The problem of constructing an FMM for axisymmetric density distributions

May 4, 2016

1 Introduction

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

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

$$r = \sqrt{x^2 + y^2} \tag{1}$$

$$\theta = \begin{cases} 0 & \text{if } x = 0 \text{ and } y = 0 \\ \arcsin(\frac{y}{r}) & \text{if } x \geq 0 \\ \arctan(\frac{y}{x}) & \text{if } x > 0 \\ -\arcsin(\frac{y}{r}) + \pi & \text{if } x < 0 \end{cases} \tag{2}$$

$$z = z \tag{3}$$

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

Density distributions on axisymmetric surfaces have many applications. For example, the problem of electromagnetic scattering by a surface of revolution has radar, geophysical exploration, and acoustics applications. This type of problem relies on creating a fast algorithm for a set of densities (charges, weights, etc.) distributed on an axisymmetric surface.

2 Statement of problem

Consider P source points $\mathbf{y}_j = (r', \theta', z')$ distributed on this surface Γ, with each assigned a charge σ_j of ± 1. We want to compute the potential at some target points $\mathbf{x}_i = (r, \theta, z)$ also on Γ (in practice sources and targets are the same points) due to the sources, which is represented by the sum

$$f(\mathbf{x}_i) = \sum_{j=1}^{P} K(\mathbf{x}_i, \mathbf{y}_j)\sigma_j \tag{4}$$

where $f(\mathbf{x}_i)$ is the potential and $K(\mathbf{x}, \mathbf{y})$ is a kernel, typically the Green's function of the partial differential equation that governs the relationship between sources and targets. In particular, we are interested in the free-space Green's function for Laplace's equation:

$$K(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|} \tag{5}$$

The goal is to construct a fast summation method for equation (4).

3 Modal Green's Function

One technique to construct a fast method for a density distribution on an axisymmetric surface is to reduce the problem in three dimensions to a series of problems in two dimensions using Fourier analysis. 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 can reduce the problem like this because the rotationally symmetric nature of the surface yields $K(\mathbf{x}, \mathbf{y})$ symmetric along the θ-axis such that the kernel is only a function of the difference $\theta - \theta'$:

$$K(\mathbf{x}, \mathbf{y}) = K(\theta - \theta', r, z, r', z') \tag{6}$$

We call such a kernel rotationally invariant.

Recall that if for two any points (r, θ, z) and (r', θ', z') in the computational domain a kernel can be written

$$K(\mathbf{x}, \mathbf{y}) = K(r, \theta, z, r', \theta', z') = K(\theta - \theta', r - r', z - z') \tag{7}$$

then it is called translation invariant. This is not the case for all kernels, however, and that fact is crucial to the difficulty of creating an optimal FMM in this scenario.

3.1 Fourier representation of 3D integral equations

Consider the Fredholm integral equation of the first kind defined on the axisymmetric surface Γ in three dimensions, which is a continuous version of equation (4):

$$f(\mathbf{x}) = \int_\Gamma K(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})d\mathbf{y} \qquad\qquad \mathbf{x},\mathbf{y} \in \Gamma \tag{8}$$

where K is a kernel function, σ are the charges, and f is the potential. Essentially equation (4) can just be viewed as a discretization of this integral equation.

We can restate this as a sequence of integral equations defined on the generating curve γ of Γ by performing a Fourier transform. If σ_n, f_n, and k_n are the Fourier modes of σ, f, and K, respectively, then we have

$$\sqrt{2\pi}\int_\gamma k_n(r,z,r',z')\sigma_n(r',z')r'dl(r',z') = f_n(r,z) \qquad\qquad (r',z') \in \gamma \tag{9}$$

for each $n \in \mathbb{Z}$, such that

$$f_n(r,z) = \int_{\mathbb{T}} \frac{e^{-in\theta}}{\sqrt{2\pi}} f(r,z,\theta)d\theta \qquad\qquad f(\mathbf{x}) = \sum_{n\in\mathbb{Z}} \frac{e^{in\theta}}{\sqrt{2\pi}} f_n(r,z) \tag{10}$$

$$\sigma_n(r,z) = \int_{\mathbb{T}} \frac{e^{-in\theta}}{\sqrt{2\pi}} \sigma(r,z,\theta)d\theta \qquad\qquad \sigma(\mathbf{x}) = \sum_{n\in\mathbb{Z}} \frac{e^{in\theta}}{\sqrt{2\pi}} \sigma_n(r,z) \tag{11}$$

$$k_n(r,z,r',z') = \int_{\mathbb{T}} \frac{e^{-in\theta}}{\sqrt{2\pi}} K(r,z,r',z',\theta)d\theta \qquad K(\mathbf{x},\mathbf{y}) = \sum_{n\in\mathbb{Z}} \frac{e^{in(\theta-\theta')}}{\sqrt{2\pi}} k_n(r,z,r',z') \tag{12}$$

As stated previously we are using the Green's function of a PDE as our kernel, so we call k_n the modal Green's function. In particular, we are interested in the Fourier expansion of the free-space Green's function for Laplace's equation

$$\frac{1}{4\pi|\mathbf{x}-\mathbf{y}|} = \frac{1}{4\pi\sqrt{r^2+r'^2-2rr'\cos(\theta-\theta')+(z-z')^2}} \tag{13}$$

$$= \sum_{n\in\mathbb{Z}} \frac{e^{in(\theta-\theta')}}{\sqrt{2\pi}} s_n(r,z,r',z') \tag{14}$$

where the modal Green's function is

$$s_n(r,z,r',z') = \frac{1}{\sqrt{8\pi^3 rr'}} \mathbf{Q}_{n-\frac{1}{2}}\left(\frac{r^2+(r')^2+(z-z')^2}{2rr'}\right) \tag{15}$$

with $Q_{n-\frac{1}{2}}$ the half-integer order Legendre function of the second kind. The efficient evaluation of modal Green's function for Laplace's equation, and the Q-function in particular,

has been explored by Abdelmageed (see equation 39), and others. These should be implemented to quickly evaluate the kernel at the different Chebyshev nodes. However, we are concerned not only with the efficient evaluation of this function but the fast evaluation of their sums.

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 $s_n(r, z, r', z')$. However, notice that s_n is translation variant. This has implications for the efficient pre-computation of translation operators which we discuss later.

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.

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

In the next sections, we look at how several well-known fast methods approach this problem.

4 Translation Operators

First, though, we need to talk about translation operators. Recall that in the analytic FMM, a translation operator translates the multipole or local expansion of one box to that of another in the computational domain. In kernel-independent methods, translation operators translate equivalent weights or densities in the same way. Translation operators are the backbone of any fast multipole method because these operations are how we reduce the number of direct computations required to compute the sum.

There are five translation operators. The source-to-multipole (S2M) operator creates a multipole expansion for a given box. The multipole-to-multipole (M2M) operator translates densities in a box to densities in its parent box. The multipole-to-local (M2L) operator translates densities in a box to another box on the same level in the hierarchical tree that divides the computational domain. The local-to-local (L2L) operator translates densities in a box to densities in a child box. Lastly, the local-to-target (L2T) operator computes the far field interaction for the target points using the local expansion.

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 relative difference between coordinate values, e.g. $(r - r', \theta - \theta', z - z')$. 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 are identical, and need not be computed repeatedly.

On the other hand, if a kernel is translation variant, then every translation operator for every box in the computational domain needs to be computed. None can be reused or recycled as above. This is the case for the modal Green's function, and we need to take this pre-computation into consideration when evaluating a fast method.

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.

5 Analytic FMM

The original FMM [4] first developed by Greengard and Rokhlin relies on analytic expansions for the potential at some target in the far field due to sources in a box (multipole) and the potential at a target in a box due to all sources in the far field (local). Originally this was done using Legendre polynomials and spherical harmonics to represent the kernel $1/r$.

Our full 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 free-space 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. In addition, this method will require the evaluation of special functions. Why is this bad/expensive?

In addition, a problem arises when we have many sources distributed in close proximity. (why is this the case again?)

6 Kernel-Independent Methods

Ruling out an analytic FMM to accelerate the sum, we're left with kernel-independent methods. 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. 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.

7 The kernel-independent FMM

Rather than using analytic multipole and local expansions like in the FMM, the KIFMM substitutes 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. The idea is that to a target box in the far field, the potential due to the source charges in the box 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. This allows us to construct an efficient FMM that only requires kernel evaluations. The KIFMM is also 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.

The original KIFMM paper explored three data sets for the 3D case: densities distributed on the unit sphere, densities distributed uniformly on the unit cube, and densities distributed at the eight corners of the unit cube. While the 3D KIFMM may be able to handle densities distributed on surfaces of revolution, [7] explains that the implementation is difficult because rather than using spherical equivalent surfaces to extrapolate from the circular surfaces used in the 2D case, we need to use cubes.

Due to this difficulty, the technique described earlier and covered at length in [8] 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.

Martinsson, Hao, and Young explored this application to modal green's functions in [8]. However, as aforementioned, the pre-computation is not optimal because of the translation operators for the translation variant kernel. In an attempt to reduce this pre-computation, we tried to find constant proportionality between the operators. For example, perhaps the $M2L$ from $[0,1] \times [0,1]$ to $[2,3] \times [0,1]$ is some explicit portion of the translation from $[0,1] \times [0,1]$ to $[3,4] \times [0,1]$. However, these attempts failed and indeed we wouldn't expect this relationship to exist because it doesn't exist in the kernel, and the translation operators in the KIFMM are simply kernel evaluations. So ever translation operator for every box on every level would need to be pre-computed. This is relatively expensive because every operator involves the evaluation of the kernel.

It's important to note that since the translation operators are 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. The full algorithm will require pre-computation $2N+1$ times the number of operators required for the computational domain, where N is the truncation parameter for the Fourier expansion.

8 The black-box FMM

The black-box FMM is a Chebyshev interpolation-based $\mathcal{O}(P)$ 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

$$f_n(\mathbf{x}_i) = \sum_{j=1}^{P} s_n(\mathbf{x}_i, \mathbf{y}_j)\sigma_{n,j} \tag{16}$$

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

The bbFMM is best for complicated analytic kernels that may not conform well to the FMM or KIFMM since it only requires a smooth kernel, and only uses the evaluation of the kernel 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 $\mathcal{O}(P)$. This is crucial for this problem since the pre-computation for the KIFMM was expensive because of the translation variant 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...

$$s_n(\mathbf{x}, \mathbf{y}) = \sum_i \alpha_i u_i(\mathbf{x}) v_i(\mathbf{y}) \tag{17}$$

and Chebyshev interpolation as in....

$$s_n(\mathbf{x}, \mathbf{y}) = \sum_i \sum_m s_n(\mathbf{x}_i, \mathbf{y}_m) w_i(\mathbf{x}) w_m(\mathbf{y}) \tag{18}$$

First we need to define the Chebyshev polynomial

$$T_k(x) = \cos(k \arccos(\frac{2}{b-a}(x - \frac{a+b}{2}))) \tag{19}$$

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

$$x = \frac{a+b}{2} + \frac{b-a}{2}\cos(\frac{2i-1}{2k}\pi) \text{ for } i = 1, \ldots, k. \tag{20}$$

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

From the Chebyshev polynomial Fong and Darve derive the interpolation functions

$$R_k(\mathbf{x}, \mathbf{y}) = (\frac{1}{k} + \frac{2}{k}\sum_{i=1}^{k-1} T_k(x_1)T_k(y_1))(\frac{1}{k} + \frac{2}{k}\sum_{i=1}^{k-1} T_k(x_2)T_k(y_2)) \tag{21}$$

for $\mathbf{x} = (x_1, x_2)$ and $\mathbf{y} = (y_1, y_2)$.

From there we use a low-rank approximation for the modal kernel given by

$$s_n(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{l}=1}^{k} \sum_{\mathbf{m}=1}^{k} s_n(\overline{\mathbf{x}}_\mathbf{l}, \overline{\mathbf{y}}_\mathbf{m}) R_k(\overline{\mathbf{x}}_\mathbf{l}, \mathbf{x}) R_k(\overline{\mathbf{y}}_\mathbf{m}, \mathbf{y}) \tag{22}$$

to approximate the sum

$$f_n(\mathbf{x}_i) = \sum_{\mathbf{l}=1}^{k} R_k(\overline{\mathbf{x}}_\mathbf{l}, \mathbf{x}_i) \sum_{\mathbf{m}=1}^{k} s_n(\overline{\mathbf{x}}_\mathbf{l}, \overline{\mathbf{y}}_\mathbf{m}) \sum_{j=1}^{P} \sigma_{n,j} R_k(\overline{\mathbf{y}}_\mathbf{m}, \mathbf{y}_j) \tag{23}$$

where P is the number of source points, k^2 is the number of Chebyshev nodes in each box, $\mathbf{y}_j$ are the source points, $\mathbf{x}_i$ are the target points, $\overline{\mathbf{x}}_\mathbf{l}$ are the Chebyshev nodes in the target box, and $\overline{\mathbf{y}}_\mathbf{m}$ are the Chebyshev nodes in the source box.

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

8.1 S2M

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

$$W^{n,B}_{m_1,m_2} = \sum_{\mathbf{y}_j \in B} \sigma_{n,j} R_k(\overline{\mathbf{y}}^B_{m_1,m_2}, \overline{\mathbf{y}}_j) \tag{24}$$

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

8.2 M2M

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

$$W^{n,A}_{m_1,m_2} = \sum_{i=1}^{4} \sum_{m'_1=1}^{k} \sum_{m'_2=1}^{k} W^{n,B_i}_{m'_1,m'_2} R_k(\overline{\mathbf{y}}^A_{m_1,m_2}, \overline{\mathbf{y}}^{B_i}_{m'_1,m'_2}) \tag{25}$$

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

8.3 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.

$$g^{n,A}_{l_1,l_2} = \sum_{B_i} \sum_{m_1=1}^{k} \sum_{m_2=1}^{k} W^{n,B_i}_{m_1,m_2} s_n(\overline{\mathbf{x}}^A_{l_1,l_2}, \overline{\mathbf{y}}^{B_i}_{m_1,m_2}) \tag{26}$$

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

8.4 L2L

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

$$f^{n,A}_{l_1,l_2} = g^{n,A}_{l_1,l_2} + \sum_{l_1'=1}^{k} \sum_{l_2'=1}^{k} f^{n,B}_{l_1,l_2} R_k(\overline{\mathbf{x}}^A_{l_1,l_2}, \overline{\mathbf{x}}^B_{l_1',l_2'}) \tag{27}$$

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

8.5 L2T

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

$$f_n(\mathbf{x}_i) = \sum_{l_1=1}^{k} \sum_{l_2=1}^{k} f^{n,B}_{l_1,l_2} R_k(\overline{\mathbf{x}}^B_{l_1,l_2}, \mathbf{x}_i) \tag{28}$$

Notice that the $S2M$, $M2M$, $L2L$, and $L2T$ translation operators in this scenario do not require kernel evaluations. Since the interpolation functions R_k 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 $2N + 1$ times in the KIFMM.

In addition, for each mode n, the weights $W^{n,B}_{m_1,m_2}$ can be represented by a matrix with k^2 rows and N, 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.

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

9 Numerical Result

I feel like there should be some numerical result, but perhaps it's too early for this and should just include appendix with an explanation of the test files. One thing the bbFMM authors did was the pre-computation time for all necessary operators over multiple computational domains with different numbers of levels. In particular, we fix the root level of the computational domain as the box $[0, 1] \times [0, 1]$ in two-dimensional (r, z)-space, and vary the number of levels in the computational domain, and the number of Chebyshev nodes in each box. Note that the number of Chebyshev nodes is the same for every box, regardless of size or level.

10 Future Work

Complete the bbFMM algorithm for modal Green's function.

Appendices

A Legend

n is the index for each Fourier mode of the Green's function

N is the truncation parameter for the Fourier expansion of the Green's function

P is the number of source points in the computational domain

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 FMM. However, 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. One key piece of this will be accelerating the $M2L$ operation using singular value decomposition (SVD) compression technique described in [1], [2], and [6].

Now we describe the computations done by each of our test programs.

2D − S2M.py

1. Put P 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

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

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

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

5. Compute the potential at the target by interpolation

$$f(\mathbf{x}_i) = \sum_{l_1} \sum l_2 f_{l_1,l_2} R_k(\overline{\mathbf{x}}_{l_1,l_2}, \mathbf{x}_i)$$

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

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

2D − M2M

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

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

$$W_{m_1,m_2} = \sum_{y_j \in A} \sigma_j R_k(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j)$$

3. Repeat step 2 for the four child boxes B_i of A.

4. Use the $M2M$ operation to compute

$$W^A_{m_1,m_2} = \sum_i \sum_{m_1'} \sum_{m_2'} R_k(\overline{\mathbf{x}}^A_{m_1,m_2}, \overline{\mathbf{x}}^{B_i}_{m_1',m_2'}) W^{B_i}_{m_1',m_2'}$$

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

2D − M2L + L2L.py

This file compares the actual local expansion of a box with our estimate of its local expansion. In this example we consider the computational domain to be only these boxes, with empty space (no sources) outside it.

The computation of the actual local expansion is straightforward. This is the potential at a target point inside the target box due to all sources in the far field $\mathbf{F}$. For our example, this means a target point in the box T due to sources in $P_1, \ldots, P_7$ and $C_1, \ldots, C_{27}$. For the target point $\mathbf{x_0}$ and source points $\mathbf{x_j}$ with weight σ_j, the calculation is:

$$f(\mathbf{x_0}) = \sum_{\mathbf{x_j} \in \mathbf{F}} K(\mathbf{x_0}, \mathbf{x_j})\sigma_j \tag{29}$$

Now we compute the estimated local expansion. In general in an FMM, the local expansion for a box is the sum of the multipole-to-local operations and the local-to-local operation for this box.

We start by placing sources points in each box $P_1, \ldots, P_7$ and $C_1, \ldots, C_{27}$, and assigning each source a charge of $+1$ or -1. The C boxes are in the interaction list of the target box. The P boxes are in the interaction list of the parent box of the target box. We need to develop an equivalent density for each box $P_1, \ldots, P_7$ and $C_1, \ldots, C_{27}$, which is a set of n^2 weights at the Chebyshev nodes in each box. For each box, take $P1$ for example, this equivalent density W at a node $\overline{y}_{m_1,m_2}$ is the sum over $\mathbf{y}_j$ in P_1 of the R function at each node and each source multiplied by the charge at the source:

$$W(m_1, m_2) = \sum_{y_j \in P_1} R_n(\overline{\mathbf{y}}_{m_1,m_2}, \mathbf{y}_j)\sigma_j \tag{30}$$

Once these weights have been computed for every box in the far field, we compute the multipole-to-local operations for each of the boxes in the interaction list of the parent box. The sum of these over each interaction list box will be the entire local expansion for the parent, since the contribution of the local-to-local operator for this parent is 0 as the grandparent in our case has no interaction list in the computational domain. The $M2L$ operation at each node in the parent box is such for parent box P and interaction boxes $P_1, \ldots, P_7$:

$$g^P_{l_1,l_2} = \sum_{i=1}^{7} \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W^{P_i}_{m_1,m_2} K(\overline{\mathbf{x}}^P_{l_1,l_2}, \overline{\mathbf{y}}^{P_i}_{m_1,m_2}) \tag{31}$$

Now we compute the same $M2L$ operations at each node in the target box T:

$$g^T_{l_1,l_2} = \sum_{i=1}^{27} \sum_{m_1=1}^{n} \sum_{m_2=1}^{n} W^{C_i}_{m_1,m_2} K(\overline{\mathbf{x}}^C_{l_1,l_2}, \overline{\mathbf{y}}^{C_i}_{m_1,m_2}) \tag{32}$$

Then we compute the estimated local expansion f at each node in the box T:

$$f^T_{l_1,l_2} = g^T_{l_1,l_2} + \sum_{l'_1=1}^{n} \sum_{l'_2=1}^{n} g^P_{l'_1,l'_2} R_n(\overline{\mathbf{x}}^T_{l_1,l_2}, \overline{\mathbf{x}}^P_{l'_1,l'_2}) \tag{33}$$

Finally we compute the estimated potential at the target point due to sources in the far field $\mathbf{F}$:

$$f(\mathbf{x}_0) = \sum_{l_1=1}^{n} \sum_{l_2=1}^{n} f^T_{l_1,l_2} R_n(\overline{\mathbf{x}}^T_{l_1,l_2}, \mathbf{x}_0) \tag{34}$$

Compare this with the naive computation of the potential.

References

[1] 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.

[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] 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.

[6] 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.

[7] 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.

[8] 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.