Accelerating the Multipole to Local Field Translations

Srinath Kailasa *
University College London

June 12, 2023

Contents

1 Accelerating the M2L with the SVD

This is the method first presented in [1]. We've also done some improvements to this based on the suggestions in [2], however I haven't added them to this discussion yet, hence the empty section.

*srinath.kailasa.18@ucl.ac.uk

Consider the application of the M2L operator K to a multipole expansion w to get the check potential g.

$$g = Kw \tag{1}$$

This can be approximated with a rank k SVD,

$$\tilde{g} = U_k \Sigma_k V_k^T \tag{2}$$

Stacking the M2L operators for all the source nodes in a given target node's interaction list can be done in two ways, column wise,

$$\begin{aligned} K_{\text{fat}} &= \left[K^1, ..., K^3 16\right] && (3)\\ &= U\Sigma \left[V^{(1)T}, ..., V^{(316)T}\right] && (4) \end{aligned}$$

where we use the fact that there are at most 316 unique orientations for the M2L operator in 3D. Similarly they can be stacked row wise,

$$\begin{aligned} K_{\text{thin}} &= \left[K^1; ...; K^{316}\right] && (5)\\ &= \left[R^{(1)T}; ...; R^{(316)T}\right] \Lambda S^T && (6) \end{aligned}$$

we note that

$$K_{\text{thin}} = K_{\text{fat}}^T \tag{7}$$

for symmetric kernels.

We can do some algebra to reduce the application cost of K when we've done these two SVDs. Consider the application of a single M2L operator corresponding to a single source box in a target box's interaction list,

$$K^{(i)} w = R^{(i)} \Lambda S^T w \tag{8}$$

Using the fact that S is unitary, $S^T S = I$, we can insert into the above equation,

$$K^{(i)}w = R^{(i)}\Lambda SS^TS^Tw \tag{9}$$
$$= K^{(i)}SS^Tw \tag{10}$$
$$= U\Sigma V^{(i)T}SS^Tw \tag{11}$$
$$\tag{12}$$

Now using the fact that U is also unitary, such that $U^TU = I$, we find

$$K^{(i)}w = UU^TU\Sigma V^{(i)T}SS^Tw \tag{13}$$
$$= U[U^TU\Sigma V^{(i)T}S]S^Tw \tag{14}$$
$$= U[U^TK^{(i)}S]S^Tw \tag{15}$$

The term in the brackets can be calculated using the low rank (k-rank) terms from the SVD,

$$[U^TK^{(i)}S] = \Sigma V^{(i)T}S \tag{16}$$
$$= U^TR^{(i)}\Lambda \tag{17}$$

We call this previous equation the compressed M2L operator,

$$C^{i,k} = U^TK^{(i)}S \tag{18}$$

This object can be pre-computed for each unique interaction. The M2L operation can be then broken down into 4 steps

1. Find the 'compressed multipole expansion'

$$w_c = S^Tw \tag{19}$$

2. Compute the convolution to find the compressed check potential

$$g_c = \sum_{i\in I} C^{i,k}w_c \tag{20}$$

where the sum is over the interaction list I.

3. A post processing step to recover the check potential

$$g = U g_c \tag{21}$$

4. The calculatation of the local expansion, as usual, in the KIFMM.

Doing this the convolution step is reduced to matrix vector products involving the compressed M2L matrix, which is only of size $k \times k$, rather than $6(p-1)^2 + 2$ where p is the expansion order.

1.1 Taking Advantage of Modern Compute Architectures

For scale invariant kernels (e.g. Laplace, Helmholtz etc) many M2L translations can be seen to be rotations/scalings of each other. The authors of [2] take advantage of this to batch together the matvecs that correspond to M2L interactions into cache-efficient matrix-matrix products that take advantage of highly-efficient BLAS L3 operations. We describe our adaption of this approach here, as well as its limitations ...

2 Accelerating M2L with FFT

The M2L operation can also be accelerated with a fast fourier transform (FFT). The M2L accelerated this way is quite natural, as it's simply a convolution operation, however computing it in practice can be be tricky. Here I document how I've managed to compute it, as well as a summary of the relevant FFT theory as a background.

2.1 Fourier Transform Theoretical Background

A lot of the theoretical background I want to keep at hand is taken from the excellent course notes [3]. I summarise the key aspects here as related to the FFT, especially when discussing padding/indexing, as these issues come up most pertinently in real implementations.

2.1.1 Going from Fourier Series to Fourier Transforms

Starting off with Fourier Series (FS), i.e. representing periodic functions using a periodic (trig) basis, and generalising to non-periodic (i.e. ∞ period) functions takes us to Fourier Transforms (FT).

Q: Is the sum of two periodic functions also periodic?

A: No if you're a mathematician, e.g. $cos(t)$ and $cos(\sqrt{2}t)$ are each periodic with periods 2π and $2\pi/\sqrt{2}$ resp. But the sum is not periodic. ie. no common divisors in the periods.

When considering a sum of sinusoids, as Fourier pitched,

$$\sum_{n=1}^{N} A_n \sin(n\theta + \phi_n) \tag{22}$$

The sum is also periodic as the frequencies are multiples of the fundamental frequency $1/2\pi$.

It's more common to write a general trig sum as,

$$\frac{a_0}{2} + \sum_{n=1}^{N} (a_n \cos(2\pi nt) + b_n \sin(2\pi nt)) \tag{23}$$

Where the zeroth component is often referred to as a DC component (from electrical engineering contexts). The half is a simplifying factor that comes up. Expressing this instead using complex exponentials, the sum can be written as,

$$\sum_{n=-N}^{N} c_n e^{2\pi i n t} \tag{24}$$

One can refer to RHB to see how the coefficients are related between forms. In particular we find $c_0 = a_0/2$. The complex conjugate property of the coefficients,

$$c{-}n = \bar{c_n} \tag{25}$$

is important, it allows us to group terms such that

$$\sum_{n=-N}^{N} c_n e^{2\pi i n t} = 2\mathrm{Re}\left\{\sum_{n=0}^{N} c_n e^{2\pi i n t}\right\} \tag{26}$$

Our goal is to express a general periodic function $f(t)$ as an FS.

$$f(t) = \sum_{-N}^{N} c_n e^{2\pi i n t} \tag{27}$$

Take a given coefficient, can we solve for it ?

$$f(t) = \sum_{-N}^{N} c_n e^{2\pi i n t} \tag{28}$$

$$e^{-2\pi i k t} f(t) = e^{-2\pi i k t} \sum_{-N}^{N} c_n e^{2\pi i n t} \tag{29}$$

Therefore,

$$c_k = e^{-2\pi i k t} f(t) - \sum_{n=-N, n\neq k}^{N} c_n e^{2\pi i (n-k) t} \tag{30}$$

We've pulled the coefficient out, but the expression involves all the other coefficients! Instead, we can try and integrate both sides over 0 to 1 (any function can be made to have this period if it's periodic). The integrals in the sum all cancel out,

$$\int_0^1 e^{2\pi (n-k) t} dt = \frac{1}{2\pi i (n-k)} e^{2\pi i (n-k) t} \Big|_{t=0}^{t=1} = 0 \tag{31}$$

With this trick, the expression for the coefficient reduces to,

$$c_k = \int_0^1 e^{-2\pi i k t} f(t) dt \tag{32}$$

We haven't stated whether any periodic function *can* be expressed in such a way that we can apply this analysis, but if we can express it in the periodic form we started off with, we have a way of evaluating the coefficients.

Note in particular that the zeroth coefficients corresponds to an average value of the function over its period.

$$\hat{f}(0) = \int_0^1 f(t)dt \tag{33}$$

The case when all the coefficients are real is when the signal is real and even. For then,

$$\bar{\hat{f}}(n) = \hat{f}(-n) = \int_0^1 e^{-2\pi i(-n)t} f(t)dt = \int_0^1 e^{2\pi int} f(t)dt \tag{34}$$

$$= -\int_0^{-1} e^{-2\pi ins} f(-s)ds, \text{ subs t = -s, changing lims} \tag{35}$$

$$= \int -1^0 e^{-2\pi ins} f(-s)ds, \text{ even f(s)} \tag{36}$$

$$= \hat{f}(n) \tag{37}$$

So the coefficients are real. The evenness of f seems to pass over into its fourier coefficients too.

We haven't yet answered when a periodic function can be approximated by a fourier series . . . We're basically allowed to if $f(t) \in L^2([0,1])$ as then the integral defining its Fourier coefficients exists. The fourier approximation is the best approximation in $L^2([0,1])$ by a trigonemtric polynomial of degree N. The complex exponentials form a basis for this space, and the partial sums converge to $f(t)$ in its norm,

$$\lim_{N\to\infty} \left\| \sum_{n=-N}^{N} \hat{f}(n) e^{-2\pi int} - f(t) \right\| = 0 \tag{38}$$

For Fourier Transforms, lets start off by considering a box function.

$$\Pi(t) = \begin{cases} 1 & \text{if } |t| < 1/2, \\ 0 & \text{if } |t| \geq 1/2. \end{cases} \tag{39}$$

This isn't periodic, and doesn't have an FS. However, if we make it repeat with intervals T, we can find a representation with coefficients given by,

$$c_n = \frac{1}{T}\int_0^T e^{-2\pi int/T} f(t)dt = \frac{1}{T}\int_{-T/2}^{T/2} e^{-2\pi int/T} f(t)dt = \frac{1}{\pi n}\sin(\frac{\pi n}{T}) \quad (40)$$

The coefficients tend to 0 for large T as $1/T$, to compensate for this we can scale by T. Using a change of variables $s = n/T$ we can write,
$\Pi(s) = \frac{\sin(\pi s)}{\pi s}$
We can now take a limit as $T \to \infty$,

$$\hat{\Pi}(s) = \int_{-\infty}^{\infty} e^{-2\pi ist}\Pi(t)dt = \int_{-1/2}^{1/2} e^{-2\pi ist}\cdot 1 dt = \frac{\sin(\pi s)}{\pi s} \quad (41)$$

We are lead to the same idea - scale the Fourier coefficients by T - if we had started off periodising any function that is zero outside of some interval and letting the period tend to infinity. This gives us the following definition for Fourier Transforms,

$$\hat{f}(s) = \int_{-\infty}^{\infty} e^{-2\pi ist} f(t)dt \quad (42)$$

where the coefficients are in general complex. FTs produce continuous spectra, in contrast to a discrete set of (potentially infinitely many) frequencies as in FS.

We can push this to get a definition for the dual, the inverse transform. Again supposing that we have a non-periodic function that we can say is zero outside of an interval, we find an expression for its FS, and fourier coefficients

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi int/T} \quad (43)$$

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} e^{-2\pi int/T} f(t)dt = \frac{1}{T}\int_{-\infty}^{\infty} e^{-2\pi int/T} f(t)dt \quad (44)$$

$$\text{extension to infty ok as zero outside interval} \quad (45)$$

$$= \frac{1}{T}\hat{f}(\frac{n}{T}) = \frac{1}{T}\hat{f}(s) \quad (46)$$

Plugging back in, and thinking of Riemann sum to approximate an integral,

$$f(t) = \sum_{-\infty}^{\infty} \frac{1}{T} \hat{f}(s_n) e^{2\pi i s_n t} = \sum_{-\infty}^{\infty} \hat{f}(s_n) e^{2\pi i s_n t} \Delta s \approx \int_{-\infty}^{\infty} \hat{f}(s_n) e^{2\pi i s_n t} ds \quad (47)$$

2.1.2 The Convolution

In general we want to modify signals by each other. Is there a combination of signals $f(t)$ and $g(t)$ such that in the frequency domain the FT is:

$$\mathcal{F}g(s)\mathcal{F}f(s)$$

i.e. is there a combination of the signals such that frequency components are scaled by each other?

Very roughly, we find,

$$\mathcal{F}g(s)\mathcal{F}f(s) = \int_{-\infty}^{\infty} e^{-2\pi i s t} g(t) ds \int_{-\infty}^{\infty} e^{-2\pi i s x} f(x) dx \quad (48)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-2\pi i s (t+x)} g(t) f(x) dt dx \quad (49)$$

using the change of variable $u = t + x$ for the inner integral,

$$\int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} e^{-2\pi i s u} g(u - x) du \right) f(x) dx \quad (50)$$

switching the order of integration,

$$\int_{-\infty}^{\infty} e^{-2\pi i s u} \left(\int_{-\infty}^{\infty} g(u - x) f(x) dx \right) du \quad (51)$$

The inner integral can be seen to be a function of u, we can write it as $h(u)$, the outer integral reduces to:

$$\int_{-\infty}^{\infty} e^{-2\pi i s u} h(u) du = \mathcal{F}h(s) \quad (52)$$

This defines our convolution,

$$(g * f)(t) = h(t) = \int_{-\infty}^{\infty} g(t-x)f(x)dx \tag{53}$$

And the following theorem,

$$\mathcal{F}(g * f)(s) = \mathcal{F}g(s)\mathcal{F}f(s) \tag{54}$$

Most significantly for us, convolving in the time domain reduces to a multiliplication in the frequency domain.

The convolution is defined by flipping the kernel, and dragging it over the signal.

2.1.3 Discrete Fourier Transforms, and the Fast Fourier Transform

We want to find a discrete analogue to the FT for real signals which are sampled at a certain rate.

Let's suppose that $f(t)$ is zero outside of an interval $0 \leq t \leq L$, similarly the FT $\mathcal{F}f(s)$ is assumed zero outside of $0 \leq s \leq 2B$ (indexing is easier if we ignore negative frequencies), L and B are both integers.

According to Shannon, we can reconstruct $f(t)$ perfectly if we sample at a rate of $2B$ per second. So in total we want,

$$N = \frac{L}{1/2B} = 2BL$$

evenly spaced samples, notice that this is even. Sampled at points,

$$t_0 = 0, t_1 = \frac{1}{2B}, ..., t_{N-1} = \frac{N-1}{2B}$$

$$f_{discrete}(t) = \sum_{n=0}^{N-1} \delta(t - t_n)f(t_n) \tag{55}$$

and therefore,

$$\mathcal{F}f_{discrete}(t) = \sum_{n=0}^{N-1} f(t_n)\mathcal{F}\delta(t - t_n) = \sum_{n=0}^{N-1} f(t_n)e^{-2\pi i s t_n} \tag{56}$$

which is almost what we need, it's the continuous FT of the sampled form of $f(t)$.

Shifting to the frequency domain, we find the number of sample points to be,

$$N = \frac{2B}{1/L} = 2BL$$

the same as in the time domain. We base the discrete version of the FT using the discrete version of the signal,

$$F(s_0) = \sum_{n=0}^{N-1} f(t_n)e^{-2\pi i s_0 t_n} \tag{57}$$

etc. We now have a way of converting from the discrete signal to the discrete FT,

$$F(s_m) = \sum_{n=0}^{N-1} f(t_n)e^{-2\pi i s_m t_n} \tag{58}$$

It's possible to link this to the continuous case by discretising the integral defining a continuous FT, we see that this sum (up to a scaling) comes out.

using,

$$t_n = \frac{n}{2B}, \;\; s_m = \frac{m}{L} \tag{59}$$

we can write in terms of indices,

$$F(s_m) = \sum_{n=0}^{N-1} f(t_n)e^{-2\pi i n m/2BL} = \sum_{n=0}^{N-1} f(t_n)e^{-2\pi i n m/N} \tag{60}$$

Thinking about io as sequences of numbers, we can write in 'array' form, where the transform is just defined on a sequence.

$$\mathbf{F}[m] = \sum_{n=0}^{N-1} \mathbf{f}[n] e^{-2\pi i m n/N}, \;\; m = 0, 1, ..., N-1 \tag{61}$$

The input sequence can be complex, it's not less valid, but the output sequence is always complex.

A common notation is to write the complex exponentials as,

$$\omega = e^{2\pi i/N} = \omega_N$$

s.t.

$$\omega_N^N = 1$$

for any integer n and k,

$$\omega_N^{Nn} = 1$$

$$\omega_N^{Nn+k} = \omega_N^k$$

and,

$$\omega_N^{N/2} = -1$$

so,

$$\omega_N^{kN/2} = (-1)^k$$

We write a vector of the Nth roots of unity as,

$$\omega = (1, \omega, \omega^2, ..., \omega^{N-1})$$

the components,

$$\omega^k[m] = \omega^{km}$$

The DFT can be thought of as a linear transfrom between $\mathbb{C}^N$ to $\mathbb{C}^N$. This linear transform can be explicitly written out as a matrix, which I won't bother with here, look at 257 in [3].

This is a dense $N \times N$ matrix! The FT is in general hard to compute, hence the revolution of the FFT which can do it in log-linear time.

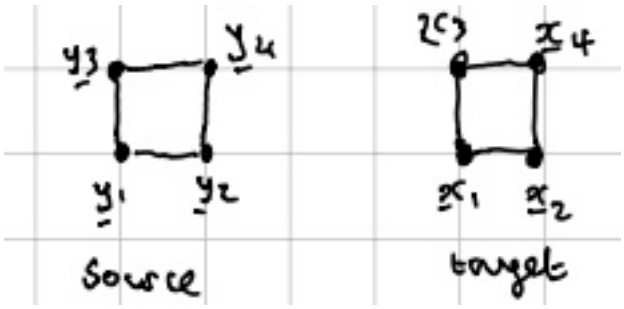
2.2 The M2L Translation as a Fourier Convolution

For the M2L operation we're computing the following convolution,

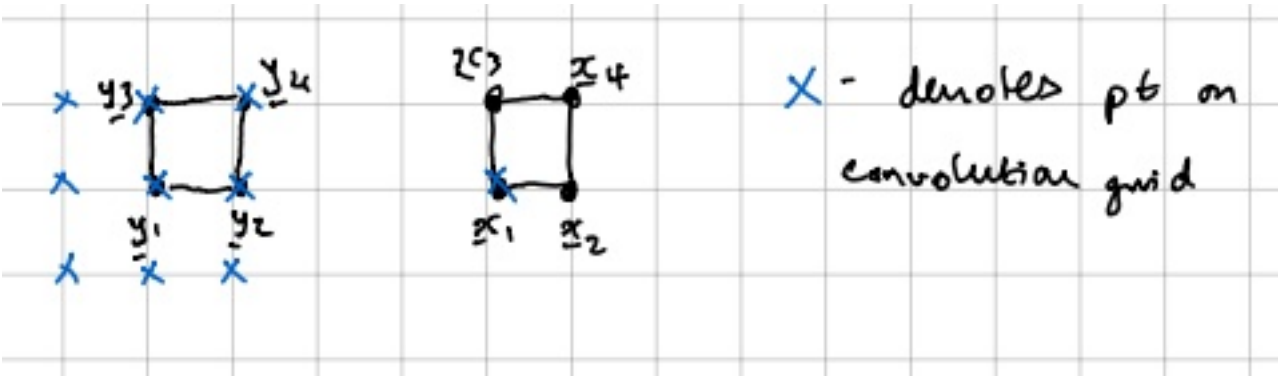
$$\phi(x) = \int G(x-y)q(y)dy \tag{62}$$

where we're attempting to compute the far-field potential as a convolution of the Green's function with a charge distribution (multipole expansion) at some local box. This is definitely somewhere we can apply the FT/FFT. How is this actually done in practice though, we're only concerned about the BBFMM [1] case where the charge distributions/multipole expansions are placed at regular intervals on the surface of a box enclosing a node in the octree. In the literature there is a significant gap in describing how to actually setup the convolution operation such that we can apply the FFT to accelerate it, I illustrate it pictorially below.

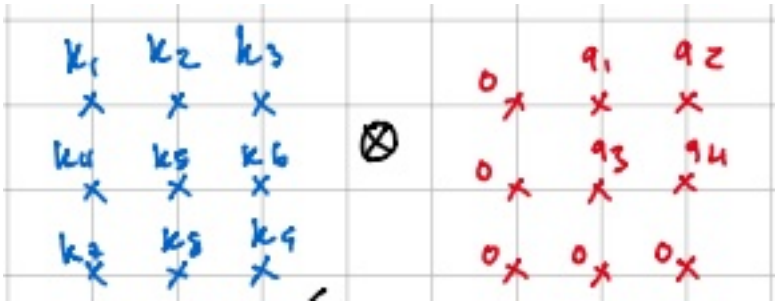
Consider two boxes (source and target) in 2D for simplicity, the expansion order is set to $p = 2$



The boxes are referred to as lying in a 'surface grid', with the equivalent charges, $q_1, ..., q_4$, placed at $y_1, ...y_4$. We embed the unique kernel interactions between these two boxes on a so called 'convolution grid', we define them wrt to a fixed point - we can take this to be just x_1. These can be pre-computed and stored.



The unique interactions define the convolution grid points. We label these $K_1, ..., K_9$. This is how the convolution is defined practically.



When we go ahead and compute this, taking care to 'flip' the kernel values, we find the potentials we're looking for embedded in at the following corresponding points on the convolution grid (flipped wrt to the positions of the equivalent densities). Only the four positions that correspond to the positions of the original equivalent densities are significant, the remainder can be ignored.

We can accelerate this convolution computation using the FFT as normal. Mathematically, this can be expressed as follows ...

2.3 Taking Advantage of Modern CPU Architectures

We want to batch together computations sibling interactions and take advantage of SIMD to maximally take advantage of the cache hierarchies in modern CPUs. One strategy is as follows, we conclude by contrasting it with the BLAS3/SVD approach, and include some numerical benchmarks to contrast the two ...

References

[1] William Fong and Eric Darve. "The black-box fast multipole method". In: *Journal of Computational Physics* 228.23 (2009), pp. 8712–8725. ISSN: 10902716. DOI: 10.1016/j.jcp.2009.08.031.

[2] Matthias Messner et al. "Optimized M2L Kernels for the Chebyshev Interpolation based Fast Multipole Method". In: (2012), pp. 1–23. arXiv: 1210.7292. URL: http://arxiv.org/abs/1210.7292.

[3] B Osgood. "EE261 - Fourier Transform and its applications". In: *Lecture Notes for EE 261 - The Fourier Transform and its Applications* (2014), pp. 1–498.