An FMM for the Modal Green's Function

May 4, 2016

1 Introduction

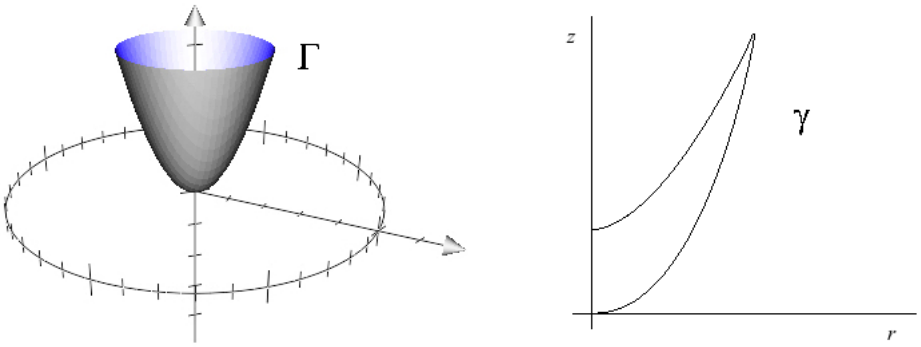
This project examines the problem of constructing of a fast multipole method (FMM) algorithm for an important category of non-uniform density distributions: axisymmetric surfaces of revolution.

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

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ z &= z \end{aligned}$$

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.

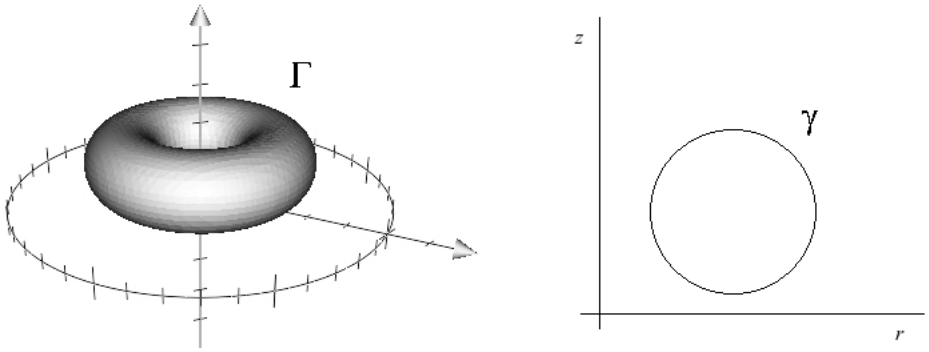
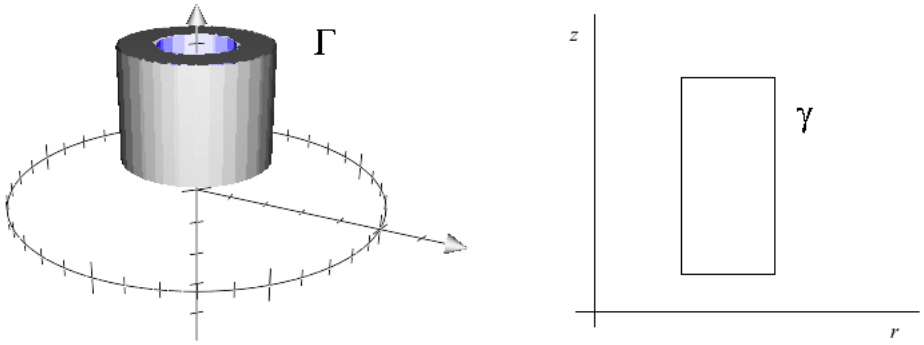


Figure 3: An axisymmetric pipe surface Γ and its rectangular generating curve γ.



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-

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 [7]. Consider the Fredholm integral equation of the first kind defined on Γ:

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

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

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

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

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

such that

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

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

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

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 [7].

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.

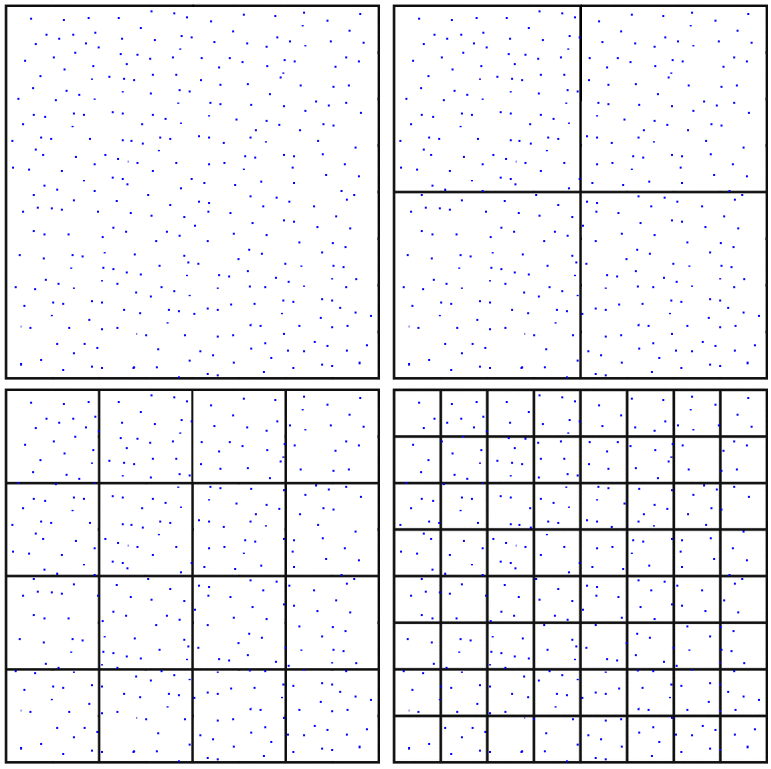
4 Fast multipole methods

Fast multipole methods are a category of algorithms that were developed to speed the computation of the potential in a system. Equation (1) shows the naive computation of the potential, which we can think of as a matrix-vector multiplication. In general, FMMs reduce the computational 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. FMMs are fast approximation algorithms, with higher accuracy costing higher computational complexity. The FMM gains this computational advantage by approximating the Green's function for the system in an efficient way. There are two fundamental categories of FMMs that differ in the way that they efficiently represent source densities: analysis-based algorithms that use analytic multipole and local expansions, and kernel-independent methods that use equivalent densities or weights.

4.1 Hierarchical tree structure

Before we get further into these two types of FMMs, we need to discuss the tree structure of the computational domain which allows us to perform multilevel FMMs. This structure is used for all FMMs. 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 4 shows the partitioning process.

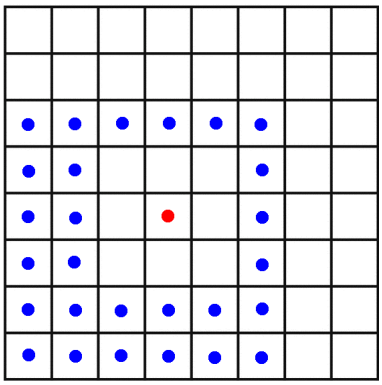
Figure 4: The hierarchical partitioning of the computational domain, levels 0 through 3, with sources and targets in blue.



We will use some terminology when discussing the tree, which we define now. 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 4, the only box on level 0 is the parent of all boxes on level 1. The near-neighbors of a box are the boxes on the same level with which it shares an edge or corner. In 2D, a box has at most 8 near-neighbors. 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 5 shows the interaction list of a box. Each box in 2D has at-most 27, $6^2 - 3^2$, boxes in its interaction list.

Figure 5: The blue boxes make up the interaction list of the red box.



In the analysis-based FMM, for each box, the potential induced by its source densities at a well-separated target is represented using a multipole expansion, while the potential induced by the sources from non-adjacent boxes is encoded in a local expansion. In kernel-independent methods, for each box, the potential induced by its source densities at a well-separated target is represented using an upward equivalent density, which the potential induced by the sources from non-adjacent boxes is encoded using a downward equivalent density. The basic idea is that for well-separated points, the naively-computed potential between sources and targets is the same as the potential due to the efficient density representation. The target points can't tell the difference between the actual source densities and the density representation, so we can take advantage of the fast that it's less computationally expensive to use these representations.

4.2 Analysis-based FMMs

The classical analysis-based FMM first developed by Greengard and Rokhlin in [4] relies on analytic expansions for the potential. As mentioned above, they used a multipole expansion 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.

As an example consider the 2D single-layer Laplacian kernel $K(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi}\log(|\mathbf{x}-\mathbf{y}|)$. It is convenient to write this as $K(\mathbf{x}, \mathbf{y}) = Re(\log(z_x - z_y))$, where z_x and z_y are complex number corresponding to points $\mathbf{x}$ and $\mathbf{y}$ in the plane. Now 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

$$f(z) = a_0 \log(z - z_C) + \sum_{k=1}^{p} \frac{a_k}{(z - z_C)^k} + \mathcal{O}\left(\frac{r^p}{R^p}\right) \tag{9}$$

This is the multipole expansion. 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

$$f(z) = \sum_{k=0}^{p} c_k (z - z_C)^k + \mathcal{O}\left(\frac{r^p}{R^p}\right) \tag{10}$$

This is the local expansion. In both expansions, p is usually a small constant determined by the desired accuracy of the result. For 3D applications, rather than Laurent series', the far field is represented by expansions using spherical harmonics and Legendre polynomials.

For our problem of a modal FMM for axisymmetric density distributions, however, an analysis-based FMM for the series of modal Green's functions is not possible because the necessary analytic machinery (multipole and local expansions) for the modal Green's function, s_n, has not been found yet, if it exists at all. Even if it does exist, it is inefficient to construct.

4.3 Kernel-independent methods

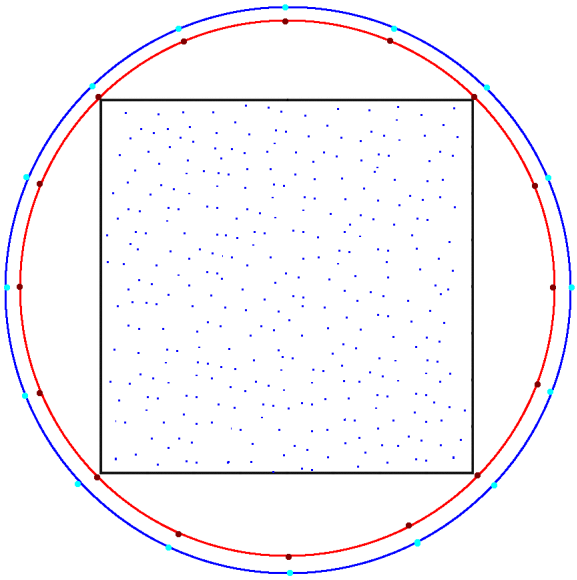
Since we don't have the analytic machinery required for an analysis-based FMM, we focus on kernel-independent methods to construct a modal FMM. One advantage of these methods is that they are relatively easy to implement, since in general they apply to an arbitrary kernel. To change the kernel in the classical FMM, we would need to develop analytic multipole and local expansions, which as aforementioned are difficult to find if they exist at all. These methods can be separated into two categories. The first requires that the kernel is the Green's function for some elliptic partial differential equation, satisfying Green's third identity. The most well-known method of this type is the kernel-independent FMM (KIFMM), which was constructed by Ying, Biros, and Zorin in [6]. Second are methods which work with any smooth kernel. The most well-known method of this type is the black-box FMM (bbFMM) by Fong and Darve in [2].

Kernel-independent methods use equivalent densities to represent the source densities rather than analytic expansions. As an example, in the KIFMM, an upward equivalent density for a box is a continuous distribution surrounding the box that matches the potential due to the box's source densities at a target in the far-field. This is analogous to a multipole expansion. We find this density, shown in Figure 6, by solving the following equation for $\sigma^{B,u}$.

$$\int_{\mathbf{y}^{B,u}} K(\mathbf{x}, \mathbf{y}) \sigma^{B,u}(\mathbf{y}) d\mathbf{y} = \sum_{j=1}^{N} K(\mathbf{x}, \mathbf{y}_j) \sigma_j \text{ for all } \mathbf{x} \in \mathbf{x}^{B,u} \tag{11}$$

To solve, the equation is discretized using points on the equivalent and check surfaces, shown in Figure 6. 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. This corresponds to the application of a matrix of kernel evaluations. Similarly, the downward equivalent density for a box is a continuous distribution surrounding the box that matches the potential at a target in the box due to all source densities in the far field. Again we solve a boundary integral equation using a discretization scheme. This is analogous to the local expansion in the analysis-based FMM. In this way, the KIFMM only requires kernel evaluations, although this is not always a positive attribute as we will see later.

Figure 6: The red circle is the upward equivalent surface with dark red discretization points. The blue circle is the upward check surface with light blue discretization points.



Now before we address the application of the KIFMM to a modal FMM on axisymmetric surfaces, we discuss translation operators and the general structure of the FMMs, as these two elements need to be covered to discuss the inefficiencies of the KIFMM in this scenario.

4.4 Translation operators

The FMM employs these multipole and local expansions or upward and downward equivalent densities recursively in a multilevel scheme. Translations between these efficient representations are what make FMMs $\mathcal{O}(N)$ algorithms.

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 expansions of functions. In kernel-independent methods, translation operators translate equivalent densities in the same way but with 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 analogously substitute upward and downward equivalent densities. In particular, five translations are used:

- $S2M$: The source to multipole operator creates a multipole expansion for a box.
- $M2M$: The multipole to multipole operator translates the multipole expansion of a box to that of its parent.
- $M2L$: The multipole to local operator translates the multipole expansion of a box to the local expansion 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.

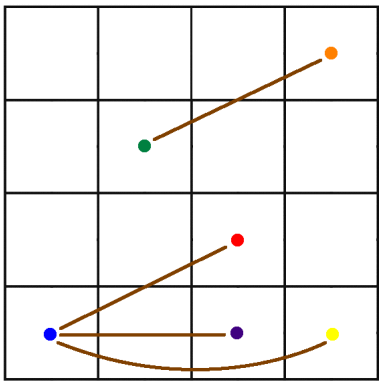
It's important to note that the sum of the contributions from the $M2L$ operation and $L2L$ operation make up the local expansion (or downward equivalent density) for a box.

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). The $M2M$, $M2L$, and $L2L$ translation operators can be computed without any knowledge of the source or target points, so they are a true pre-computation. The only requirement is a partitioned computational domain.

An important consideration in evaluating a fast method is the cost of pre-computing these operators. For translation invariant kernels, the translation operators will only differ based on relative position and level in the hierarchical tree. This is due to the fact that the kernel only depends on the difference between coordinate values, ($|\mathbf{x} - \mathbf{y}|$). For example, using Figure 7, with a translation invariant kernel the $M2L$ operator used to translate from the green box to the orange box would be the same one used to translate from the blue box to the red box. So in the case of translation invariant kernels, many $M2L$ 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 in this case.

On the other hand, for not translation invariant kernels, every translation operator for every box in the computational domain needs to be computed. Every transfer vector is unique, so none of them 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. In particular, this is a negative for the KIFMM, which we discuss later.

Figure 7: We use this figure to describe relationships between translation operators.



4.5 General multilevel algorithm structure

These methods, whether analysis-based or kernel-independent, use the same general steps for a multilevel algorithm to accelerate the summing in equation (1). Here we'll use terminology from kernel-independent methods (upward and downward equivalent densities), but we could analogously substitute the analysis-based FMM language. In 2D, FMMs follow this general algorithmic scheme.

1. Hierarchically partition the computational domain 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 an upward equivalent density using the $S2M$ operator.

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

4. For each box on levels $2, \ldots, L$, compute the contribution to the box's downward equivalent density by translating the upward equivalent densities of boxes in its interaction list using the $M2L$ operator.

5. For each box on levels $0, \ldots, L-1$, translate the downward equivalent density of the box to the downward equivalent densities of its child boxes using the $L2L$ operator. Steps 4 and 5 are referred to as the downward pass, and the summing of these operations creates a downward equivalent density for every box in the computational domain.

6. Compute the total far-field contribution for each box by using the $L2T$ operator to evaluate its downward equivalent density 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. This sum is the total potential at targets in the box.

4.6 Translations in the KIFMM

As explained in §4.3, rather than using analytic multipole and local expansions, 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 implement the KIFMM for axisymmetric density distributions, the technique of using the modal Green's functions described at length by Yao, Young, and Martinsson in [7] is used – replacing the 3D integral equations required by the 3D KIFMM with their Fourier representations, sequences of 2D integral equations. This way we avoid the 3D KIFMM 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.

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.

There are a few negatives to this possible application. One is that the pre-computation of translation operators is not optimal due to the not translation invariant kernel, the modal Green's function. As aforementioned, for not translation invariant kernels, we need to pre-compute all translation operators for every box in the computational domain. In an attempt to reduce this pre-computation, we tried to find constant proportionality between the operators. For example, using Figure 7 we consider that perhaps the $M2L$ operator used to translate from the blue box to the purple box is in proportion to the $M2L$ operator used to translate from the blue box to the yellow box. However, these attempts failed and indeed since the translation operators in the KIFMM are simply kernel evaluations, we wouldn't expect this proportionality to exist because it doesn't exist in the kernel. So every translation operator for every box on every level would need to be pre-computed.

Further adding to the computational cost of this method is that every operation involves the evaluation of the kernel. Essentially, each translation is simply applying a series of matrices that depend on the kernel, which is expensive. Since the translation operators rely on kernel evaluations, they will 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 operations 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, once for each mode of the kernel. 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 The black-box FMM

Now we discuss the black-box FMM, a kernel-independent a Chebyshev interpolation-based $\mathcal{O}(N)$ algorithm for non-oscillatory kernels. We will focus on the bbFMM for developing a modal FMM.

5.1 Advantages

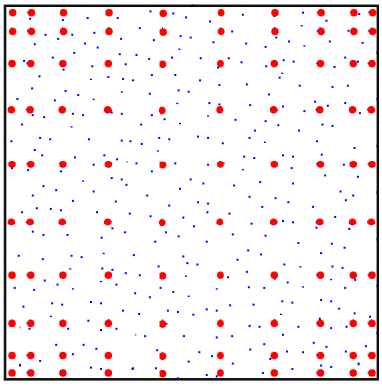
The advantages of this method are manifold. This is the optimal scheme for axisymmetric density distributions with difficult geometries because it is based on Chebyshev interpolation. In general, this algorithm uses equivalent densities in the form of weights in each box placed at Chebyshev nodes, which themselves clump in corners such that source densities located there will be well-represented. This is shown in Figure 8. The bbFMM is also good for complicated analytic kernels like the modal Green's function, s_m. 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.

5.2 Vectorization

In addition, for each mode m, the weights $W^{m,B}_{m_1,m_2}$ 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.

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

Figure 8: The bbFMM uses equivalent densities in the form of weights at the Chebyshev nodes in each box to represent the source densities. Here this is shown for $n^2 = 100$ Chebyshev nodes in red representing source densities in blue.



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. $W^{m,B}_{m_1,m_2}$, 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.

$$W^m = R_n \sigma_m$$

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

We consider this scheme to quickly compute this sum for each modal Green's function

$$f_m(\mathbf{x}_i) = \sum_{j=1}^{N} s_m(\mathbf{x}_i, \mathbf{y}_j)\sigma_{m,j} \tag{12}$$

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

To find these nodes, first we need to define the nth Chebyshev polynomial

$$T_n(x) = \cos\left(n \arccos\left(\frac{2}{b-a}\left(x - \frac{a+b}{2}\right)\right)\right) \tag{13}$$

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

$[a, b]$ are

$$x = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i-1}{2n}\pi\right) \text{ for } i = 1, \ldots, n. \tag{14}$$

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. Note that the intervals on each axis don't need to be the same.

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

$$R_n(\mathbf{x}, \mathbf{y}) = \left(\frac{1}{n} + \frac{2}{n}\sum_{i=1}^{n-1} T_n(r)T_n(z)\right)\left(\frac{1}{n} + \frac{2}{n}\sum_{i=1}^{n-1} T_n(r')T_n(z')\right) \tag{15}$$

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

The full method combines the methods of SVD compression from [3] and [**?**],

$$s_m(\mathbf{x}, \mathbf{y}) = \sum_i \alpha_i u_i(\mathbf{x}) v_i(\mathbf{y}) \tag{16}$$

and Chebyshev interpolation

$$s_m(\mathbf{x}, \mathbf{y}) = \sum_i \sum_j s_m(\mathbf{x}_i, \mathbf{y}_j) w_i(\mathbf{x}) w_j(\mathbf{y}) \tag{17}$$

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

$$s_m(\mathbf{x}, \mathbf{y}) = \sum_{l_1=1}^{n}\sum_{l_2=1}^{n}\sum_{m_1=1}^{n}\sum_{m_2=1}^{n} s_m(\overline{\mathbf{x}}_{l_1,l_2}, \overline{\mathbf{y}}_{m_1,m_2}) R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}) R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}) \tag{18}$$

to approximate the sum in equation (12) as

$$f_m(\mathbf{x}_i) = \sum_{l_1=1}^{n}\sum_{l_2=1}^{n} R_n(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}_i) \sum_{m_1=1}^{n}\sum_{m_2=1}^{n} s_m(\overline{\mathbf{x}}_{l_1,l_2}, \overline{\mathbf{y}}_{m_1,m_2}) \sum_{j=1}^{N} \sigma_{m,j} R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j) \tag{19}$$

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 m, we detail each operation.

5.3 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.

$$W^{m,B}_{m_1,m_2} = \sum_{\mathbf{y}_j \in B} \sigma_{m,j} R_n(\overline{\mathbf{y}}^B_{m_1,m_2}, \overline{\mathbf{y}}_j) \tag{20}$$

for $m_i = 1, \ldots, n$.

5.4 M2M

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

$$W^{m,A}_{m_1,m_2} = \sum_{i=1}^{4} \sum_{m'_1=1}^{n} \sum_{m'_2=1}^{n} W^{m,B_i}_{m'_1,m'_2} R_n(\overline{\mathbf{y}}^A_{m_1,m_2}, \overline{\mathbf{y}}^{B_i}_{m'_1,m'_2}) \tag{21}$$

for $m_i = 1, \ldots, n$.

5.5 M2L

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

$$g^{m,A}_{l_1,l_2} = \sum_{B_i} \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W^{m,B_i}_{m_1,m_2} s_m(\overline{\mathbf{x}}^A_{l_1,l_2}, \overline{\mathbf{y}}^{B_i}_{m_1,m_2}) \tag{22}$$

for $l_i = 1, \ldots, n$.

5.6 L2L

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

$$f^{m,A}_{l_1,l_2} = g^{m,A}_{l_1,l_2} + \sum_{l'_1=1}^{n} \sum_{l'_2=1}^{n} f^{m,B}_{l_1,l_2} R_n(\overline{\mathbf{x}}^A_{l_1,l_2}, \overline{\mathbf{x}}^B_{l'_1,l'_2}) \tag{23}$$

for $l_i = 1, \ldots, n$ where B is the parent box of box A. This f is analogous to the local expansion in the analysis-based FMM.

5.7 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.

$$f_m(\mathbf{x}_i) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} f_{l_1,l_2}^{m,B} R_n(\overline{\mathbf{x}}_{l_1,l_2}^{B}, \mathbf{x}_i) \tag{24}$$

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. This is a positive because the operations themselves will be less costly since we don't need to evaluate the kernel, and as discussed previously, they don't change based on which mode of the kernel we're using so these operations can be vectorized. The pre-computation itself will also be fast since all but the $M2L$ similarly do not depend on which mode of the kernel we're using, so we only need to compute these operators once instead of $2M + 1$ times in the KIFMM.

The $M2L$ operation deserves special attention because it does depend on the kernel s_m and is the most expensive pre-computation and operation. However, there are many techniques for not only accelerating the pre-computation of the operator as described by [2], but also its application as described in [1], [**?**], [2], and [**?**].

The computation of s_m, generates each of the $2M + 1$ Fourier modes all at once using the Fast Fourier Transform (FFT).

$$s_m = \int_0^{2\pi} \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|} e^{-im\theta'} d\theta' \tag{25}$$

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], [**?**], and [**?**]. 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.

6 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] and [5].

The following are descriptions of the computations done by each of our test programs.

- **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

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

- **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

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

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.

- **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

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

- **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

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

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

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

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

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

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

[5] 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.

[6] 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.

[7] 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.