An FMM for the Modal Green's Function

May 4, 2016

1   Introduction

This project examines the problem of constructing of a fast multipole method (FMM) algorithm for an important category of non-uniform density distributions: axisymmetric surfaces of revolution.

We will work in three-dimensional cylindrical coordinates (r, θ, z) such that a point in Cartesian coordinates (x, y, z) is represented by:

$$x = r \cos \theta$$
$$y = r \sin \theta$$
$$z = z$$

An axisymmetric, or rotationally symmetric, surface Γ is one that has symmetry along the θ-axis. The surface Γ is obtained by rotating a two-dimensional curve γ about the z axis. γ is called the generating curve like in Figure 1, e.g. In particular, $\Gamma = \gamma \times \mathbb{T}$ where $\mathbb{T}$ is the one-dimensional torus (circle) parametrized by $\theta \in (-\pi, \pi]$.

Figure 1: An axisymmetric bowl-shaped surface Γ and its generating curve γ.
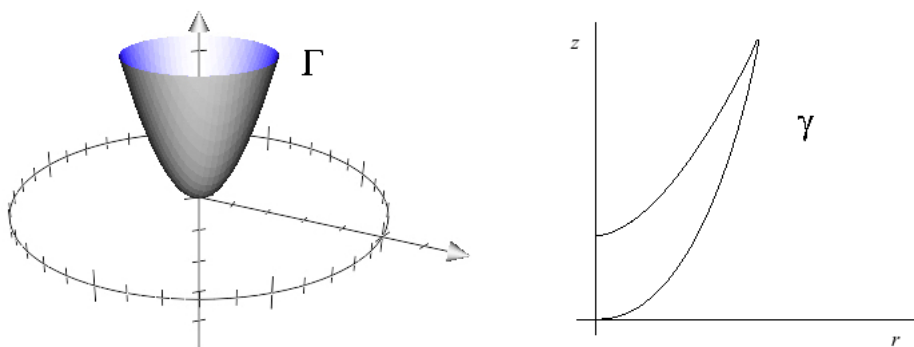
The need for a modal FMM arises from axisymmetric surfaces with difficult geometries where sources distributed in close proximity clump together and require a large number

of discretization points to be well-represented. For example, a circular generating curve like in Figure 2 is easy to discretize with a relatively low number of equally-spaced points. For a rectangular generating curve like in Figure 3, however, we need more discretization points in the corners due to density clumping. Similarly, for a bowl-shaped surface as in Figure 1, we would need a lot of discretization points near the vertex of the generating curve.

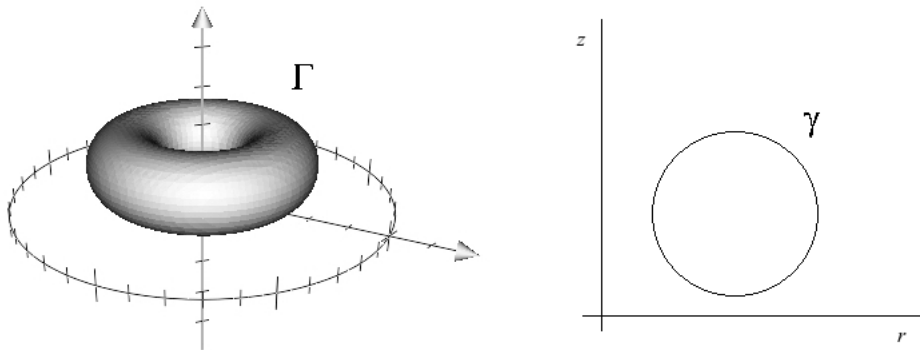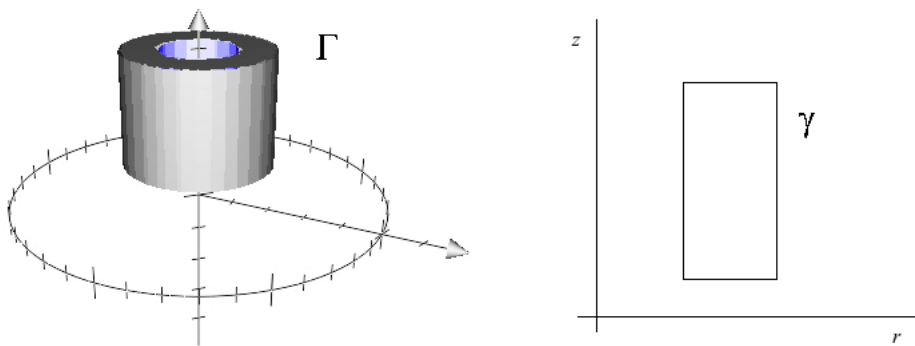Figure 2: An axisymmetric torus Γ and its circular generating curve γ.

Figure 3: An axisymmetric pipe surface Γ and its rectangular generating curve γ.

Density distributions on axisymmetric surfaces have many applications. The problem of electromagnetic scattering by a surface of revolution has radar, geophysical exploration, and acoustics applications. This type of problem relies precisely on creating a fast algorithm for an axisymmetric density distribution.

The organization of the paper is as follows: §2 states the problem, §3 gives the derivation of and motivation for using the Fourier representation of a Green's function, §4 gives an overview of fast multipole methods and discusses why existing techniques are not optimal for geometrically-challenging axisymmetric distributions, §5 details the proposed Cheby-

2

shev interpolation-based FMM to deal with this problem, and §6 describes future work to make this a complete and tested algorithm.

2 Statement of problem

Consider N source points $\mathbf{y}_j = (r'_j, \theta'_j, z'_j)$ distributed on this surface Γ, with each assigned a charge σ_j. We want to compute the potential f at target points $\mathbf{x}_i = (r_i, \theta_i, z_i)$ also on Γ due to the source densities. In practice, the same set of points act as both sources and targets. The potential is represented by the sum

$$f(\mathbf{x}_i) = \sum_{j=1}^{N} K(\mathbf{x}_i, \mathbf{y}_j)\sigma_j \tag{1}$$

where $K(\mathbf{x}, \mathbf{y})$ is a kernel, typically the Green's function of the partial differential equation that describes the potential between two points in the domain. In particular, we are interested in the free-space (no boundary conditions) Green's function for Laplace's equation, $\Delta u = 0$:

$$K(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|} \tag{2}$$

The goal is to construct a fast summation method for equation (1) that takes advantage of the rotational symmetry of Γ.

3 Modal Green's Function

The crucial technique to construct a fast method for an axisymmetric density distribution is to reduce the problem in three dimensions to a series of problems in two dimensions using Fourier analysis. In §3.1, we show the derivation of this Fourier representation for an arbitrary kernel, then for the Green's function of Laplace's equation. This strategy is grounded in the fact that it is easier to solve boundary integral equations defined on curves in $\mathbb{R}^2$ than those defined on surfaces in $\mathbb{R}^3$.

We are able to reduce the problem in this way because the rotationally symmetric nature of Γ yields $K(\mathbf{x}, \mathbf{y})$ symmetric along the θ-axis such that the kernel is a function of the difference $\theta - \theta'$:

$$K(\mathbf{x}, \mathbf{y}) = K(\theta - \theta', r, z, r', z')$$

We call such a kernel rotationally invariant.

Recall that if for two any points $\mathbf{x}$ and $\mathbf{y}$ in the computational domain a kernel can be written

$$K(\mathbf{x}, \mathbf{y}) = K(|\mathbf{x} - \mathbf{y}|)$$

then it is called translation invariant. Notice that the Green's function for Laplace's equation is translation invariant. Not all kernels are translation invariant, however, and the fact that what we'll define in §3.1 as the modal Green's function is not translation invariant is crucial to the difficulty of creating an optimal FMM in this scenario.

3.1 Fourier representation of 3D integral equations

This follows closely [10]. Consider the Fredholm integral equation of the first kind defined on Γ:

$$f(\mathbf{x}) = \int_{\Gamma} K(\mathbf{x}, \mathbf{y}) \sigma(\mathbf{y}) d\mathbf{y} \qquad\qquad \mathbf{x}, \mathbf{y} \in \Gamma \qquad (3)$$

This is a 3D integral equation, essentially a continuous version of equation (1), which can just be viewed as a discretization of this integral equation.

We can represent (3) as a sequence of 2D integral equations defined on the generating curve γ by performing Fourier transformations on f, σ, and K.. If f_m, σ_m, and k_m are the Fourier modes of f, σ, and K, respectively, then we have

$$f_m(r, z) = \int_{\mathbb{T}} \frac{e^{-im\theta}}{\sqrt{2\pi}} f(r, z, \theta) d\theta \qquad\qquad f(\mathbf{x}) = \sum_{m \in \mathbb{Z}} \frac{e^{im\theta}}{\sqrt{2\pi}} f_m(r, z)$$

$$\sigma_m(r, z) = \int_{\mathbb{T}} \frac{e^{-im\theta}}{\sqrt{2\pi}} \sigma(r, z, \theta) d\theta \qquad\qquad \sigma(\mathbf{x}) = \sum_{m \in \mathbb{Z}} \frac{e^{im\theta}}{\sqrt{2\pi}} \sigma_m(r, z)$$

$$k_m(r, z, r', z') = \int_{\mathbb{T}} \frac{e^{-im\theta}}{\sqrt{2\pi}} K(r, z, r', z', \theta) d\theta \qquad K(\mathbf{x}, \mathbf{y}) = \sum_{m \in \mathbb{Z}} \frac{e^{im(\theta - \theta')}}{\sqrt{2\pi}} k_m(r, z, r', z')$$

such that

$$\sqrt{2\pi} \int_{\gamma} k_m(r, z, r', z') \sigma_m(r', z') r' dl(r', z') = f_m(r, z) \qquad\qquad (r, z), (r', z') \in \gamma \qquad (4)$$

for each $m \in \mathbb{Z}$.

In practice, we choose a truncation parameter, $M \in \mathbb{N}$, such that f is well-represented by its lowest $2M + 1$ Fourier modes

$$||f - \sum_{m=-M}^{M} \frac{e^{im\theta}}{\sqrt{2\pi}} f_m|| \leq \epsilon \qquad\qquad (5)$$

and only compute the lowest $2M + 1$ Fourier modes for each of the functions f, σ, and K.

We are using the Green's function of a PDE as our kernel, so k_m is called the modal Green's function. In particular, we are interested in the Fourier expansion of the Green's

function for Laplace's equation

$$\frac{1}{4\pi|\mathbf{x}-\mathbf{y}|} = \frac{1}{4\pi\sqrt{r^2 + r'^2 - 2rr'\cos(\theta-\theta') + (z-z')^2}} \tag{6}$$

$$= \sum_{m\in\mathbb{Z}} \frac{e^{im(\theta-\theta')}}{\sqrt{2\pi}} s_m(r,z,r',z') \tag{7}$$

where the modal Green's function is

$$s_m(r,z,r',z') = \frac{1}{\sqrt{8\pi^3 rr'}} \mathbf{Q}_{m-\frac{1}{2}}\left(\frac{r^2 + (r')^2 + (z-z')^2}{2rr'}\right) \tag{8}$$

with $Q_{m-\frac{1}{2}}$ the half-integer order Legendre function of the second kind. Notice that while K was translation invariant, s_m is not. This has implications which we discuss later.

Now rather than constructing a fast summation for a density distribution on Γ using $K(\mathbf{x},\mathbf{y})$, we can construct a series of fast summations for a density distribution on γ using $s_m(r,z,r',z')$.

We should note that for now we are only considering the Green's function for Laplace's equation because its Fourier modes can be solved for analytically. For many other kernels, it is not possible to find the same type of explicit formula, and we would need to approximate the modes using a discretization as in [10].

We should also note that for densities distributed on the axisymmetric surface Γ the computational domain was a subset of $\mathbb{R}^3$. For the densities distributed on γ, however, we only need to consider the right half-plane of $\mathbb{R}^2$ as the computational domain.

In the next sections, we give an overview of the FMM, and discuss how several well-known fast methods approach the problem of axisymmetric density distributions.

4   Fast multipole methods

Fast multipole methods are a category of algorithms that were developed to speed the computation of the sum in equation (1). Equation (1) shows the naive computation of the potential. However, by representing the Green's function in a different way, we can accelerate this computation. This is precisely the advantage of fast multipoles methods. FMMs are fast approximation algorithms, with higher accuracy costing higher computational complexity. In general, an FMM reduces the complexity of this matrix-vector multiplication from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ or $\mathcal{O}(N\log N)$ for a given error level ϵ.

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.

There are two fundamental categories of FMMs: analysis-based algorithms that use analytic expansions for the multipole and local expansions, and kernel-independent methods that use equivalent densities to represent source densities.

4.1 Hierarchical tree structure

In 2D, the computational domain is a box containing all source and target points. This box is hierarchically partitioned into a quadtree. The box containing all sources and targets is called level 0 or the root level, with the next level of partitioning called level 1, and so on. The computational domain is partitioned until there are at most a pre-specified number of points in each box. Figure shows this process.

A box's parent is the box one level up that contains it. Similarly a box's children are the boxes one level down that it contains. For example in Figure , box 1 is the parent of boxes 4, and box 6 is a child of box 2. The near-neighbors of a box are the boxes on the same level with which it shares an edge or corner. If two boxes are not near-neighbors they are called well-separated. Each box has an interaction list, which is made up of boxes on the same level that are children of the parent's near-neighbors but are not near-neighbors. Figure shows this. Each box in 2D has at-most 27, $6^2 - 3^2$, boxes in its interaction list.

4.2 Analysis-based FMMs

The classical analysis-based FMM was developed by Greengard and Rokhlin using a multipole expansion for the Green's function of a system such that sources clumped together could be viewed as a single source to a far-field target, and a local expansion to represent the potential at a target due to all sources in the far-field, to accelerate the potential computation. now we also have linear algebra-based kernel-independent FMMs.

The classical FMM [5] first developed by Greengard and Rokhlin relies on analytic expansions for the potential. The multipole expansion for a box is the potential at some target in the far field due to sources in the box. The local expansion for a box is the potential at a target in the box due to all sources in the far field. They used Legendre polynomials and spherical harmonics to represent the kernel $1/\mathbf{r}$.

$$\frac{1}{|\mathbf{x} - \mathbf{y}|} = \sum \tag{9}$$

The three-dimensional problem is currently not possible as stated with an analytic FMM since we do not yet have multipole or local expansions for the Green's function for Laplace's equation in three-dimensional cylindrical coordinates.

We can consider an analytic FMM for the series of modal Green's functions, but again the analytic expansion has not been determined, and in fact, may not even exist. Our attempts to construct such expansions have so far failed. The necessary analytic machinery for the kernels s_n does not exist yet if at all, and is in fact, very inefficient to construct. In addition, this method will require the evaluation of special functions. This is costly.

For two dimensions, the single-layer Laplacian kernel is $K(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi} \log(|\mathbf{x} - \mathbf{y}|)$. It is convenient, however, to write $K(\mathbf{x}, \mathbf{y}) = Re(\log(z_x - z_y))$ where z_x and z_y are complex number corresponding to points $\mathbf{x}$ and $\mathbf{y}$ in 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 $\{a_k; 0 \leq k \leq p\}$, where

$$f(z) = a_0 \log(z - z_C) + \sum_{k=1}^{p} \frac{a_k}{(z - z_C)^k} + \mathcal{O}\left(\frac{r^p}{R^p}\right) \tag{10}$$

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 $\{c_k, 0 \leq k \leq p\}$, where

$$f(z) = \sum_{k=0}^{p} c_k(z - z_C)^k + \mathcal{O}\left(\frac{r^p}{R^p}\right) \tag{11}$$

In both expansions, p is usually a small constant determined by the desired accuracy of the result.

Instead of Laurent series, in three dimensions the far field is represented by spherical harmonics and Legendre polynomials.

4.3   Kernel-independent methods

linear algebraic and PDE(green's idnentities)

Since we don't have the analytic machinery required for an analysis-based FMM, we consider kernel-independent methods. These methods don't require the same analytic expansions, and instead use equivalent densities to represent the source densities. These methods can be separated into two categories. First, the not so kernel-independent methods which require that the kernel is the Green's function for some elliptic partial differential equation, or satisfies Green's identity. The most well-known method of this type is the kernel-independent FMM (KIFMM), which was proposed by Ying, Biros, and Zorin. Second are methods which work with any smooth kernel and require only its evaluation at some points. The most well-known method of this type is the black-box FMM (bbFMM) by Fong and Darve.

In the first type of method, we want to use the properties of the PDE to construct translation operators. There are three approaches we can use to represent the potential in the far-field: 1) use Green's third identity – differential equations, 2) use integral equations, or 3) create equivalent densities and potentials. 3) is the choice of YBZ, and further detail of the construction of these operators is explored below. Essentially, an algorithm of this type depends on the source to multipole translation, and choice of a discretization scheme.

4.4 Translation operators

The FMM employs these multipole and local expansions and densities recursively. 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 the analysis-based FMM, a translation operator translates a multipole or local expansion of one box to that of another in the computational domain using a function. In kernel-independent methods, translation operators translate equivalent densities in the same way but using linear operators or matrices.

To describe each operator below, we use the terminology for the analysis-based FMM (multipole and local expansions), but we could equivalently substitute upward and downward equivalent densities. In particular, five translations are used:

- $S2M$: The source to multipole operator creates a multipole expansion or density for a box.

- $M2M$: The multipole to multipole operator translates the multipole expansion or density of a box to that of its parent.

- $M2L$: The multipole to local operator translates the multipole expansion or density of a box to the local expansion or density of a box in its interaction list.

- $L2L$: The local to local operator translates the local expansion of a box to the local expansion of a child box.

- $L2T$: The local to target operator computes the far-field interaction contribution by evaluating the local expansion of a box at the target points in that box.

The sum of the contributions from the $M2L$ operation and $L2L$ operation for a box is called the local expansion.

These translation operators are the backbone of a fast multipole algorithm because they reduce the number of direct computations required to find the sum in equation (1). Translation operators can be computed without any knowledge of the source or target points, so they are a true pre-computation. The only thing required is a computational domain and choice of a number of level.

If a kernel is translation invariant, then translation operators will only differ based on relative position and level in the hierarchical tree. This is due to the the kernel's dependence only on the difference between coordinate values, e.g. ($|\mathbf{x} - \mathbf{y}|$). For example, with a translation invariant kernel the $M2L$ operation from the box $[0,1] \times [0,1]$ to $[0,1] \times [2,3]$ would be the same as the operation from $[1,2] \times [1,2]$ to $[1,2] \times [3,4]$. Therefore many translation operators on a given level are identical, and need not be computed repeatedly. In fact, each level has at most 40 $(7^2 - 3^2)$ unique transfer vectors.

On the other hand, if a kernel is not translation invariant, then every translation operator for every box in the computational domain needs to be computed. Every transfer vector is unique, so none can be reused or recycled as above. This is the case for the modal Green's function, and we need to take this more costly pre-computation into consideration when evaluating a fast method.

4.5   General multilevel algorithm structure

These methods, whether analysis-based or kernel-independent, involve the same general steps for a multilevel algorithm to accelerate the summing in equation (1). Fast multipole methods follow this general algorithmic scheme:

1. Hierarchically partition the computational using a quadtree of boxes with $L+1$ levels $0, \ldots, L$ until each box on the finest level L contains no more than s source points. The parameter s is chosen based on a desired accuracy level ϵ.

2. For each box on the finest level L, construct a multipole expansion or upward equivalent density using the $S2M$ operator.

3. For each box on levels $1, \ldots, L$, shift the multipole expansion of the box to the multipole expansion of the parent box using the $M2M$ operator. Steps 3 and 4 are referred to as the upward pass, and together they accumulate multipole expansions for every box in the computational domain.

4. For each box on levels $2, \ldots, L$, compute the contribution to the box's local expansion by translating the multipole expansions of boxes in its interaction list using the $M2L$ operator.

5. For each box on levels $0, \ldots, L-1$, translate the local expansion of the box to the local expansion of its child boxes using the $L2L$ operator. Steps 4 and 5 are referred to as the downward pass, and together they create a local expansion for every box in the computational domain. Note that the local expansion, which is the contribution from all far-field sources, for a box is the sum of contributions from its $M2L$ and $L2L$ operations.

6. Compute the total far-field contribution for each box by using the $L2T$ operator to evaluate its local expansion at the target points in the box.

7. Lastly, for each box, directly compute the contribution from near-field interactions (sources in near-neighbor boxes) and add them to the far-field contribution.

5 The KIFMM

Rather than using analytic multipole and local expansions like in the FMM, the KIFMM substitutes a continuous distribution of upward and downward equivalent densities that lie on surfaces surrounding each box in the hierarchical tree to represent the potential generated by sources in that box and the potential in that box due to sources in the far-field. 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.

$$\int_{\mathbf{y}^{B,u}} K(\mathbf{x}, \mathbf{y})\phi^{B,u}(\mathbf{y})d\mathbf{y} = \sum_{i \in I_s^B} K(\mathbf{x}, \mathbf{y}_i)\phi_i \text{ for all } \mathbf{x} \in \mathbf{x}^{B,u} \tag{12}$$

EXPLAIN EQUIVALENT DENSITIES BETTER. In this algorithm the equivalent of a multipole expansion is the potential at a far-field target due to an equivalent density lying on an equivalent surface surrounding the box of sources. The equivalent of a local expansion is the potential at a target due to the equivalent densities of all far-field source boxes. This allows us to construct an efficient FMM that only requires kernel evaluations. This method relies on Green's third identity. The idea is that for well-separated boxes (at least one box away), the actual naively-computed potential between sources and targets in these boxes is the same as the potential due to these equivalent densities. The target box can't tell the difference between the real source densities and the equivalent densities. One advantage of the KIFMM is that it is relatively easy to implement, since in general it applies to an arbitrary kernel that is the fundamental solution of some elliptic PDE. To change the kernel in the original FMM, one would need to develop analytic multipole and local expansions, which as aforementioned are difficult to find if they exist at all, and the requirement of special functions makes them inefficient to construct.

To avoid the difficulty required in the implementation of the 3D KIFMM, the technique of using the modal Green's functions described earlier and covered at length by Yao, Young, and Martinsson, in [10] is used – replacing the 3D integral equations required by the 3D KIFMM with their Fourier representations, sequences of 2D integral equations. In this way, the 3D KIFMM implementation is avoided in favor of repeatedly applying the 2D KIFMM. As mentioned earlier, this strategy is currently only applicable to the modal Green's function for Laplace's equation because we can analytically determine its Fourier modes.

One downside to this application is that the pre-computation of translation operators is not optimal due to the not translation invariant kernel. In an attempt to reduce this pre-computation, we tried to find constant proportionality between the operators. For example, we consider that perhaps the $M2L$ from $[0, 1] \times [0, 1]$ to $[2, 3] \times [0, 1]$ is in proportion to the translation from $[0, 1] \times [0, 1]$ to $[3, 4] \times [0, 1]$. However, these attempts failed and indeed since the translation operators in the KIFMM are simply kernel evaluations, we

wouldn't expect this relationship to exist because it doesn't hold for the not translation invariant kernel. So every translation operator for every box on every level would need to be pre-computed. This is computationally expensive because every operator involves the evaluation of the kernel.

It's important to note that since the translation operators rely on kernel evaluations, they will also differ for each mode of the Fourier expansion of the Green's function since we're essentially using a different kernel. That means the full algorithm will require the pre-computation of $2M + 1$ times the number of operators required for the computational domain, where M is the truncation parameter for the Fourier expansion. Another consequence of the kernel reliance for translation operators is the added cost due to the fact that these translations will depend on which mode of the Fourier expansion we are using. Essentially, the entire FMM process would need to be done separately $2M + 1$ times. This is in sharp contrast to the computational savings of the vectorization of this process that's possible in the black-box FMM, which we discuss next.

5.1 Application

The upward and downward formulation of the aforementioned equivalent densities and their translation are explained in detail in §3.2 and shown in Figures 3, 4, 5, and 6.

5.2 Full algorithm

Recall that in the 3D KIFMM, we needed to solve surface integral equations like:

$$\text{S2M:} \int_{\mathbf{y}^{B,u}} K(\mathbf{x}, \mathbf{y}) \phi^{B,u}(\mathbf{y}) d\mathbf{y} = \sum_{i \in I_s^B} K(\mathbf{x}, \mathbf{y}_i) \phi_i \text{ for all } \mathbf{x} \in \mathbf{x}^{B,u} \tag{13}$$

We can write (15) as a sequence of 2D equations, (16), by using the Fourier representation explained in §2.2. If this is unclear, equation (15) is to (16) as equation (1) is to (3). In calculations below, I will skip this derivation and just state the sequences of 2D equations.

Now that we have our kernel k_n, we proceed through the standard 2D KIFMM algorithm for the sources on the 2D generating curve γ **for each** $n \in [-N, -N + 1, \ldots, N]$ where N is the truncation parameter chosen earlier in (10). The KIFMM algorithm is very similar to the FMM algorithm described in [**?**], apart from how the equivalent densities are represented, and how the translation operators are computed. As mentioned earlier, rather than by multipole expansions, the potential due to sources in a box is matched to an equivalent density at discretization points on a surface enclosing the box. In 2D, these surfaces are circles with radii prescribed in [9]. To compute these equivalent densities, we will need to discretize several integral operators on different surfaces, which is explained in §3.2.2.

In the following equations, $\mathbf{x} = (r, z)$ and $\mathbf{y} = (r', z')$. Also, please note the seemingly out-of-place r' term under each integral, and recall that this is actually part of the integral operator K_n as in (3).

5.2.1 Equivalent densities

After partitioning the hierarchical tree with no more than a prescribed number of sources in each box, compute the upward equivalent density for each leaf box. Similar to the **S2M** step in the FMM, solving the following equation for $\phi^{B,u}$ gives the upward equivalent density for a box B in the KIFMM:

$$\text{S2M: } \int_{\mathbf{y}^{B,u}} k_n(\mathbf{x}, \mathbf{y})\phi_n^{B,u}(\mathbf{y})r'd\mathbf{y} = \sum_{i \in I_s^B} k_n(\mathbf{x}, \mathbf{y}_i)\phi_i r_i' \text{ for all } \mathbf{x} \in \mathbf{x}^{B,u} \quad (14)$$

$$\text{Discretized S2M: } M_n\phi_n^{B,u} = q_n^{B,u} \quad (15)$$

where in (17), M_n is the matrix of kernel evaluations of pairs of discretization points on the upward check surface and upward equivalent surface of a box B, $q_n^{B,u}$ is the upward check potential, and the densities at the actual source points in the box are ϕ_i. This is similar to (7), and this process is illustrated in Figure 3. Once the upward equivalent density is computed for each leaf box, we use translation operators to find the upward and downward equivalent densities for every other box.

Figure 4: Left: The upward check (blue) and equivalent (red) surfaces of a box used to compute the upward equivalent density due to source densities (black) in the box. Right: The downward check (blue) and equivalent (red) surfaces of a box used to compute the downward equivalent density due to sources densities (black) in the far field on the box. In the algorithm, we never actually directly compute a downward equivalent density.

5.2.2 Translation Operators

Translation operators limit the number of equivalent densities we need to compute directly by translating upward equivalent densities of the leaf boxes to upward and downward equivalent densities of all other boxes. In the KIFMM in particular, the translation operators in their discretized form are matrices that translate an equivalent density from one box to that of another. They are a pre-computation, as they are constant regardless of the source distribution.

In the previous step, we computed the upward equivalent density for each leaf box. The **M2M** operator translates the upward equivalent density from a leaf box A to the upward

equivalent density of its parent box B. Solving the following equation for $\phi_n^{B,u}$ gives the M2M operator.

$$\text{M2M: } \int_{\mathbf{y}^{B,u}} k_n(\mathbf{x}, \mathbf{y})\phi_n^{B,u}(\mathbf{y})r'd\mathbf{y} = \int_{\mathbf{y}^{A,u}} k_n(\mathbf{x}, \mathbf{y})\phi_n^{A,u}(\mathbf{y})r'd\mathbf{y} \text{ for all } \mathbf{x} \in \mathbf{x}^{B,u}$$

(16)

Discretized M2M: $M_n^B \phi_n^{B,u} = M_n^A \phi_n^{A,u}$ (17)

where M_n^A is the matrix of kernel evaluations of pairs of discretization points on the upward check surface of box B and the upward equivalent surface of box A, and M_n^B is the matrix of kernel evaluations of pairs of discretization points on the upward check surface of box B and the upward equivalent surface of box B. Figure 4 gives a graphical representation of this step.

So the M2M translation operator is:

$$T^{M2M} = (M_n^B)^{-1} M_n^A \tag{18}$$

$$= \begin{pmatrix} K(\mathbf{x_1}, \mathbf{y_1}) & \cdots & K(\mathbf{x_1}, \mathbf{y_m}) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x_m}, \mathbf{y_1}) & \cdots & K(\mathbf{x_m}, \mathbf{y_m}) \end{pmatrix}^{-1} \begin{pmatrix} K(\mathbf{x_1}, \mathbf{y_1}) & \cdots & K(\mathbf{x_1}, \mathbf{y_m}) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x_m}, \mathbf{y_1}) & \cdots & K(\mathbf{x_m}, \mathbf{y_m}) \end{pmatrix} \tag{19}$$

Figure 5: The upward equivalent density due to the source densities (black) in the child box computed from the upward equivalent (red) and check (blue) surfaces of the child box is translated to the upward equivalent density of the parent box, checking against its upward check (blue) and equivalent (red) surfaces.

After repeating that step, every box now has an upward equivalent density. The **M2L** operator translates the upward equivalent density from a non-leaf box A to the downward equivalent density of a box B on the same level.

$$\text{M2L: } \int_{\mathbf{y}^{B,d}} k_n(\mathbf{x}, \mathbf{y})\phi_n^{B,d}(\mathbf{y})r'd\mathbf{y} = \int_{\mathbf{y}^{A,u}} k_n(\mathbf{x}, \mathbf{y})\phi_n^{A,u}(\mathbf{y})r'd\mathbf{y} \text{ for all } \mathbf{x} \in \mathbf{x}^{B,d}$$

(20)

Discretized M2L: $M_n^B \phi_n^{B,d} = M_n^A \phi_n^{A,u}$ (21)

where M_n^A is the matrix of kernel evaluations of pairs of discretization points on the downward check surface of box B and the upward equivalent surface of box A, and M_n^B is the matrix of kernel evaluations of pairs of discretization points on the downward check surface of box B and the downward equivalent surface of box B.

13

Figure 6: The upward equivalent density due to the source densities (black) in a box computed from the upward equivalent (red) and check (blue) surfaces of the child box is translated to the downward equivalent density of another box on the same level, checking against its upward check (blue) and equivalent (red) surfaces.

After repeating that step, every non-leaf box has a downward equivalent density. The **L2L** operator translates the downward equivalent density of a non-leaf box A to the downward equivalent density of a child box B.

$$\text{L2L: } \int_{\mathbf{y}^{B,d}} k_n(\mathbf{x}, \mathbf{y}) \phi_n^{B,d}(\mathbf{y}) r' d\mathbf{y} = \int_{\mathbf{y}^{A,d}} k_n(\mathbf{x}, \mathbf{y}) \phi_n^{A,d}(\mathbf{y}) r' d\mathbf{y} \text{ for all } \mathbf{x} \in \mathbf{x}^{B,d}$$

(22)

$$\text{Discretized L2L: } M_n^B \phi_n^{B,d} = M_n^A \phi_n^{A,d}$$

(23)

where M_n^A is the matrix of kernel evaluations of pairs of discretization points on the downward check surface of box B and the downward equivalent surface of box A, and M_n^B is the matrix of kernel evaluations of pairs of discretization points on the downward check surface of box B and the downward equivalent surface of box B. After repeating this step, every box now has an upward and downward equivalent density.

Figure 7: The downward equivalent density due to the source densities (black) in a parent box computed from the upward equivalent (red) and check (blue) surfaces of the parent box is translated to the downward equivalent density of a child box, checking against its upward check (blue) and equivalent (red) surfaces.

Once we have these upward and downward equivalent densities for every $n \in [-N, \dots, N]$, then we can reconstruct the 3D equivalent densities ϕ_{approx} by summing as in (10). This summing can be accelerated via the FFT. Having these equivalent densities in 3D, we can complete the last step of the KIFMM algorithm by evaluating the far- and then near-field interactions.

Note that all requirements for smoothness and uniqueness of the solution to the integral equations in the KIFMM listed in §3.1.5 of [9] (mostly pertaining to the sizes and positions of the surfaces) are satisfied here.

5.2.3 Discretization details

The equations in §3.2.1 are discretized using p points on the equivalent and check surfaces. p is constant for all boxes, and the choice of p determines the error level of the computa-

tions.This discretization requires two steps. First, the right hand side is the evaluation of a check potential. This step checks that the potential represented by the equivalent density and the actual source densities are the same to all boxes in the far field. Then on the left hand side, we invert a Dirichlet-type boundary integral equation to obtain the equivalent density. Essentially, each translation is simply applying a series of matrices.

For example, the M2L operator is the matrix T_n^{M2L} for a mode n is obtained by solving (20):

$$\phi_n^{B,d} = \left[\left[\alpha I + (M_n^B)^T M_n^B \right] (M_n^B)^T M_n^A \right] \phi_n^{A,u} \tag{24}$$

$$\implies T_n^{M2L} = \left[\left[\alpha I + (M_n^B)^T M_n^B \right] (M_n^B)^T M_n^A \right] \tag{25}$$

6 The black-box FMM

The optimal scheme for axisymmetric density distributions is the black-box FMM. The black-box FMM is a Chebyshev interpolation-based $\mathcal{O}(N)$ algorithm for non-oscillatory kernels. In general, this algorithm uses equivalent densities in each box placed at Chebyshev nodes. In this section, we consider this scheme to quickly compute this sum for each modal Green's function

$$f_m(\mathbf{x}_i) = \sum_{j=1}^{N} s_m(\mathbf{x}_i, \mathbf{y}_j) \sigma_{m,j} \tag{26}$$

for targets $\mathbf{x}_i$ and sources $\mathbf{y}_j$.

The bbFMM is good for complicated analytic kernels. Unlike the FMM or KIFMM, it requires only a smooth kernel, and only uses kernel evaluations at specific points. Another advantage for this method is that it has a small pre-computation even for large systems. The pre-computation for any kernel is $\mathcal{O}(N)$. This is a crucial advantage for the bbFMM in this problem since the pre-computation for the KIFMM varied based on the kernel evaluations and so was expensive because of the not translation invariant kernel. Another important advantage for the bbFMM is that only the $M2L$ operation requires an evaluation of the kernel. This means that the other four operations do not change based on which mode of the Fourier expansion of the Green's function, so they need not be computed separately for each mode.

The full method combines the methods of SVD compression from...

$$s_m(\mathbf{x}, \mathbf{y}) = \sum_i \alpha_i u_i(\mathbf{x}) v_i(\mathbf{y}) \tag{27}$$

and Chebyshev interpolation as in....

$$s_m(\mathbf{x}, \mathbf{y}) = \sum_i \sum_j s_m(\mathbf{x}_i, \mathbf{y}_j) w_i(\mathbf{x}) w_j(\mathbf{y}) \tag{28}$$

First we need to define the nth Chebyshev polynomial

$$T_n(x) = \cos\left(n \arccos\left(\frac{2}{b-a}\left(x - \frac{a+b}{2}\right)\right)\right) \tag{29}$$

on $[a, b]$. The Chebyshev nodes, which are the n roots of the Chebyshev polynomial, on $[a, b]$ are

$$x = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{2i-1}{2n}\pi\right) \text{ for } i = 1, \ldots, n. \tag{30}$$

in one dimension. For two dimensions, there are n^2 Chebyshev nodes that are the tensor product of the Chebyshev nodes of each interval on the two axes r and z. The nodes in the box these intervals create are all of the possible (r, z) pairs of the nodes in each respective interval, making n^2 nodes in each box.

From the Chebyshev polynomial Fong and Darve derive the optimal (why?) interpolation functions

$$R_n(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{n} + \frac{2}{n}\sum_{i=1}^{n-1} T_n(r)T_n(z)\right)\left(\frac{1}{n} + \frac{2}{n}\sum_{i=1}^{n-1} T_n(r')T_n(z')\right) \tag{31}$$

for $\mathbf{x} = (r, z)$ and $\mathbf{y} = (r', z')$.

From there we use a low-rank approximation for the mth modal Green's function given by

$$s_m(\mathbf{x}, \mathbf{y}) = \sum_{l_1=1}^{n}\sum_{l_2=1}^{n}\sum_{m_1=1}^{n}\sum_{m_2=1}^{n} s_m(\overline{\mathbf{x}}_{l_1,l_2}, \overline{\mathbf{y}}_{m_1,m_2})R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x})R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}) \tag{32}$$

to approximate the sum

$$f_m(\mathbf{x}_i) = \sum_{l_1=1}^{n}\sum_{l_2=1}^{n} R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}_i) \sum_{m_1=1}^{n}\sum_{m_2=1}^{n} s_m(\overline{\mathbf{x}}_{l_1,l_2}, \overline{\mathbf{y}}_{m_1,m_2}) \sum_{j=1}^{N} \sigma_{m,j} R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j) \tag{33}$$

where N is the number of source points, n^2 is the number of Chebyshev nodes in each box, $\mathbf{y}_j$ are the source points, $\mathbf{x}_i$ are the target points, $\overline{\mathbf{x}}_{l_1,l_2}$ are the Chebyshev nodes in the target box, and $\overline{\mathbf{y}}_{m_1,m_2}$ are the Chebyshev nodes in the source box.

This low-rank approximation yields the translation operators for this method. For a given mode $n = m$, we detail each operation.

6.1   S2M

The source-to-multipole operation computes weights $W^{m,B}_{m_1,m_2}$ at each Chebyshev node $\overline{\mathbf{y}}_{m_1,m_2}$ by anterpolation in a box B on the finest level.

$$W^{m,B}_{m_1,m_2} = \sum_{\mathbf{y}_j \in B} \sigma_{m,j} R_n(\overline{\mathbf{y}}^B_{m_1,m_2}, \overline{\mathbf{y}}_j) \tag{34}$$

for $m_1, m_2 = 1, \ldots, n$.

6.2 M2M

The multipole-to-multipole operation translates weights at the nodes of four child boxes B_i, W_{m_1,m_2}^{m,B_i}, to weights at the nodes of its parent box A, $W_{m_1,m_2}^{m,A}$.

$$W_{m_1,m_2}^{m,A} = \sum_{i=1}^{4} \sum_{m_1'=1}^{n} \sum_{m_2'=1}^{n} W_{m_1',m_2'}^{m,B_i} R_n(\overline{\mathbf{y}}_{m_1,m_2}^{A}, \overline{\mathbf{y}}_{m_1',m_2'}^{B_i}) \tag{35}$$

for $m_1, m_2 = 1, \ldots, n$.

6.3 M2L

The multipole-to-local operation computes the far-field contribution at Chebyshev nodes $\overline{\mathbf{x}}_{l_1,l_2}^{A}$ in box A from all boxes B_i in the interaction list of A.

$$g_{l_1,l_2}^{m,A} = \sum_{B_i} \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W_{m_1,m_2}^{m,B_i} s_m(\overline{\mathbf{x}}_{l_1,l_2}^{A}, \overline{\mathbf{y}}_{m_1,m_2}^{B_i}) \tag{36}$$

for $l_1, l_2 = 1, \ldots, n$.

6.4 L2L

On the root level let $f_{l_1,l_2}^{m,A} = g_{l_1,l_2}^{m,A}$ and use the local-to-local operation to obtain the full local expansion by adding the far-field contribution from the parent box B

$$f_{l_1,l_2}^{m,A} = g_{l_1,l_2}^{m,A} + \sum_{l_1'=1}^{n} \sum_{l_2'=1}^{n} f_{l_1,l_2}^{m,B} R_n(\overline{\mathbf{x}}_{l_1,l_2}^{A}, \overline{\mathbf{x}}_{l_1',l_2'}^{B}) \tag{37}$$

for $l_1, l_2 = 1, \ldots, n$ where B is the parent box of box A.

6.5 L2T

The local-to-target operation computes $f_m(\mathbf{x}_i)$ for target point $\mathbf{x}_i$ in box B by interpolating the far-field approximation.

$$f_m(\mathbf{x}_i) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} f_{l_1,l_2}^{m,B} R_n(\overline{\mathbf{x}}_{l_1,l_2}^{B}, \mathbf{x}_i) \tag{38}$$

Notice that the $S2M$, $M2M$, $L2L$, and $L2T$ translation operators in this scenario do not require kernel evaluations. Since the interpolation functions R_n do not depend on which

mode of the Fourier representation of the Green's function, we only need to compute these operators once instead of $2M + 1$ times in the KIFMM.

In addition, for each mode m, the weights $W_{m_1,m_2}^{m,B}$ can be represented by a matrix with n^2 rows and M, the truncation parameter discussed earlier chosen for the Fourier representation, columns. Therefore these translation operations are just matrix-matrix multiplications which can be performed quickly in practice.

6.6 Vectorization

Vectorization of the bbFMM for modal Green's function. A key advantage of this method over the KIFMM is the fast pre-computation and indeed fast application of these operators. Not only can they all be computed at once since the interpolation function does not depend on the Fourier mode m, but the operators can be applied quickly as well. To see this, consider that $\sigma_{m,j}$ is an $N \times (2M + 1)$ matrix, where each row is the source point and each column is a Fourier mode, which we can apply via BLAS3 matrix-matrix multiplication to the interpolation function R_n. $W_{m_1,m_2}^{m,B}$, too, is then simply a matrix for each box with rows representing each Chebyshev node and each column representing a Fourier mode. We can now do all of these operations for EVERY MODE at once! This is a huge advantage over the KIFMM where we would need to separately complete the entire pre-computation and application process $2M + 1$ times for each node.

$$W^m = R_n \sigma_m$$

$$W_{all} = R_n \sigma_{all}$$

The $M2L$ operation deserves a little special attention since it does depend on the kernel s_m and is the most expensive pre-computation and operation. The computation of s_m, however, generates each of the $2M + 1$ Fourier modes all at once using the Fast Fourier Transform (FFT).

$$s_m = \int_0^{2\pi} \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|} e^{-im\theta'} d\theta' \tag{39}$$

for x far from y, use discrete FFT on —x-y— to get s_m for M2L. for x near y, use Johan Helsing or Martinsson and Miller's algorithm to compute $Q_{m-1/2}$, $Q_{m+1/2}$, $Q_{m+3/2}$, etc.

As previously mentioned, the evaluation of s_m at the Chebyshev nodes can be accelerated using any of the techniques described in [1], [8], and [11]. The efficient evaluation of modal Green's function for Laplace's equation, and the Q-function in particular, has been explored. The methods described there can be implemented to quickly evaluate s_m.

The full bbFMM also reduces the cost of the $M2L$ operation by using SVD compression. However, with a not translation invariant kernel some of these operations may not save anything since recyclable transfer vectors on each level was crucial to their argument.

7 Conclusion and Future Work

Complete the bbFMM algorithm for modal Green's function.

Appendices

A Legend

> N the number of source points in the computational domain
>
> m the index for each Fourier mode of the Green's function
>
> M the truncation parameter for the Fourier expansion of the Green's function
>
> n the number of Chebyshev nodes in each interval
>
> $\mathbf{x}_i$ target points
>
> $\overline{\mathbf{x}}_{l_1,l_2}$ Chebyshev nodes in the target box
>
> $\mathbf{y}_j$ source points
>
> $\overline{\mathbf{y}}_{m_1,m_2}$ Chebyshev nodes in the source box

B Implementation and Testing

Currently, we are able to produce and have tested all equivalent densities and translation operators with Python programs for the Chebyshev interpolation-based bbFMM. However, for now we are just using the kernel, $\log |\mathbf{x} - \mathbf{y}|$, instead of the modal Green's function for Laplace's equation. These functions are very similar, though, so we expect these tests to succeed with the modal Green's function as well.

Much of the future work to do is a full FMM implementation in Python. Key pieces of this will be accelerating the $M2L$ operation by efficiently evaluating the modal Green's function, using the vectorization discussed in §6, and using singular value decomposition (SVD) compression technique described in [2], [3], [4] and [7].

The following are descriptions of the computations done by each of our test programs.
2D − S2M.py

1. Put N sources in box A and assign charges of ± 1.

2. Put target in a box B in the far field.

3. For each Chebyshev node of A, assign a weight

$$W_{m_1,m_2} = \sum_{j=1}^{N} \sigma_j R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j)$$

4. Compute the potential at each Chebyshev node in a target box B

$$f_{l_1,l_2} = \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W_{m_1,m_2} \log(\overline{\mathbf{x}}_{l_1,l_2}, \overline{\mathbf{y}}_{m_1,m_2})$$

5. Compute the potential at the target by interpolation

$$f(\mathbf{x}_i) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} f_{l_1,l_2} R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}_i)$$

6. Compare the result with the naive computation of the potential

$$f(\mathbf{x}_i) = \sum_{j=1}^{N} \log(\mathbf{x}_i, \mathbf{y}_j) \sigma_j$$

2D − M2M

1. Put N sources in box A and assign charges of ± 1.

2. For each Chebyshev node of A, assign a weight

$$W_{m_1,m_2} = \sum_{j=1}^{N} \sigma_j R_k(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j)$$

3. Repeat step 2 for the four child boxes B_i of A. Note that there will be $\leq N$ sources in each child box.

4. Use the $M2M$ operation to compute

$$W_{m_1,m_2}^{A} = \sum_{i=1}^{4} \sum_{m_1'=1}^{n} \sum_{m_2'=1}^{n} R_n(\overline{\mathbf{x}}_{m_1,m_2}^{A}, \overline{\mathbf{x}}_{m_1',m_2'}^{B_i}) W_{m_1',m_2'}^{B_i}$$

5. Compare the weight at each node with the computation in step 2.

Note that at this point if the weights match, there is no need to make sure the potential estimates match the naive potential computation.

2D − M2L.py

Note that this is the case where there is no $L2L$ contribution to the local expansion of a box B, since the parent of B has no interaction list.

1. Put a target point in a box B.

2. Put N sources in each box I_i in the interaction list of B and assign charges of ± 1.

3. For each Chebyshev node of I_i, assign a weight

$$W^{I_i}_{m_1,m_2} = \sum_{j=1}^{N} \sigma_j^{I_i} R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j^{I_i})$$

4. Calculate the far-field contribution at the Chebyshev nodes of B

$$f_{l_1,l_2} = \sum_{I_i} \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W^{I_i}_{m_1,m_2} \log(\overline{\mathbf{x}}^{B}_{l_1,l_2}, \overline{\mathbf{y}}^{I_i}_{m_1,m_2})$$

5. Compute the potential at the target point by interpolation

$$f(\mathbf{x}_i) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} f_{l_1,l_2} R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}_i)$$

6. Compare the result with the naive computation of the potential

$$f(\mathbf{x}_i) = \sum_{I_i} \sum_{j=1}^{N} \log(\mathbf{x}_i, \mathbf{y}_j^{I_i}) \sigma_j^{I_i}$$

2D − L2L.py

Note that this is the case where there is no $M2L$ contribution to the local expansion of a target box B, since B has no interaction list.

1. Put a target point in a box B, with parent box P.

2. Put N sources in each box I_i in the interaction list of P and assign charges of ± 1.

3. For each Chebyshev node of I_i, assign a weight

$$W^{I_i}_{m_1,m_2} = \sum_{j=1}^{N} \sigma_j^{I_i} R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j^{I_i})$$

4. Calculate the far-field contribution at the Chebyshev nodes of the parent of B

$$f_{l_1,l_2} = \sum_{I_i} \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W_{m_1,m_2}^{I_i} \log(\overline{\mathbf{x}}_{l_1,l_2}^{P}, \overline{\mathbf{y}}_{m_1,m_2}^{I_i})$$

5. Compute the potential at the target point by interpolation

$$f(\mathbf{x}_i) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} f_{l_1,l_2} R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}_i)$$

6. Compare the result with the naive computation of the potential

$$f(\mathbf{x}_i) = \sum_{I_i} \sum_{j=1}^{N} \log(\mathbf{x}_i, \mathbf{y}_j^{I_i}) \sigma_j^{I_i}$$

Note that in the case where there is a contribution from both the $M2L$ and $L2L$ operations, the local expansion is simply the sum of these contributions. Also note that the efficacy of the $L2T$ computation is shown in the last step of the $S2M$, $M2L$, and $L2L$ computations.

References

[1] Adbelmageed, A., *Efficient Evaluation of Modal Green's Functions Arising in EM Scattering by Bodies of Revolution.* Progress in Electromagnetics Research, PIER 27, (2000), 337-356.

[2] Cheng, H., Gimbutas, Z., Martinsson, P.G., Rokhlin, V., *On the Compression of Low Rank Matrices.* SIAM Journal of Scientific Computing, 26(4), (2005), 1389-1404.

[3] Fong, W., Darve, E., *A black-box fast multipole method.* Journal of Computational Physics, 228, (2009), 8712-8725.

[4] Gimbutas, Z., Rokhlin, V., *A Generalized Fast Multipole Method for Nonoscillatory Kernels.* SIAM Journal of Scientific Computing, 24(3), (2002), 796-817.

[5] Greengard, L., Rokhlin, V., *A Fast Algorithm for Particle Simulations.* Journal of Computational Physics, 73, (1987), 325-348.

[6] Hao, S., Martinsson, P.G., Young, P., *An efficient and highly accurate solver for multi-body acoustic scattering problems involving rotationally symmetric scatterers.* Computers and Mathematics with Applications, 69, (2015), 304-318.

[7] Martinsson, P.G., Rokhlin, V., *An accelerated kernel-independent fast multipole method in one dimension.* SIAM Journal of Scientific Computing, Vol. 29, No. 3, (2007), 1160-1178.

[8] Vaessen, J.A.H.M., van Beurden, M.C., Tijhuis, A.G., *Accurate and Efficient Computation of the Modal Green's Function Arising in the Electric Field Integral Equation for a Body of Revolution.* IEEE Transactions on Antennas and Propagation, 60(7), (2012), 3294-3304.

[9] Ying, L., Biros, G., Zorin, D., *A kernel-independent adaptive fast multipole method algorithm in two and three dimensions.* Journal of Computational Physics, 196, (2004), 591-626.

[10] Young, P., Yao, S., Martinsson, P.G., *A high-order Nyström discretization scheme for boundary integral equations defined on rotationally symmetric surfaces.* Journal of Computational Physics, 231, (2012), 4142-4159.

[11] Yu, W.M., Fang, D.G., Cui, T.J., *Closed Form Modal Green's Functions for Accelerated Computation of Bodies of Revolution.* IEEE Transactions on Antennas and Propagation, 56(11), (2008), 3452-3461.