discard the rest. This approach gives us an efficient representation of M. In 3D, this approach does not yield satisfactory results. Although M2L operators are low rank, in practice the cutoff number of equivalent density points in which the compression is effective, is very large. For this reason an FFT-based approach is preferable.

3.4.1.2. FFT-based acceleration (3D). Suppose box A is in the interaction list of box B. As mentioned in Section 3.2, $\mathbf{y}^{A,\mathrm{u}}$ is chosen to be the boundary of A, and the integration points are the nodes of a Cartesian grid which are on the boundary of of A. The same is true for $\mathbf{x}^{B,\mathrm{d}}$. Therefore, by assigning zero density to the grid points in the interior of B we can view the evaluation of the potential $q^{B,\mathrm{d}}$ from the density $\phi^{A,\mathrm{u}}$ as a 3D convolution. This convolution can be evaluated efficiently by FFT. Since we use 3D convolutions, there are $\mathcal{O}(p^{3/2})$ instead of p densities and targets in each M2L translation. For each box, we carry out the FFT and inverse FFT only once, to obtain an $\mathcal{O}(p^{3/2}\log(p))$ complexity. The convolution (pointwise vector multiplication) is applied I times for each box, with $\mathcal{O}(Ip^{3/2})$ complexity.

Several acceleration schemes for the M2L translation of the analytic FMM have been proposed in the past. In [8], a 2D FFT-based scheme is used to transform the multipole coefficients to the local coefficients. This scheme gives a $\mathcal{O}(p\log(p))$ complexity for each M2L translation. In [7,15] *exponential representation,* an intermediate representation between multipole and local expansions is introduced. Based on this new representation, a *diagonal transformation* is used to transfer between exponential expansions efficiently. This technique cuts down the complexity to $\mathcal{O}(Ip)$. An essential step of the translation to exponential representation is the computation of some nontrivial kernel-dependent quadrature weights. While both of these two schemes give asymptotically superior complexity than the $\mathcal{O}(Ip^{3/2})$ complexity of our FFT-based acceleration technique, our FFT-based technique only involves potential evaluations and thus is kernel-independent.

Storage compression. Since the M2M, M2L and L2L translations are used repeatedly, we precompute and store the matrices of these operators. Three storage compression techniques are used to reduce the memory usage.

3.4.1.3. Homogeneity. Many kernels in the problems we are considering are homogeneous: if we scale the distance between the source point and the target point by a factor α, the potential at the target is amplified by a factor α^k, where k is a constant. For example, the 3D Laplace single layer kernel, $S(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi}\frac{1}{r}$, has this property. Since the integration points of the equivalent densities of a box are fixed relative to the box, the translation operators between different levels of the computation tree only differ by a constant, usually a power of 2. Hence, instead of storing the matrices for each level, we store only the matrices for a single level. Modified kernels, like modified Laplace, modified Stokes and modified Navier equations, do not have this property.

3.4.1.4. Symmetry. In 2D, the integration points are equally spaced on a circle; in 3D the integration points of the equivalent densities are chosen to be the nodes of a regular Cartesian grid. In both cases, they are symmetric with respect to the x, y and z axes. For example, if we flip the positive x-direction to be the negative x-direction, the positions of the set of the integration points do not change, even though two integration points might swap their positions. Consider the M2M translation: Suppose B is the parent box of two different boxes C_1 and C_2 and we need to evaluate the potential $q^{B,\mathrm{u}}$ at $\mathbf{x}^{B,\mathrm{u}}$, the contribution from $\phi^{C_1,\mathrm{u}}$ at $\mathbf{y}^{C_1,\mathrm{u}}$ and from $\phi^{C_2,\mathrm{u}}$ at $\mathbf{y}^{C_2,\mathrm{u}}$. Further suppose we already have the matrix of the operator from $\mathbf{y}^{C_1,\mathrm{u}}$ to $\mathbf{x}^{B,\mathrm{u}}$. In order to evaluate the contribution from $\phi^{C_2,\mathrm{u}}$ at $\mathbf{x}^{B,\mathrm{u}}$, we first perform a change of coordinates to move $\mathbf{y}^{C_2,\mathrm{u}}$ to $\mathbf{y}^{C_1,\mathrm{u}}$, and then evaluate the contribution using the operator from $\mathbf{y}^{C_1,\mathrm{u}}$ to $\mathbf{x}^{B,\mathrm{u}}$. We then perform another change of coordinates to move $\mathbf{y}^{C_1,\mathrm{u}}$ back to $\mathbf{y}^{C_2,\mathrm{u}}$. The same techniques can be carried out for M2L and L2L translations.

The above procedure is only correct in the case of a scalar density and a scalar potential. In the cases with vector or tensor densities and potentials, the change of coordinates not only affects the support of the density or potential, but it also modifies their values. Therefore, a rescaling step is necessary after each change of coordinates. A general translation using symmetry involves five steps: (a) forward change of coordinates, (b) rescaling of density, (c) translation using stored matrix, (d) rescaling of potential, and (e) backward change of coordinates. This technique works for all the kernels considered in this paper, and gives us a compression factor of eight in 3D and four in 2D.

3.4.1.5. Lazy computation. In the case of nonuniform density distribution, the depth of the computation tree can be quite large. However, not all the M2L translations are actually needed in the computation. Therefore, in our algorithm, the matrix representation of a M2L translation is only computed where it is actually needed by some box. This *lazy computation* strategy results in significant savings on memory usage in nonuniform density distributions, and modified kernels.

Complexity. The analysis of the adaptive algorithm is essentially the same, but more involved and requires assumptions about the particle distribution. For simplicity, we give the complexities of our method and FMM in [7] for 3D uniform particle distribution. The number of boxes M is approximately N/s. We use p to denote the number of coefficients.

Step	Our method	FMM
S2M translation	$\mathcal{O}(Np + Mp^2)$	$\mathcal{O}(Np)$
M2M translation	$\mathcal{O}(Mp^2)$	$\mathcal{O}(Mp^{3/2})$
M2L translation	$\mathcal{O}(Mp^{3/2}\log p + 189Mp^{3/2})$	$\mathcal{O}(20Mp^{3/2} + 189Mp)$
L2L translation	$\mathcal{O}(Mp^2)$	$\mathcal{O}(Mp^{3/2})$
L2T translation	$\mathcal{O}(Np)$	$\mathcal{O}(Np)$
Near interaction	$\mathcal{O}(27Ns)$	$\mathcal{O}(27Ns)$

The hidden constants in the complexity estimates are approximately the same for all translations; 189 is the number of the M2L boxes and 27 is the number of boxes in the near interaction. In practice, s is of the same order as p. Therefore, the S2M and L2T steps of both methods are of the same order $\mathcal{O}(Np)$. Our M2L translation is also of the same order as that of [7]. The M2M and L2L steps have higher complexity in our method, due to the fact that no acceleration techniques are applied in these two steps. However, in all experiments in Section 5, we observe that this does not slow down our method significantly since these steps are applied once for each box.

4. Error analysis

Given the direct interaction operator G between the sources in a box B at level l and targets in a well-separated target box A at level m, we examine the error related to the FMM approximation. First, we show that our FMM acceleration can be viewed as a factorization of G, provided that all integrations are carried out exactly. Then we present analysis of the discretization error behavior for homogeneous kernels from scale invariant PDE in 2D case. The scale invariance means that the PDE only involves the second-order derivatives of the potential variable, such PDEs includes Laplace, Stokes and Navier equations.

Numerical results indicate that the method works well in 3D and for inhomogeneous kernels; we leave derivation of rigorous error bounds in these cases as future work.

It is important to point out that here we prove an error bound of the FMM approximation of the interaction operator G. This error is a relative error in the sense that the absolute error for the computed potential is bounded by the product of the relative error with the magnitude of the exact potential.

4.1. FMM factorization

FMM can be viewed as a factorization of the operator G. Suppose the M2L translation operator is applied at level k when the interaction between A and B is evaluated. Let $B = b_l, b_{l-1}, \ldots, B_k$ be the sequence of ancestor boxes of B up to level k, and $A = A_m, A_{m-1}, \ldots, A_k$ the sequence of ancestor boxes of A. For our purposes, it is convenient to consider a single sequence of boxes, $B_l, \ldots, B_k$, $A_k, \ldots, A_m$, of length $l + m - 2k + 2$; we denote this single sequence $\{C_i\}$, $i = 0, \ldots, n+1$, where $n = l + m - 2k$. With each box C_i, we associate an equivalent surface $\mathbf{y}_i$, and a check surface $\mathbf{x}_i$, with equivalent density ϕ_i defined on $\mathbf{y}_i$ and potential q_i defined on $\mathbf{x}_i$. For boxes B_i upward surfaces are used, and for boxes A_i downward surfaces are used.

We introduce sequences of operators K_i and E_i mapping densities defined on equivalence surfaces to potentials defined on check surfaces. These operators correspond to left and right-hand sides of (5)–(7). We use an auxiliary operator $\mathscr{K}[Y \to X]$: $C(Y) \to C(X)$, where Y and X are regions in 2D or 3D (typically surfaces or boxes). The operator $\mathscr{K}$ is defined by

$$(\mathscr{K}[Y \to X]f)(\mathbf{x}) = \int_Y G(\mathbf{x}, \mathbf{y}) f(\mathbf{y})\, \mathrm{d}\mathbf{y} \quad \text{for } \mathbf{x} \in X.$$

Then

$$K_i = \mathscr{K}[\mathbf{y}_i - \mathbf{x}_i], \quad E_i = \mathscr{K}[\mathbf{y}_i \to \mathbf{x}_{i+1}], \quad L_i = E_i K_i^+, \tag{8}$$

where $K_i^+ = (K_i^* K_i)^{-1} K_i^*$ is the pseudoinverse of K_i.

Finally, let $D = \mathscr{K}[\mathbf{y}^{A,\mathrm{u}} \to A]$, the operator evaluating the density on the upward equivalent surface of $A = C_{n+1}$ at an arbitrary point inside A. Using these operators, evaluation of the potential q_A at the target box due to the sources in B using our hierarchical decomposition can be written in the following form:

$$q^A_{\text{hier}} = DK^+_{n+l}E_n \cdots E_0K^+_0 q^{B,\text{u}}. \tag{9}$$

Existence of the inverse operators K is discussed in Appendix C.

As illustrated in Fig. 7, the first sequence part of the sequence of operators corresponds to the upward traversal of the tree, with the M2M translation defined by (5) applied on each step. It is followed by the M2L translation (6) and the downward traversal with the L2L translation (7) applied on each step. Since the kernels are homogeneous, the operators K_i and E_i are level-independent of C_i up to an identical scale factor, and the composition $L_i = E_iK_i^+$ is level-independent as these factors cancel. For such kernels, we rescale E_i and k_i to make them completely level-independent.

In comparison, direct evaluation yields

$$q_{\text{direct}} = \sum_{i\in I^B_s} G(\mathbf{x},\mathbf{y})\phi_i.$$

Expression (9) can be viewed as a sequence of transformations of densities, starting with $\phi_0 = \phi^{B,\text{u}}$ to $\phi_{n+1} = \phi^{A,\text{d}}$, defined on the sequence of upward and downward equivalent surfaces. Let $\{D_i\}$ be the sequence of nested open domains with boundaries $\mathbf{x}_i$: $\text{Ext}(\mathbf{x}^{B_l,\text{u}}) \supset \cdots \supset \text{Ext}(\mathbf{x}^{B_k,\text{u}}) \supset \text{Int}(\mathbf{x}^{A_k,\text{d}}) \supset \cdots \supset \text{Int}(\mathbf{x}^{A_m,\text{d}}) \supset A$ (for the upward traversal, we use exterior domains, for the downward traversal, interior). Similarly we define $\{F_i\}$ to be the sequence of the nested open domains with boundary $\mathbf{y}_i : B \subset \text{Int}(\mathbf{y}^{B_l,\text{u}}) \subset \cdots \subset \text{Int}(\mathbf{y}^{B_k,\text{u}}) \subset \text{Ext}(\mathbf{y}^{A_k,\text{d}}) \subset \cdots \subset \text{Ext}(\mathbf{y}^{A_m,\text{d}})$.

It is sufficient to show that the potential q_i^{vol} in D_i induced by ϕ_i, $q_i^{\text{vol}} = \mathscr{K}[\mathbf{y}_i \to D_i]\phi_i$, is equal to q_{i+1}^{vol} in $D_{i+1} \subset D_i$, and that the potential induced by the first density ϕ_0 is the same as q_{direct} in D_0, the exterior of $\mathbf{x}^{B,\text{u}}$. Equivalence of q^A_{hier} and q_{direct} in the interior of A follows by induction.

The key is the observation that in the interior of D_i, q_i^{vol} satisfies the elliptic PDE for which the kernel $G(\mathbf{x},\mathbf{y})$ satisfies the underlying elliptic PDE. Therefore, we can regard it as the solution of the Dirichlet problem with boundary conditions $q_i^{\text{vol}}|_{\mathbf{x}_i} = q_i$. The Dirichlet problem is exterior for upward check surfaces $\mathbf{x}_i$, and interior for downward surfaces $\mathbf{x}_i$. In either case, from the uniqueness of the solution of the Dirichlet problem, it follows that the potential is defined uniquely by its boundary values. The density ϕ_{i+i} is computed from ϕ_i using $K_{i+1}\phi_{i+1} = E_i\phi_i$, i.e., the potentials induced by these densities on $\mathbf{x}_{i+1}$ are required

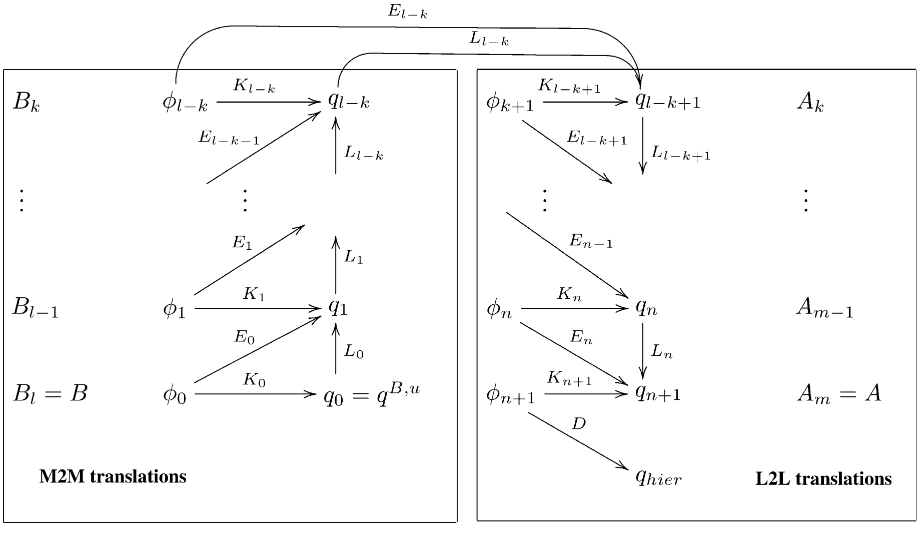


Fig. 7. Operators used in the error analysis.

to coincide. It follows that the potentials coincide in all of D_{i+1}. Similarly, ϕ_0 is computed using the condition that the induced potential coincides with $p^{B,\mathrm{u}}$, i.e., q_{direct} evaluated at $\mathbf{x}^{B,\mathrm{u}} = \mathbf{x}_0$; therefore, q_0 coincides with q_{direct} in D_0.

4.2. Discretization error

We present a qualitative error analysis in 2D, determining the dependence of the error on the tree depth l and the discretization error ϵ introduced at a single translation step. In 2D, the equivalent surfaces and check surfaces are chosen to be circles. Our analysis is carried out in the Sobolev spaces on a unit circle $H^t[0, 2\pi]$ for $t \geqslant 1$, which we denote H^t. We use $\| \cdot \|$ to denote the H^t norm. Since the kernel is C^∞ everywhere away from the singularity, q^A is in H^t for any t. Although the error is more naturally measured in L_2, H^t is a more convenient choice for analysis of our method, as the Nyström method for integral equations is norm-convergent in H^t for $t \geqslant 1$ in 2D. Note that this approach also yields an upper bound for the L_2 error, although this bound is likely to be too conservative.

We also define S_i, a subspace of H^t, with

$$S_i = \{\mathscr{K}[F_i \cup \mathbf{y}_i \to \mathbf{x}_i](q), q \in H^t\}. \tag{10}$$

Since the potential produced by the density in F_i, can be represented by the one produced by the density on $\mathbf{y}_i$, we can also write S_i to be $\{\mathscr{K}[\mathbf{y}_i \to \mathbf{x}_i](q), q \in H^t\}$.

To simplify the exposition, in our error analysis we omit the last step DK_{n+1}^{+} which introduces an additional fixed error due to solution of $K_{n+1}\phi_{n+1} = q_n$. Expression (9) with the last step excluded can be written as

$$q_n = L_n L_{n-1} \cdots L_0 q_0. \tag{11}$$

We use notation $L^{(j:i)}$ for the composition $L_j L_{j-1} \cdots L_i$ for $j \geqslant i$; we also abbreviate $L^{(j:0)}$ as $L^{(j)}$. We define $L^{(j:i)}$ to be the identity for $j < i$.

We use the following four auxiliary results in in our error analysis. The proofs of the first two lemmas can be found in Appendix C.

Lemma 4.1. *$E_i : H^t \to S_{i+1}$, $K_i : H^t \to S_i$ and $L_i : S_i \to S_{i+1}$ are all compact in the H^t norm.*

Lemma 4.2. *The H^t norm of any operator $L^{(j:i)} = L_j L_{j-1} \cdots L_i : S_i \to S_{j+1}$ is uniformly bounded independently of i and j.*

Lemma 4.3. *Suppose P_n is a sequence of bounded symmetric operators from H^t to H^t with $P_n \to I$ pointwisely, and D is a compact operator also from H^t to H^t. Then sequences P_nD and DP_n are norm convergent to D.*

Proof. Approximate D by a finite dimensional operator. □

Lemma 4.4. *In 2D, the Nyström method with trapezoidal rule is H^t norm convergent for second-kind Fredholm integral equations with smooth kernels.*

Proof. See Chapter 12 of [18]. □

As mentioned in Section 3.2, we use Tikhonov regularization to invert K_i. We introduce the regularized operator $\bar{L}_i$ as

$$\bar{L}_i = E_i(\alpha_i I + K_i^* K_i)^{-1} K_i^*$$

and its Nyström discretization by

$$\tilde{L}_i = \tilde{E}_i(\alpha_i I + \tilde{K}_i^*\tilde{K}_i)^{-1}\tilde{K}_i^*.$$

$\tilde{K}_i$ is the discretization of K_i defined by $\tilde{K}_i f(\mathbf{x}) = \sum_{r=1}^{p_i} w_i^r G(\mathbf{x}, \mathbf{y}_i^r) f \mathbf{y}_i^r$ for $\mathbf{x} \in \mathbf{x}_i$, where q_i is the number of quadrature points and w_i^r and $\mathbf{y}_i^r$ are quadrature weights and discretization points, respectively, $\tilde{E}_i$ is defined in the same way. It is important to notice that $\tilde{K}_i$ is from H^t to S_i since the quadrature points $\{\mathbf{y}_i^r\}$ stay on $\mathbf{y}_i$. Similarly, $\tilde{E}_i$ is an operator from H^t to S_{i+1}. Therefore, both $\tilde{L}_i$ and L_i are well-defined operators from S_i, to $S_i + 1$.

It can be shown that closure of S_i in H^t is the orthogonal complement of a finite number of functions. These functions span the null space of K. Therefore, L_i, can be extended to be defined over the whole H^t by using continuity and assigning L_i to be zero operator on these finite number of functions. The norm of the extension of L_i is bounded by the H^t norm of L_i on S_i. The compactness of L_i is also preserved. Similarly, the same argument applies to $L^{(j:i)}, \bar{L}_i$ and $\tilde{L}_i$. All of them can be defined over H^t. The goal of our analysis is to estimate the H^t norm of $\tilde{L}^{(n)} - L^{(n)} = \tilde{L}_n\tilde{L}_{n-1}\cdots\tilde{L}_0 - L_nL_{n-1}\cdots L_0$.

4.2.1. Single step error

Our first step is to estimate the error $\tilde{L}_i - L_i$ for a single translation step. We split the error into two parts: $\bar{L}_i - L_i$ and $\bar{L}_i - \tilde{L}_i$.

We regard H^t as a Hilbert space with the standard scalar product defined by $(f,g) = \sum_{i=0}^{t} \int_0^{2\pi} D^i f D^i g$. Since K_i is a compact operator in H^t (Lemma 4.1), for any $f \in H^t$ we can expand $K_i f$ as

$$K_i f = \sum_{r=0}^{\infty} \sigma_i^r (f, v_i^r) u_i^r,$$

where $\{u_i^r\}$ and $\{v_i^r\}$ are orthonormal bases in H^t and σ_i^r are singular values of K_i. In operator form, this decomposition can be written as $U_iS_iV_i$, where $V_i : H^t \to l_2$ is defined by the map from f to the sequence $\{(f, v_i^r)\}, U_i : l_2 \to H^t$ maps a sequence of coefficients $\{a^r\}$ to $\sum_r a^r u_i^r$, and $S_i : l_2 \to l_2$ is a diagonal operator with entries σ_i^r. Clearly, $U_iU_i^* = I$ and $V_iV_i^* = I$ because the bases $\{u_i^r\}$ and $\{v_i^r\}$ are orthonormal. Then

$$\bar{L}_i = E_i(\alpha_i I + K_i^*K_i)^{-1}K_i^* = E_iK_i^+K_i(\alpha_i I + K_i^*K_i)^{-1}K_i^* = L_iU_iS_i^2(\alpha_i I + S_i^2)^{-1}U_i^*.$$

As α_i, approaches 0, $U_iS_i^2(\alpha_i I + S_i^2)^{-1}U_i^*$ approaches I pointwisely. Since L_i is compact in H^t norm, $\bar{L}_i \to L_i$ in H^t norm as $\alpha_i \to 0$ (Lemma 4.3 applied to the extensions of L_i and $\bar{L}_i$ to H^t). Hence, for any fixed ϵ, we can choose a fixed α_i, such that $\|L_i - \bar{L}_i\| \leqslant \frac{\epsilon}{2}$.

Since Nyström's method is norm convergent for second-kind Fredholm integral equations in H^t (Lemma 4.4), as p_i increases, $(\alpha_i I + \tilde{K}_i^*\tilde{K}_i)^{-1}$ approaches $(\alpha_i I + \tilde{K}_i^*\tilde{K}_i)^{-1}$ in H^t norm. Therefore, for any fixed ϵ we can find p_i such that $\|\bar{L}_i - \tilde{L}_i\| \leqslant \frac{\epsilon}{2}$.

Combining the above estimates we get

$$\|\tilde{L}_i - L_i\| \leqslant \|L_i - \bar{L}_i\| + \|\bar{L}_i - \tilde{L}_i\| \leqslant \epsilon \tag{12}$$

by choosing α_i, and p_i based on ϵ.

Since the kernel is homogeneous and related to a scale invariant PDE, S_i and L_i depend only on the relative positions of the boxes C_i and C_{i+1}. Therefore, there are only finite number of operators L_i that can appear in the above analysis: 4 from each of the M2M and L2L translations and $7^2 - 3^2 = 40$ from the M2L translations. As we stated before, E_i and K_i can also be chosen to be level independent. Similarly, there are only a finite number of E_i and K_i operators as well. Therefore, we can choose α and p uniformly so that the estimate (12) applies for any L_i, K_i and E_i.

Total discretization error estimate. Using a single step norm estimate of $\tilde{L} - L$, we can estimate $\|\tilde{L}^{(n)} - L^{(n)}\|$ using Lemma 4.2. We use a constant C to denote the uniform bound for $L^{(i:j)}$ for all $0 \leqslant j \leqslant i \leqslant n$. Then for any i,

$$\|L^{(i)} - \tilde{L}^{(i)}\| = \left\| \sum_{j=0}^{i} L^{(i:j+1)}(L_j - \tilde{L}_j)\tilde{L}^{(j-1)} \right\| \leqslant C\epsilon\left(1 + \sum_{j=0}^{i-1} \|\tilde{L}^{(j)}\|\right).$$

This expression gives us a recurrence relationship on the norm of $\|\tilde{L}^{(i)}\|$:

$$\|\tilde{L}^{(i)}\| \leqslant C + C\epsilon\left(1 + \sum_{j=0}^{i-1} \|\tilde{L}^{(j)}\|\right).$$

Assuming $C \geqslant \epsilon$, from the recurrence we obtain

$$\|\tilde{L}^{(j)}\| \leqslant 2C(1 + C\epsilon)^j$$

and thus

$$\|\tilde{L}^{(n)} - L^{(n)}\| \leqslant C\epsilon\left(1 + \sum_{j=0}^{n-1} 2C(1 + C\epsilon)^j\right) = C(2((1 + C\epsilon)^n - 1) + \epsilon). \tag{13}$$

Although this estimate has an exponential dependence on n, it is only an upper bound and, in our experience, quite pessimistic. Moreover, our numerical experiments show that the uniform bound C is a small constant both in 2D and 3D for various kernels. Further, in actual calculations n is likely to be less than 40, and ϵ at least of order 10^{-4}. Therefore, in practice $(1 + C\epsilon)^n - 1$ behaves as $Cn\epsilon$.

Remark 4.1. Unlike our method, in the original analytic FMM method, there is no error associated with M2M, M2L and L2L transformations. The only error introduced in the analytic FMM are the S2M and M2T operators.

Remark 4.2. The basic parameters in our approximation are the regularization parameter a and the number of quadrature points p. In general, the regularization parameter a is chosen to filter out the noise or error in the data. In our experiments, we choose a to be a constant factor of the desired accuracy of the FMM approximation (ϵ) and then we choose the correct number of quadrature points by trial-and-error. The latter is very inexpensive because is independent of the size of the problem, and thus can be estimate quickly with a small test case.

Remark 4.3. The error associated with an approximate integral evaluation

$$q - \tilde{q} = \int G(\mathbf{x}, \mathbf{y})\phi - \sum_i w_i G(\mathbf{x}, \mathbf{y}_i)\phi_i$$

is the quadrature error. In 2D, we use the trapezoidal rule on the circle which is super-algebraically convergent. This enables us to approximate the operator L with $\tilde{L}$ with a small number of quadrature points. However, to our knowledge, in 3D there is no simple integration rule on the sphere that will result in similar high order accuracy; standard polynomial accuracy algorithms must be used. This is an important difference with the analytic FMM, which guarantees exponential convergence (on the number of multipole terms) for the far field approximation. Nonetheless, in our numerical experiments we did not observe noticeable differences between the 2D and 3D version.

5. Numerical results

In this section, we present numerical results for our method. First, we examine the accuracy of the equivalent density approximation. Second, we present results on the overall accuracy of the method.

5.1. Accuracy on the equivalent density approximation

In this section, we present results that indicate that our equivalent density approximations give good accuracy in both two and three dimensions.

For two and three dimensions, we show the results of three kernels: the Laplace single layer kernel, the modified Stokes double layer kernel and the Navier single layer kernel (Figs. 8 and 9). For each kernel, the left plot is the accuracy of the upward equivalent density approximation, and the right one is the accuracy of the downward equivalent density approximation. For the upward equivalent density, we give the error for points in the exterior of the source box in the region corresponding to the interaction list of the box. For the downward equivalent density, we give the error in the interior of the box. In all plots, the side length of the box is 2; we calculate the error by taking the maximum norm over a sphere centered at the center of the box. The abscissa of a plot is the radius of the sphere, and the ordinate is the logarithm of the error.

5.1.1. 2D case

Fig. 8 shows the error of the equivalent density approximation for the 2D Laplace single layer kernel, the 2D modified Stokes double layer kernel and the 2D Navier single layer kernel. In all three cases, the source density is located close to a corner of the box. The regularization parameter a is chosen to be 10^{-12} in all plots. Although not reported here, we have generated similar plots for all kernels given in Appendix A. All results exhibited similar accuracy. [7] We do not have a strict analytic error bound like the analytic FMM algorithm for the Laplace equation. However, Fig. 8 shows that our scheme gives comparable accuracy.

5.1.2. 3D case

Fig. 9 shows the equivalent density approximation errors for the 3D Laplace single layer kernel, the 3D modified Stokes double layer kernel and the 3D Navier single layer kernel. In each case, the source density is again placed close to one corner of the cube. The regularization parameter a used in these plots is 10^{-9}.

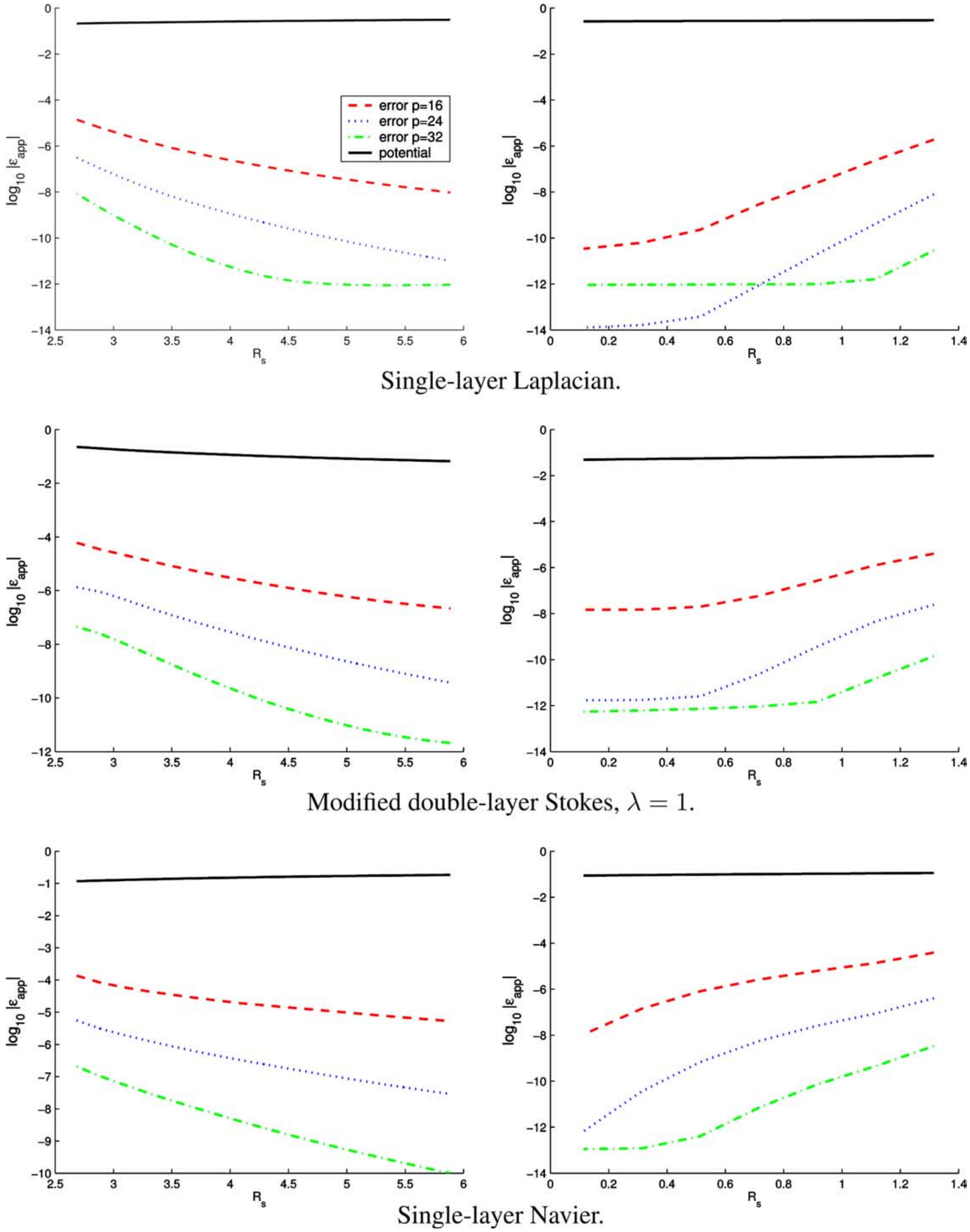
5.2. Overall approximation error

In this section, we give wall-clock time and memory requirements for several kernels. All experiments were performed on a Sun Ultra 80 workstation with a 450 MHz CPU. In 3D case, the FFTW package is used for FFT computation. Our code has been implemented in C++.

In our experiments, we assume that the source points and the target points coincide. We use three sets of density distributions in the cube with range $[-1, 1]$ in each dimension. The first set is a distribution on a sphere, which is typically nonuniform. The second set is a uniform distribution of density in a cube. The last set has densities only at one of the box corners. The objective of this set of points is to check the stability of multiple M2M and L2L transformations of our method. For all density distributions, the densities are chosen randomly from $[0, 1)$. The three data sets for the 3D case are shown in Fig. 10.

We organize the table in a way similar to [7].

[7] In some plots for 2D case, the 32-point error curve has larger error than the 24-point error curve. This is related to the regularization: we use 10^{-12} for α when solving the inverse problem and this complicates direct comparisons as we increase p.

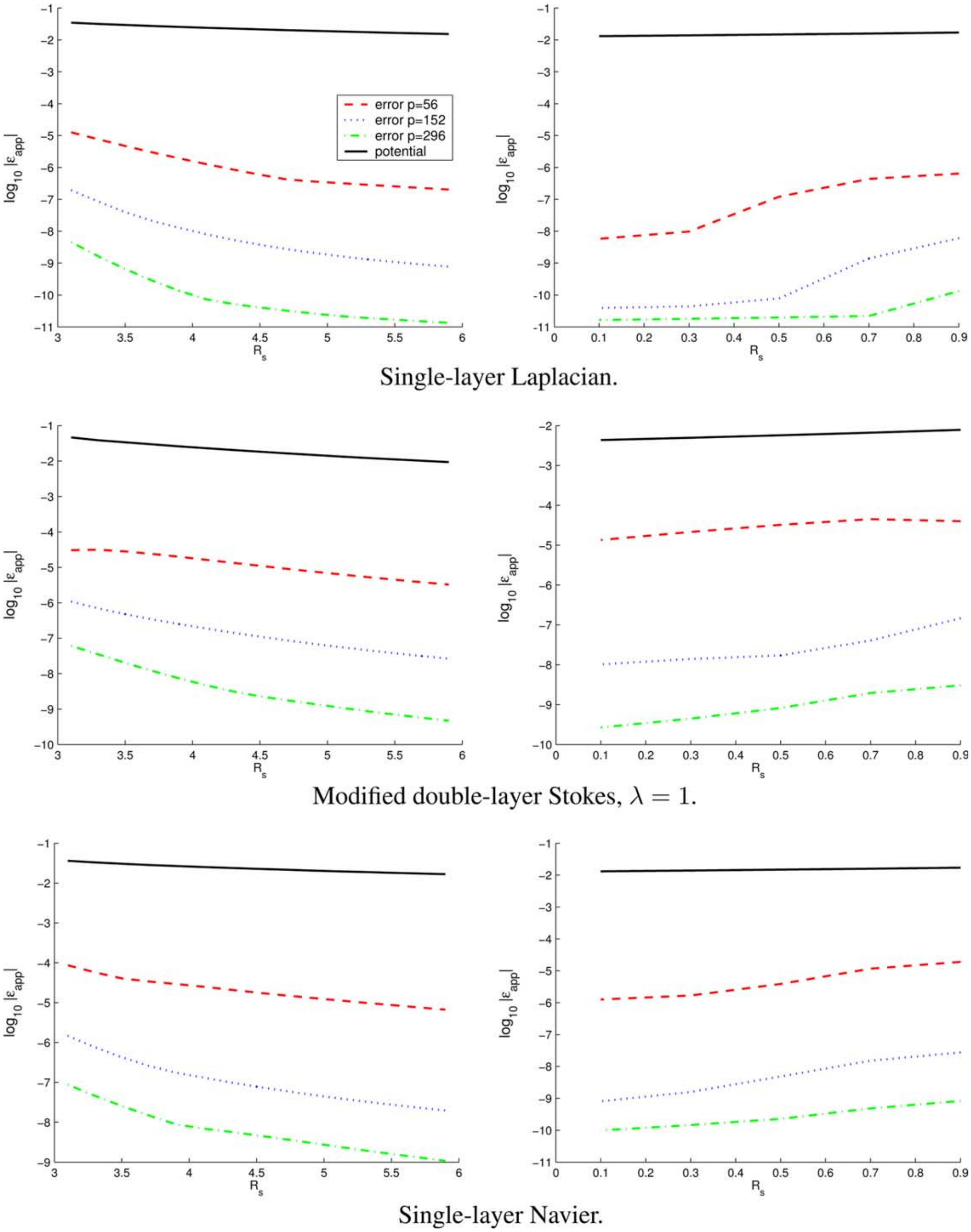


Single-layer Laplacian.

Modified double-layer Stokes, $\lambda = 1$.

Single-layer Navier.

Fig. 8. Results of the equivalent density approximation in 2D. Left: the error of the upward equivalent density approximation. Right: the error of the downward equivalent density approximation. The abscissa of the plots is the radius of the sphere R_s, and the ordinate is the logarithm of the error ϵ_{app}. The solid curve is the maximum norm of the potential. The remaining three curves show the maximum norm error for 16-, 24- and 32-point approximation of the equivalent densities. For modified Stokes, we tested λ from 1E − 3 to 1E + 3 and obtained similar error plots. For λ greater than 1E + 3, far field interaction is negligible.



Single-layer Laplacian.

Modified double-layer Stokes, $\lambda = 1$.

Single-layer Navier.

Fig. 9. Results of the equivalent density approximation in 3D. Left: the error of the upward equivalent density approximation. Right, the error of the downward equivalent density approximation. The abscissa of the plots is the radius of the sphere R_s, and the ordinate is the logarithm of the error ϵ_{app}. For each plot, the solid curve shows the maximum norm of the potential. The rest three plots show the maximum norm error where the equivalent density is approximated with 56, 152 and 296 points. These numbers correspond to discretization points that are the boundary nodes of volume Cartesian grids of size 4 × 4 × 4, 6 × 6 × 6, 8 × 8 × 8 (per box).

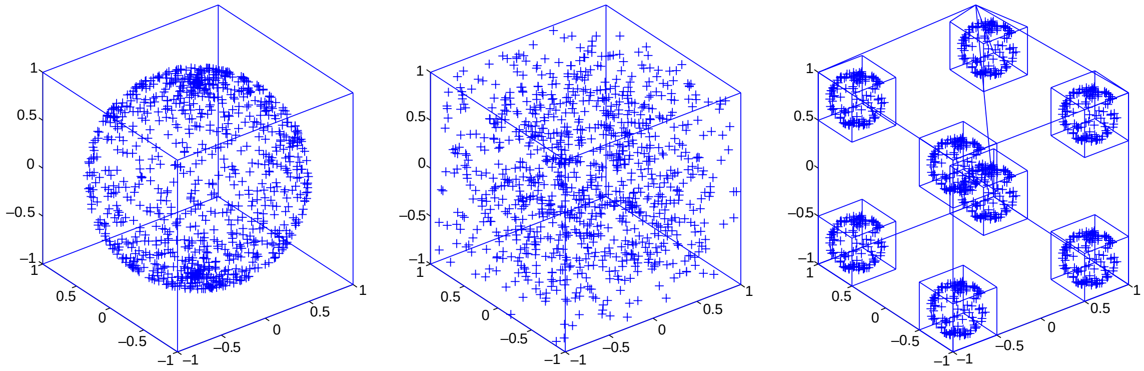


Fig. 10. Three data sets in 3D: Left: densities distributed on the unit sphere, Middle: densities distributed uniform in the unit cube, Right: densities distributed at the eight corners of the unit cube.

The columns of every table represent the following quantities.

N: the number of points used in computation (we use the same number of source and target points).

R: the number of levels of the computation tree.

M: the number of boxes in the computation tree.

p: the number of discretization points used in the equivalent density approximations. In 2D examples, we use 16, 24 and 32 points. In 3D examples, we choose the discretization points to be the boundary nodes of volume Cartesian grids of size $4 \times 4 \times 4$, $6 \times 6 \times 6$, $8 \times 8 \times 8$. These numbers correspond to 56, 152 and 296 points, respectively.

s: the maximum number of points allowed in a leaf box of the computation tree.

Storage: the memory used to store M2M, M2L, and L2L translations.

T_{fmm}: the running time of our algorithm.

T_{dir}: the running time of the direct evaluation. For each table, only the number in the first line is actually tested; all other numbers are obtained by extrapolation. The error is computed in relative 2-norm. We randomly select k points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$, evaluate the potential q_i, using our algorithm and the potential $\tilde{q}_i$ using direct evaluation at these k points. The error is estimated using the formula from [7]:

$$E = \left(\frac{\sum_{i=1}^{k} |q_i - \tilde{q}_i|^2}{\sum_{i=1}^{k} |\tilde{q}_i|^2} \right)^{1/2},$$

where k is chosen to be 40 in all experiments.

Below, we report the results on the first two data sets (nonuniform and uniform distribution) for five different kernels:

- 2D Laplace single layer kernel (Table 1),
- 3D Laplace single layer kernel (Table 2),
- 3D Modified Laplace single layer kernel (Table 3),
- 3D Modified Stokes double layer kernel (Table 4),
- 3D Navier single layer kernel (Table 5).

Our results from 2D are quite satisfactory since we can compute interactions between 2 million particles in six digits of accuracy in around 90 s, as we can see in Table 1. We discuss relative performance of our method in greater detail in the 3D case since this is more difficult to implement efficiently. We compare with results from two papers: the single-layer 3D Laplacian results of Cheng et al. [7] and modified single-layer 3D Laplacian results of Greengard and Huang [13].

Table 1
Performance for particles interacting via the single-layer Laplacian in 2D

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
The particles are uniformly distributed on the perimeter of a circle								
32,768	10	2989	16	40	1.52E+00	1.53E+00	1.71E+02	2.80E−06
131,072	12	11,857	16	40	1.91E+00	5.85E+00	2.74E+03	1.24E−06
524,288	14	47,241	16	40	2.30E+00	2.36E+01	4.39E+04	1.51E−06
2097,152	16	190,601	16	40	2.69E+00	9.32E+01	7.02E+05	2.80E−06
32,768	9	1597	24	60	2.97E+00	1.92E+00	1.71E+02	2.68E−08
131,072	12	6505	24	60	3.94E+00	7.47E+00	2.74E+03	2.84E−08
524,288	14	26,073	24	60	5.10E+00	2.97E+01	4.39E+04	3.36E−08
2,097,152	16	104,129	24	60	5.98E+00	1.24E+02	7.02E+05	2.24E−08
32,768	9	1493	32	80	5.28E+00	2.23E+00	1.71E+02	1.89E−10
131,072	11	5953	32	80	6.84E+00	1.03E+01	2.74E+03	1.77E−10
524,288	13	23,825	32	80	8.41E+00	4.04E+01	4.39E+04	7.05E−10
2,097,152	15	95,425	32	80	9.97E+00	1.49E+02	7.02E+05	6.03E−10
The particles are uniformly distributed inside a cube								
32,768	8	2837	16	40	1.14E+00	1.45E+00	1.71E+02	5.72E−07
131,072	10	12,245	16	40	1.53E+00	5.26E+00	2.74E+03	3.71E−07
524,288	12	47,829	16	40	1.92E+00	2.16E+01	4.39E+04	4.46E−07
2,097,152	14	189,717	16	40	2.31E+00	8.89E+01	7.02E+05	5.24E−07
32,768	7	1557	24	60	2.13E+00	1.78E+00	1.71E+02	2.05E−09
131,072	9	5909	24	60	3.01E+00	7.21E+00	2.74E+03	2.50E−09
524,288	11	25,557	24	60	3.88E+00	2.75E+01	4.39E+04	1.64E−09
2,097,152	14	104,085	24	60	4.85E+00	1.07E+02	7.02E+05	1.48E−09
327,68	7	1557	32	80	3.78E+00	2.12E+00	1.71E+02	2.83E−11
131,072	9	5269	32	80	5.34E+00	8.81E+00	2.74E+03	2.87E−11
524,288	11	23,893	32	80	6.91E+00	3.54E+001	4.39E+04	2.17E−11
2,097,152	13	95,253	32	80	8.47E+00	1.34E+02	7.02E+05	6.50E−11

In the first paper, the authors use a 167 MHz Sun workstation and in the second a 440 MHz Sun platform. As mentioned before we are using a 450 MHz Sun workstation. The metric we use for the purposes of comparison is the total number of CPU cycles in millions per grid point. We compute this number as

$$\eta = \frac{T_{\mathrm{fmm}} \times \mathrm{CPU}}{N},$$

where η_{a} and η are the numbers of cycles per particle for the analytic FMM and and for our algorithm, respectively. This is a only rough estimate that does not take into account the difference in chip architecture (e.g., memory bus clock), different floating point precision of the calculations (most calculations in the first paper were performed in single precision, all our results are in double precision), and different input densities.

First, we compare Table 2 with [7, Tables IV–VI]. For the three digit accuracy (Table IV) the average η_{a} is 0.07 for single precision. Our method achieves an η equal to 0.11 (in double digit accuracy), approximately a factor of 1.5 slower. Similar conclusions hold for the six-digit accuracy results (Table V), for which the analytic FMM achieves $\eta_{\mathrm{a}} = 0.15$ in single precision, whereas our method achieves $\eta = 0.23$ in double precision. For the modified single layer Laplacian we compare the six-digit accuracy entries [13, Table I] with Table 3 (uniform distribution in a cube). In this case, $\eta_{\mathrm{a}} = 0.3$ and $\eta = 0.4$, which is slightly better than 1.5; the actual difference in performance is even less, since we achieving about one additional digit of accuracy (average error 7×10^{-7} for the analytic FMM compared an average of 7×10^{-8} in our case).

Table 2
Performance for particles interacting via the single layer Laplacian in 3D

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
The particles are distributed on the surface of a sphere								
24,576	6	1377	56	60	1.72E+00	5.72E+00	9.74E+01	2.12E−05
98,304	7	5049	56	60	1.72E+00	2.38E+01	1.56E+03	3.21E−05
393,216	8	19,065	56	60	1.72E+00	9.51E+01	2.49E+04	6.08E−05
1,572,864	9	76,185	56	60	1.72E+00	3.82E+02	3.99E+05	6.03E−05
24,576	5	585	152	150	5.90E+00	1.16E+01	9.74E+01	3.34E−07
98,304	6	2289	152	150	5.90E+00	4.76E+01	1.56E+03	5.86E−08
39,3216	7	11,193	152	150	5.90E+00	2.18E+02	2.49E+04	2.45E−07
1,572,864	9	44,145	152	150	5.90E+00	8.35E+02	3.99E+05	3.08E−07
24,576	4	273	296	250	1.47E+01	1.81E+01	9.74E+01	1.59E−09
98,304	6	1449	296	250	1.47E+01	8.15E+01	1.56E+03	1.40E−09
393,216	7	5073	296	250	1.47E+01	3.41E+02	2.49E+04	1.10E−09
1,572,864	8	19,161	296	250	1.47E+01	1.38E+03	3.99E+05	2.81E−09
The particles are uniformly distributed inside a cube								
24,576	4	585	56	60	1.72E+00	6.40E+00	9.74E+01	6.64E−06
98,304	5	3657	56	60	1.72E+00	3.11E+01	1.56E+03	1.27E−05
393,216	7	28,233	56	60	1.72E+00	1.30E+02	2.49E+04	5.00E−05
1,572,864	8	88,137	56	60	1.72E+00	4.08E+02	3.99E+05	5.84E−05
24,576	4	585	152	150	5.90E+00	1.60E+01	9.74E+01	1.54E−08
98,304	5	3657	152	150	5.90E+00	9.28E+01	1.56E+03	4.70E−08
393,216	6	14,409	152	150	5.90E+00	3.18E+02	2.49E+04	1.10E−07
1,572,864	7	37449	152	150	5.90E+00	8.47E+02	3.99E+05	2.13E−07
24,576	4	585	296	250	1.47E+01	3.65E+01	9.74E+01	5.25E−10
98,304	4	585	296	250	1.47E+01	1.11E+02	1.56E+03	4.57E−10
393,216	5	3657	296	250	1.47E+01	4.31E+02	2.49E+04	6.85E−10
1,572,364	6	17,481	296	250	1.47E+01	1.46E+03	3.99E+05	1.46E−09

Another reason our method is slower might be related to the dense interactions. In order to save storage we do not precompute them, and we have found that this slows down our method by a factor of 2–4. The most time consuming part is computing the $1/\sqrt{(\mathbf{r}\cdot\mathbf{r})}$ term, which we have found impossible to optimize either with lookup tables or with special vector routines available from most vendors. For large problems that require several summations for the same particle partitions further running time improvements can be achieved by precomputing and storing all dense interactions. The memory requirements in this case can be substantial.

In conclusion, it appears that our method compares reasonably well with the analytic FMM by being a factor of 1.5 or less slower. Extending our code from the Laplacian to the modified Laplacian was very easy as we simply implemented a different kernel evaluation. Inspecting the results for the other kernels, we can confirm the $\mathcal{O}(N)$ complexity of our method and the convergence to the exact sum as we increase the number of quadrature points.

In all experiments, we store only the linear operators for M2M, M2L and L2L translations, since these operators are applied repetitively in a single pairwise interaction evaluation. The dense interactions between adjacent boxes are not stored. The storage number reported in all tables considers only the memory used by M2M, M2L and L2L operators, while the storage used to store the densities and potentials (which scales linearly with respect to the number of points and boxes) is not included. This explains why for the results of

Table 3
Performance of our method for particles interacting via the modified single layer Laplacian in 3D

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
The particles are distributed on the surface of a sphere								
6144	5	441	56	60	4.55E+00	1.97E+00	1.15E+01	3.55E−05
24,576	6	1377	56	60	6.27E+00	8.24E+00	1.83E+02	7.71E−05
98,304	7	5049	56	60	8.29E+00	3.33E+01	2.94E+03	3.11E−05
39,3216	8	19,065	56	60	1.00E+01	1.28E+02	4.70E+04	8.22E−05
6144	4	225	152	150	1.08E+01	4.38E+00	1.15E+0l	2.48E−07
24,576	5	585	152	150	1.57E+01	1.99E+01	1.83E+02	9.55E−08
98,304	6	2289	152	150	2.26E+01	7.58E+01	2.94E+03	3.18E−07
393,216	7	11,193	152	150	2.85E+01	3.39E+02	4.70E+04	3.63E−07
6144	3	57	296	250	1.18E+01	6.90E+00	1.15E+01	2.50E−09
24,576	4	273	296	250	2.64E+01	3.00E+01	1.83E+02	1.88E−09
98,304	6	1449	296	250	5.30E+01	1.23E+02	2.94E+03	1.96E−09
39,3216	7	5073	296	250	6.99E+01	5.35E+02	4.70E+04	3.71E−09
The particles are uniformly distributed in a cube								
6144	4	585	56	60	3.35E+00	3.72E+00	1.15E+01	5.28E−06
24,576	4	585	56	60	3.35E+00	1.06E+01	1.83E+02	2.29E−05
98,304	5	3657	56	60	5.07E+00	4.25E+01	2.94E+03	3.98E−05
393,216	1	28,233	56	60	8.14E+00	1.64E+02	4.70E+04	4.88E−05
6144	3	73	152	150	5.38E+00	4.09E+00	1.15E+01	2.10E−08
24,576	4	585	152	150	1.13E+01	2.11E+01	1.83E+02	9.86E−08
98,304	5	3657	152	150	1.72E+01	1.08E+02	2.94E+03	7.23E−08
393,216	6	14,409	152	150	2.31E+01	4.14E+02	4.70E+04	4.57E−08
6144	3	73	296	250	1.29E+01	5.87E+00	1.15E+01	7.15E−10
24,576	4	585	296	250	2.75E+01	4.39E+01	1.83E+02	6.02E−10
98,304	4	585	296	250	2.75E+01	1.98E+02	2.94E+03	4.28E−10
393,216	5	3657	296	250	4.22E+01	6.65E+02	4.70E+04	8.24E−10

homogeneous kernels (Tables 2 and 5), the storage numbers remain small and do not increase with the number of points and the number of levels.

Stability of multiple M2M and L2L translations. Here we test the stability of the M2M and L2L translations of our algorithm using the last data set which only has density distribution at the corners of the cube. Table 6 shows the result on this data set with 2D Laplace single layer kernel. Table 7 reports the errors with 3D Laplace single layer kernel.

6. Conclusions and future work

We have presented a new kernel-independent fast multipole method, which generalizes FMM to a broad class elliptic kernels while attaining an algorithmic complexity (including constants) which is on par with the analytic FMM. Here, we summarize the main features of our algorithm.

- Our algorithm has the same structure as the original adaptive FMM method.
- We have demonstrated that the method performs well for single and double layers, the Laplacian, the modified Laplacian, the Stokes, the modified Stokes, and the Navier kernels in two and three dimensions. By providing just a kernel evaluation routine our method is immediately applicable, as long as the kernel is associated with a non-oscillatory second-order elliptic PDEs.

Table 4
Performance of our method for particles interacting via the modified double layer Stokes kernel in 3D

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
The particles are distributed on the surface of a sphere								
6144	5	441	56	60	8.18E+01	2.65E+01	1.04E+02	9.56E-04
24,576	6	1377	56	60	1.13E+02	1.02E+02	1.66E+03	1.45E-03
98,304	1	5049	56	60	1.49E+02	3.91E+02	2.66E+04	1.47E-03
6144	4	225	152	150	2.00E+02	7.59E+01	1.04E+02	5.66E-06
24,576	5	585	152	150	2.92E+02	2.39E+02	1.66E+03	6.90E-06
98,304	6	2289	152	150	4.20E+02	l.OlE+03	2.66E+04	1.06E-05
6144	3	57	296	250	2.16E+02	6.44E+01	1.04E+02	8.77E-08
24,576	4	273	296	250	4.89E+02	3.59E+02	1.66E+03	1.67E-07
98,304	6	1449	296	250	9.87E+02	1.69E+03	2.66E+04	1.88E-07
The particles are uniformly distributed in a cube								
6144	4	585	56	60	6.03E+01	6.97E+01	1.04E+02	5.32E-04
24,576	4	585	56	60	6.03E+01	1.23E+02	1.66E+03	5.01E-04
98,304	5	3657	56	60	9.13E+01	6.09E+02	2.66E+04	7.00E-04
6144	3	73	152	150	9.87E+01	4.35E+01	1.04E+02	1.77E-06
24,576	4	585	152	150	2.09E+02	3.57E+02	1.66E+03	2.96E-06
98,304	5	3657	152	150	3.19E+02	2.04E+03	2.66E+04	9.32E-06
6144	3	73	296	250	2.36E+02	7.63E+01	1.04E+02	3.71E-08
24,576	4	585	296	250	5.09E+02	8.28E+02	1.66E+03	8.02E-08
98,304	4	585	296	250	5.09E+02	2.01E+03	2.66E+04	9.88E-08

Table 5
Performance of our method for particles interacting via the single layer Navier kernel in 3D

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
The particles are distributed on the surface of a sphere								
6144	5	441	56	60	1.55E+01	1.29E+01	5.91E+01	8.54E-05
24,576	6	1377	56	60	1.55E+01	4.93E+01	9.46E+02	6.71E-05
98,304	1	5049	56	60	1.55E+01	1.98E+02	1.51E+04	6.32E-05
6144	4	225	152	150	5.50E+01	3.29E+01	5.91E+01	1.07E-06
24,576	5	585	152	150	5.50E+01	1.10E+02	9.46E+02	1.66E-06
98,304	6	2289	152	150	5.50E+01	4.59E+02	1.51E+04	1.02E-06
6144	3	57	296	250	1.08E+02	3.28E+01	5.91E+01	7.30E-09
24,576	4	273	296	250	1.36E+02	1.82E+02	9.46E+02	8.51E-09
98,304	6	1449	296	250	1.36E+02	8.51E+02	1.51E+04	8.73E-09
The particles are uniformly distributed in a cube								
6144	4	585	56	60	1.55E+01	3.41E+01	5.91E+01	3.70E-05
24,576	4	585	56	60	1.55E+01	6.65E+01	9.46E+02	4.82E-05
98,304	5	3657	56	60	1.55E+01	3.13E+02	1.51E+04	6.68E-05
6144	3	73	152	150	4.94E+01	2.19E+01	5.91E+01	1.81E-07
24,576	4	585	152	150	5.50E+01	1.62E+02	9.46E+02	3.50E-07
98,304	5	3657	152	150	5.50E+01	9.48E+02	1.51E+04	4.86E-07
6144	3	73	296	250	1.18E+02	3.78E+01	5.91E+01	2.56E-09
24,576	4	585	296	250	1.36E+02	4.22E+02	9.46E+02	3.58E-09
98,304	4	585	296	250	1.36E+02	1.00E+03	1.51E+04	4.39E-09

Table 6
Performance of our method for a the 2D single layer Laplacian

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
524,288	18	47,449	16	40	2.17E + 00	2.17E + 01	4.39E + 04	4.46E – 06
524,288	18	26,041	24	60	4.54E + 00	2.63E + 01	4.39E + 04	1.20E – 08
524,288	17	23,833	32	80	7.91E + 00	3.50E + 01	4.39E + 04	1.04E – 10

In this experiment the particles are distributed over the boundaries of four circles. These circles are quite small compared the size of the (square) computational domain, and located near to the four corners of the domain. In this way the tree is ”forced” to have many levels (up to 18). We use this experiment to test the numerical stability of our M2M and L2L translations.

Table 7
Performance of our method for a the 3D single layer Laplacian

N	R	M	P	s	Storage (Mb)	T_{fmm} (s)	T_{dir} (s)	Error
196,608	12	11057	56	60	1.72E + 00	4.58E + 01	6.23E + 03	1.75E – 05
196,608	11	4721	152	150	5.90E + 00	1.04E + 02	6.23E + 03	1.20E – 07
196,608	10	2225	296	250	1.47E + 01	1.50E + 02	6.23E + 03	1.53E – 09

In this experiment the particles are distributed over the boundaries of eight spheres. These spheres are quite small compared the size of the (cubic) computational domain, and located near to the eight corners of the box.

- Comparisons of the running times between our method and the best known FMM implementations, and for same accuracy levels, indicate that our approach was successful in efficiently extending FMM to other kernels.
- I'o our knowledge, our results are the first fast summation computations for the modified Stokes and Navier operators.
- Our method is also directly applicable for derivatives of the kernels we have presented here. Indeed, we have tested our method on the hypersingular kernels resulting from differentiating the double layer Stokes and Navier equations.
- The M2L translations in our method are suboptimal. In 3D, the analytic exponential translations require $\mathcal{O}$(p), whereas our method requires $\mathcal{O}\,(p^{3/2})$, where p is the number of coefficients used in the approximation (the number of moments in the analytic FMM, and the number of discretization points in our method).
- Our method does not have a level independent error estimate that comes with the original FMM algorithm for Laplacian kernel. However, the error analysis in Section 4 shows that in practice the error can increase with the depth at most in a linear fashion.

In this paper, we have focused on second order constant coefficient PDEs with non-oscillatory solutions. However, our method is not restricted to such systems. It should be straightforward to generalize it to higher order systems like the biharmonic equation. In such cases the Dirichlet problem involves first and second derivatives of the underlying field. We can either differentiate the kernel to obtain the derivatives or use a set of two check-point surfaces. We plan to explore this approach in the future.

Another class of problems is related to second-order PDEs with oscillatory solutions or Helmholtz-type problems. For low frequencies, we have performed preliminary tests (on the M2M and L2L transformations) that indicate that our method works as is. An implementation for this class of problems, adding the kernels and support for complex numbers, is under way.

Finally let us mention that our method has been fully parallelized using MPI. Algorithmic details and numerical results will be presented in other papers.

Appendix A. Kernels

In this section, we give a summary of the elliptic partial differential equations (PDEs) studied in this paper and their relevant kernels. In the formulas below, $\mathbf{y}$ is the location of the singularity, $\mathbf{x}$ is the location the evaluation point, $\mathbf{n}$ a unit vector (usually the normal direction at $\mathbf{y}$), $\mathbf{r} = \mathbf{x} - \mathbf{y}$ and $r = |\mathbf{r}|$, denoting the length or $\mathbf{r}$. S stands for single layer and D for double layer.

Laplace equation:

$$-\Delta u = 0,$$

$$S(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1}{2\pi}\ln\frac{1}{r} & (2\mathrm{D}), \\ \frac{1}{4\pi}\frac{1}{r} & (3\mathrm{D}), \end{cases} \quad D(\mathbf{x},\mathbf{y}) = \begin{cases} -\frac{1}{2\pi}\frac{1}{r^2}(\mathbf{r}\cdot\mathbf{n}) & (2\mathrm{D}), \\ -\frac{1}{4\pi}\frac{1}{r^3}(\mathbf{r}\cdot\mathbf{n}) & (3\mathrm{D}). \end{cases}$$

Modified Laplace equation:

$$\alpha u - \Delta u = 0,$$

$$S(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1}{2\pi}k_0(\lambda r) & (2\mathrm{D}), \\ \frac{1}{4\pi}\frac{1}{r}\mathrm{e}^{-\lambda r} & (3\mathrm{D}), \end{cases} \quad D(\mathbf{x},\mathbf{y}) = \begin{cases} -\frac{\lambda}{2\pi}\frac{k_1(\lambda r)}{r}(\mathbf{r}\cdot\mathbf{n}) & (2\mathrm{D}), \\ -\frac{1}{4\pi}\left(\frac{1}{r^3}+\frac{\lambda}{r^2}\right)\mathrm{e}^{-\lambda r}(\mathbf{r}\cdot\mathbf{n}) & (3\mathrm{D}), \end{cases}$$

where $\lambda = \sqrt{\alpha}$.

Stokes equation (incompressible creeping flows):

$$-\mu\Delta q + \nabla p = 0, \quad \mathrm{Div}\, q = 0,$$

$$S(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1}{4\pi\mu}\left(\ln\frac{1}{r}I + \frac{\mathbf{r}\otimes\mathbf{r}}{r^2}\right) & (2\mathrm{D}), \\ \frac{1}{8\pi\mu}\left(\frac{1}{r}I + \frac{\mathbf{r}\otimes\mathbf{r}}{r^3}\right) & (3\mathrm{D}), \end{cases} \quad D(\mathbf{x},\mathbf{y}) = \begin{cases} -\frac{1}{\pi}\frac{\mathbf{r}\otimes\mathbf{r}}{r^4}(\mathbf{r}\cdot\mathbf{n}) & (2\mathrm{D}), \\ -\frac{6}{8\pi}\frac{\mathbf{r}\otimes\mathbf{r}}{r^5}(\mathbf{r}\cdot\mathbf{n}) & (3\mathrm{D}). \end{cases}$$

Modified Stokes equation (unsteady incompressible creeping flows):

$$\alpha q - \mu\Delta q + \nabla p = 0, \quad \mathrm{Div}\, q = 0,$$

$$S(\mathbf{x},\mathbf{y}) = \frac{1}{\mu}(GI + H(\mathbf{r}\otimes\mathbf{r})),$$

$$D(\mathbf{x},\mathbf{y}) = A((\mathbf{r}\cdot\mathbf{n})I + \mathbf{n}\otimes\mathbf{r}) + B(\mathbf{r}\otimes\mathbf{n}) + C(\mathbf{r}\cdot\mathbf{n})(\mathbf{r}\otimes\mathbf{r}),$$

where

$$G = -f_{rr} - (d-2)\frac{f_r}{r},$$

$$H = \frac{f_{rr}}{r^2} - \frac{f_r}{r^3},$$

$$A = -\frac{f_{rrr}}{r} - (d-3)\frac{f_{rr}}{r^2} + (d-3)\frac{f_r}{r^3},$$

$$B = -p + 2\frac{f_{rr}}{r^2} - 2\frac{f_r}{r^3},$$

$$C = 2\frac{f_{rrr}}{r^3} - 6\frac{f_{rr}}{r^4} + 6\frac{f_r}{r^5},$$

and

$$f = \begin{cases} \frac{1}{2\pi\lambda^2}\left(\ln(\frac{1}{r}) - k_0(\lambda r)\right) & (2\mathrm{D}), \\ \frac{1}{4\pi\lambda^2}\left(\frac{1}{r} - \frac{1}{r}\mathrm{e}^{-\lambda r}\right) & (3\mathrm{D}), \end{cases} \quad p = \begin{cases} \frac{1}{2\pi}\frac{1}{r^2} & (2\mathrm{D}), \\ \frac{1}{4\pi}\frac{1}{r^4} & (3\mathrm{D}), \end{cases} \quad \lambda = \sqrt{\frac{\alpha}{\mu}}.$$

Navier equation (elastostatics):

$$-\mu\Delta q - \frac{\mu}{1-2v}\nabla\cdot\mathrm{Div}\,q = 0,$$

$$S(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1}{\mu}\left(\frac{3-4v}{8\pi(1-v)}\log(\frac{1}{r}) + \frac{1}{8\pi(1-v)}\frac{(\mathbf{r}\otimes\mathbf{r})}{r^2}\right) & (2\mathrm{D}), \\ \frac{1}{\mu}\left(\frac{3-4v}{16\pi(1-v)}\frac{1}{r} + \frac{1}{16\pi(1-v)}\frac{(\mathbf{r}\otimes\mathbf{r})}{r^3}\right) & (3\mathrm{D}), \end{cases}$$

$$D(\mathbf{x},\mathbf{y}) = \begin{cases} \frac{1-2v}{4\pi(1-v)}\left(-\frac{((\mathbf{r}\cdot\mathbf{n})I+\mathbf{n}\otimes\mathbf{r})}{r^2} + \frac{(\mathbf{r}\otimes\mathbf{n})}{r^2} - \frac{2}{1-2v}\frac{(\mathbf{r}\cdot\mathbf{n})(\mathbf{r}\otimes\mathbf{r})}{r^4}\right) & (2\mathrm{D}), \\ \frac{1-2v}{8\pi(1-v)}\left(-\frac{((\mathbf{r}\cdot\mathbf{n})I+\mathbf{n}\otimes\mathbf{r})}{r^3} + \frac{(\mathbf{r}\otimes\mathbf{n})}{r^3} - \frac{3}{1-2v}\frac{(\mathbf{r}\cdot\mathbf{n})(\mathbf{r}\otimes\mathbf{r})}{r^5}\right) & (3\mathrm{D}). \end{cases}$$

Modified Navier equation (Elastodynamics):

$$\alpha q - \mu\Delta q - \frac{\mu}{1-2v}\nabla\cdot\mathrm{Div}\,q = 0,$$

$$S(\mathbf{x},\mathbf{y}) = \frac{1}{\mu}(GI + H(\mathbf{r}\otimes\mathbf{r})),$$

$$D(\mathbf{x},\mathbf{y}) = ((\mathbf{r}\cdot\mathbf{n})I + \mathbf{n}\otimes\mathbf{r}) + B(\mathbf{r}\otimes\mathbf{n}) + C(\mathbf{r}\cdot\mathbf{n})(\mathbf{r}\otimes\mathbf{r}),$$

where

$$G = \eta^2 f - f_{rr} + (\beta + 1 - d)\frac{f_r}{r},$$

$$H = \beta\frac{f_{rr}}{r^2} - \beta\frac{f_r}{r^3},$$

$$A = -\frac{1}{r}f_{rrr} + \frac{2\beta+1-d}{r^2}f_{rr} + \left(\frac{\eta^2}{r} - \frac{2\beta+1-d}{r^3}\right)f_r,$$

$$B = \frac{\gamma(\beta-1)}{r}f_{rrr} + \frac{2\beta+\gamma(\beta-1)(d-1)}{r^2}f_{rr} + \left(\frac{\gamma\eta^2}{r} - \frac{2\beta+\gamma(\beta-1)(d-1)}{r^3}\right)f_r,$$

$$C = \frac{2\beta}{r^3}f_{rrr} - \frac{6\beta}{r^4}f_{rr} + \frac{6\beta}{r^5}f_r$$

and

$$f = \begin{cases} \frac{1}{2\pi(\lambda^2-\eta^2)}(k_0(\eta r) - k_0(\lambda r))(2D), \\ \frac{1}{4\pi(\lambda^2-\eta^2)}(\frac{1}{r}\mathrm{e}^{-\eta r} - \frac{1}{r}\mathrm{e}^{-\lambda r})(3D), \end{cases}$$

$$\lambda = \sqrt{\frac{\alpha}{\mu}}, \quad \eta = \sqrt{\frac{1-2v}{2(1-v)}\cdot\frac{\alpha}{\mu}}, \quad \beta = \frac{1}{2(1-v)}, \quad \gamma = \frac{2v}{1-2v}.$$

Appendix B. Coefficients of fast multipole method

We give the coefficients of the FMM for 2D single layer Laplacian. Fig. 1 illustrates the relative positions of the symbols used in the following equations.

Multipole expansion. Suppose the m source densities $\{\phi_j\}$ located at $\{z_j\}$, with $|z_j - z_\mathrm{C}| < r$, then for any $|z - z_\mathrm{C}| > R$, the induced potential $q(z)$ can be approximated by:

$$q(z) = a_0 log(z - z_\mathrm{C}) + \sum_{k=1}^{p} \frac{a_k}{(z - z_\mathrm{C})^k} = \mathcal{O}\left(\frac{r^p}{R^p}\right), \tag{B.1}$$

where $\{a_k,\ 0 \leqslant k \leqslant p\}$ satisfies

$$a_0 = \sum_{j=1}^{m} \phi_j \quad \text{and} \quad a_k = \sum_{j=1}^{m} \frac{-\phi_i (z_i - z_\mathrm{C})^k}{k}.$$

Local expansion. Suppose the m source densities $\{\phi_j\}$ located at $\{z_j\}$, with $|z_j - z_\mathrm{C}| > R$, then for any $|z - z_\mathrm{C}| < r$, the induced potential $q(z)$ can be approximated by:

$$q(z) = \sum_{k=0}^{p} c_k (z - z_\mathrm{C})^k + \mathcal{O}\left(\frac{r^p}{R^p}\right), \tag{B.2}$$

where $\{c_k,\ 0 \leqslant k \leqslant p\}$ satisfies

$$c_0 = \sum_{j=1}^{m} \phi_j \log(z_\mathrm{C} - z_j) \quad \text{and} \quad c_l = \sum_{j=1}^{m} \frac{-\phi_j}{l(z_j - z_\mathrm{C})^l}.$$

M2M translation. Suppose z_C is the center of a box and z_M is the center of its parent. Suppose further $\{a_k\}$ is the multipole expansion at z_C, then the multipole expansion at z_M can be written as:

$$q(z) = b_0 \log(z - z_\mathrm{M}) + \sum_{l=1}^{p} \frac{b_l}{(z - z_\mathrm{M})^l} + \mathcal{O}(\epsilon), \tag{B.3}$$

where $\{b_k,\ 0 \leqslant k \leqslant p\}$ satisfies

$$b_0 = a_0 \quad \text{and} \quad b_l = -\frac{a_0 (z_\mathrm{C} - z_\mathrm{M})^l}{l} + \sum_{k=1}^{l} a_k (z_\mathrm{C} - z_\mathrm{M})^{l-k} \binom{l-1}{k-1}.$$

M2L translation. Suppose z_M and z_L are the centers of two non-adjacent boxes on the same level, $\{b_k\}$ is multipole expansion at z_M. Then the local expansion at z_L transformed from $\{b_k\}$ is:

$$q(z) = \sum_{l=0}^{p} c_l (z - z_{\mathrm{L}})^l + \mathcal{O}(\epsilon), \tag{B.4}$$

where $\{c_k,\ 0 \leqslant k \leqslant p\}$ satisfies

$$c_0 = b_0 \log(z_{\mathrm{L}} - z_{\mathrm{M}}) + \sum_{k=1}^{p} \frac{b_k}{(z_{\mathrm{M}} - z_{\mathrm{L}})^k}(-1)^k,$$

$$c_l = -\frac{b_0}{l(z_{\mathrm{M}} - z_{\mathrm{L}})^l} + \frac{1}{(z_{\mathrm{M}} - z_{\mathrm{L}})^l} \sum_{k=1}^{p} \frac{b_k}{(z_{\mathrm{M}} - z_{\mathrm{L}})^k} \binom{l+k-1}{k-1}(-1)^k.$$

L2L translation. Suppose z_{T} is the center of a box and z_{L} the center of its parent. Suppose further $\{c_l\}$ is the local expansion at z_{L}, then the local expansion at z_{T} can be written as

$$q(z) = \sum_{l=0}^{p} d_l (z - z_{\mathrm{T}})^l + \mathcal{O}(\epsilon), \tag{B.5}$$

where $\{d_k,\ 0 \leqslant k \leqslant p\}$ satisfies

$$d_l = \sum_{k=1}^{p} c_k \binom{k}{l} (z_{\mathrm{T}} - z_{\mathrm{L}})^{(k-l)}.$$

Appendix C. Proofs of lemmas

Proof of Lemma 4.1. First, we prove the compactness of K_i and E_i. Since $\mathbf{y}_i$ and $\mathbf{x}_i$ are disjoint, the kernel G in K_i is C^∞ in both variables. Thus, K_i, as a convolution operator with C^∞ kernel, is compact in H^t norm. E_i is also compact in H^t norm since $\mathbf{y}_i$ is disjoint from $\mathbf{x}_{i+1}$.

Now, we prove that L_i is compact in H^t norm (see Fig. 11 for the domains involved). Suppose $q_i \in S_i$ on $\mathbf{x}_i$, we can find $\phi_i \in H^t$ on $\mathbf{y}_i$, such that $\phi_i = K_i^+ q_i$.

Since $\mathscr{K}[\mathbf{y}_i \to \mathbf{x}_i](\phi_i) = K_i\phi_i = K_i K_i^+ q_i = q_i = q_i$, $\mathscr{K}[\mathbf{y}_i \to D_i](\phi_i)$ is the solution of boundary value problem on domain D_i with boundary condition q_i. On the other hand, $q_{i+1} = E_i(\phi_i) = \mathscr{K}[\mathbf{y}_i \to \mathbf{x}_{i+1}]$ is the solution of this problem on $\mathbf{x}_{i+1}$. Hence, $L_i(q_i) = E_i K_i^+(q_i) = E_i(\phi_i) = q_{i+1}$ is equivalent to the Poisson

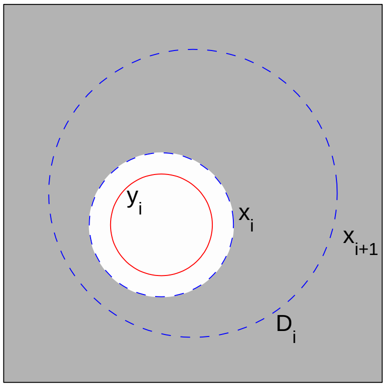


Fig. 11. The domains used in the proof of Lemma 4.1 where L_i corresponds to a M2M translation. The grayed region is D_i.

formula which evaluates the potential at $\mathbf{x}_{i+1}$ from the potential at $\mathbf{x}_i$. The kernel in Poisson formulae, which corresponds to the fundamental solution of the PDE with domain D_i, is C^∞ smooth since $\mathbf{x}_i$; and $\mathbf{x}_{i+1}$ are disjoint. This means that the Poisson formulae represents a compact operator in H^t norm for any t. Therefore, L_t is a compact operator in H^t norm. □

To clarify the idea behind the proof, we give the analytic form of the M2M translation operator for a simplified case for the single layer potential for the 2D Laplacian. The main reason for the compactness is the inclusion of $\mathbf{x}_i$, in $\mathbf{x}_{i+1}$.

We assume that the three surfaces $\mathbf{y}_i$, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are concentric circles such that their radii ρ_i^e, ρ_i^c and ρ_{i+1}^c satisfy the condition $0 \leqslant \rho_i^e \leqslant \rho_i^c \leqslant \rho_{i+1}^c$.

Standard logarithm expansion and simple algebraic manipulations yield

$$\log|\mathbf{x}-\mathbf{y}| = \log|\mathbf{x}| + \sum_{k=-\infty,k\neq 0}^{\infty} \frac{(-1)^k}{|k|}\left(\frac{|\mathbf{y}|}{|\mathbf{x}|}\right)^{|k|} \mathrm{e}^{\mathrm{i}k\theta_x}\mathrm{e}^{-\mathrm{i}k\theta_y}, \tag{C.1}$$

where θ_x and θ_y are the polar coordinate angles of the position vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. If we assume that this kernel acts on the space of continuous periodic functions in $[0, 2\pi]$ with zero mean and we can drop the $\log|\mathbf{x}|$ term. As the trigonometric functions are orthogonal on $L^2(0, 2\pi)$, the above expression is a diagonalization of the single layer operator. As the eigenvalues are all positive, they coincide with singular values.

First, we solve $K_i\phi_i = q_i$. In this case, since $|\mathbf{x}| = p_i^c \geqslant |\mathbf{y}| = \rho_i^e$, the singular values decay exponentially, so the problem $K_i\phi_i = q_i$ is ill-posed: small perturbations on the high frequency components of q^c get exponentially amplified. However, since q_i; is the potential induced by the densities in the interior of $\mathbf{y}_i$, ϕ_i is a well-defined function with the following relationship on Fourier coefficients:

$$\widehat{\phi}_i(k) = (-1)^k|k|\left(\frac{\rho_i^c}{\rho_i^e}\right)^{|k|}\widehat{q}_i(k).$$

Second, we evaluate q_{i+1} with $E_i\phi_i$. The Fourier coefficients of q_{i+1} are given by

$$\hat{q}_{i+1}(k) = \frac{(-1)^k}{|k|}\left(\frac{\rho_i^e}{\rho_{i+1}^e}\right)^{|k|} \quad \hat{\phi}_i(k) = \left(\frac{\rho_i^c}{\rho_{i+1}^c}\right)^{|k|}\hat{q}_i(k).$$

This expression actually gives the singular value decomposition of L_i using the trigonometric basis on the circle, where $\left(\rho_i^c/\rho_{i+1}^c\right)^{|k|}$ are the singular values of L_i. The singular values decay exponentially to zero since $\rho_i^c < \rho_{i+i}^c$, therefore $L_i a$ compact operator (with analytic kernel).

Proof of Lemma 4.2. A product of $L^{(k)}$ terms each one being an L operator, represents a sequence of M2M translations followed by a M2L translation and followed by a sequence L2L translations. To prove the lemma, we only need to show the existence of uniform bounds for the cases where $L^{(k)}$ corresponds to a sequence of M2M translations or a sequence of L2L translations. Here, we prove the latter case. The proof for the other case is the same.

Suppose $L^{(k)}$ transforms $q^{A,\mathrm{d}}$ at $\mathbf{x}^{A,\mathrm{d}}$ of box A into $q^{B,\mathrm{d}}$ at $\mathbf{x}^{B,\mathrm{d}}$ of box B. Since $L^{(k)}$ only involves L2L translations, B is contained in A and it is k level deeper in the computation tree. Suppose A has halfwidth r, from Section 3.2, we know $\mathbf{x}^{A,\mathrm{d}}$ has radius $(\sqrt{2+d})r$ and $\mathbf{x}^{B,\mathrm{d}}$ for any box B is contained in a circle which is concentric to $\mathbf{x}^{A,\mathrm{d}}$ and has radius $(\sqrt{2+\frac{d}{2}})r$. Hence, $\mathbf{x}^{B,\mathrm{d}}$ is always away from $\mathbf{x}^{A,\mathrm{d}}$ by a distance $\frac{d}{2}r$, which is independent of k.

As we pointed out in the proof of Lemma 4.1, the transformation $L^{(k)}$ can be viewed in a different way: it is equivalent to the Poisson formulae which evaluates the potential at $\mathbf{x}^{B,\mathrm{d}}$ from the potential $q^{A,\mathrm{d}}$ at $\mathbf{x}^{A,\mathrm{d}}$. The H^t norm of the Poisson formula grows to infinity only when $\mathbf{x}^{B,\mathrm{d}}$ and $\mathbf{x}^{A,\mathrm{d}}$ approach to each other. In our case, since $\mathbf{x}^{B,\mathrm{d}}$ and $\mathbf{x}^{A,\mathrm{d}}$ are separated by a distance $\frac{d}{2}r$ which is independent of k, the norm of $L^{(k)}$ is bouned from above uniformly. □

References

[1] C.R. Anderson, An implementation of the fast multipole method without multipoles, SIAM Journal on Scientific and Statistical Computing 13 (4) (1992) 923–947.

[2] J. Barnes, P. Hut, A hirerachical $O(N \log N)$ force-calculation algorithm, Nature 324 (4) (1986) 446–449.

[3] C. Leonard Herman, Grid-multipole calculations, SIAM Journal on Scientific Computing 16 (5) (1995) 1082–1091.

[4] G. Biros, L. Ying, D. Zorin, The embedded boundary integral method for the unsteady incompressible Navier–Stokes equations. Technical Report TR2003-838, Courant Institute, New York University, 2002. Available from: <http://www.es.nyu.edu/csweb/Research/TechReports/TR2003-838/TR2003-838.pdf>.

[5] G. Biros, L. Ying, D. Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (1) (2003) 317–348.

[6] O.P. Bruno, L.A. Kunyansky, A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests, and applications, Journal of Computational Physics 169 (2001) 80–110.

[7] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, Journal of Computational Physics 155 (1999) 468–498.

[8] W.D. Elliott, J.A. Board, Fast fourier transform accelerated fast multi-pole algorithm, SIAM Journal on Scientific Computing 17 (2) (1996) 398–415.

[9] Y. Fu et al., A fast solution for three-dimensional many-particle problems of linear elasticity, International Journal for Numerical Methods in Engineering 42 (1998) 1215–1229.

[10] Y. Fu, G.J. Rodin, Fast solution method for three-dimensional Stoke-sian many-particle problems, Communications in Numerical Methods in Engineering 16 (2000) 145–149.

[11] Z. Gimbutas, F. Rokhlin, A generalized fast mulipole method for nonoscillatory kernels, SIAM Journal on Scientific Computing 24 (3) (2002) 796–817.

[12] L. Greengard, The Rapid Evaluation of Potential Fields in Particle Systems, MIT Press, Cambridge, MA, 1988.

[13] L. Greengard, J. Huang, A new version of the fast multipole method for screened Coulomb interactions in three dimensions, Journal of Computational Physics 180 (2002) 642–658.

[14] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, Journal of Computational Physics 73 (1987) 325–348.

[15] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numerica (1997) 229–269.

[16] S. Kapur, D.E. Long, IES_3: Efficient electrostatic electromagnetic simulation, IEEE Computational Science and Engineering 5 (4) (1998) 60–67.

[17] S. Kapur, J. Zhao, A fast method of moments solver for efficient parameter extraction of MCMs. Design Automation Conference (1997) 141–146.

[18] Rainer Kress, Linear Integral Equations, Applied Mathematical Sciences, Springer, 1999.

[19] J. Makino, Yet another fast multipole method without multipoles–pseudoparticle multipole method, Journal of Computational Physics 151 (1999) 910–920.

[20] K. Nabors, F.T. Korsmeyer, F.T. Leighton, J.K. White, Preconditioned, adaptive, multipole-accelerated interative methods for three-dimensional first-kind integral equations of potential theory, SIAM Journal on Scientific and Statistical Computing 15 (1994) 713–735.

[21] J.R. Phillips, J.K. White, A precorrected-FFT method for electorstatic analysis of complicated 3-D structures, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 16 (10) (1997) 1059–1072.

[22] V. Popov, H. Power, An $O(N)$ taylor sereis multipole boundary element method for three-dimensional elasticity problems, Engineering Analysis with Boundary Elements 25 (2001) 7–18.

[23] V. Rokhlin, Rapid solution of integral equations of classical potential theory, Journal of Computational Physics 60 (1983) 187–207.

[24] N. Yarvin, V. Rokhlin, Generalized Gaussian quadratures and singular value decompositions of integral operators, SIAM Journal on Scientific Computing 20 (2) (1998) 699–718.

[25] N. Yarvin, V. Rokhlin, An improved fast multipole algorithm for potential fields on the line, SIAM Journal on Numerical Analysis (1999) 629–666.

[26] K. Yoshida, N. Nishimura, S. Kobayashi, Application of fast multipole Galerkin boundary integral equation method to elastostatic crack problems in 3D, International Journal for Numerical Methods in Engineering 50 (3) (2001) 525–547.