Implementing Fast Multipole Methods with High-Level Interpreted Languages

Srinath Kailasa

A thesis submitted in partial fulfillment of the requirements
for the degree Master of Science

Department of Physics
University College London
August 18, 2020

Declaration

I, Srinath Kailasa, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

The Fast Multipole Method (FMM) is a numerical method to accelerate the solution of the N-Body problem, which appears in numerous contexts in science and engineering, for example in solving for gravitational or electrostatic potentials in multi-particle systems. It does so by approximating the Green's function of the system with analytic infinite series expansions (the origin of the 'multipole' in its name), and coalescing the effect of distinct distant sources together so as to greatly reduce the number of computations. The analytic expansions of the original Fast Multipole Method depend on the Green's function of the system in question (Helmholtz, Laplace etc.), and in practice a new implementation must be written for a given system.

The Kernel-Independent Fast Multipole Method (KIFMM), first presented by Ying et al. [6], is a similar approach that replaces the analytic series expansions with a continuous distribution of so called 'equivalent density' supported at discrete points on a box enclosing a set of particles. These equivalent densities are found by matching the potential they generate to those generated by the original sources at another surface in the far field. Usefully, this approach doesn't require multipole expansions of the Green's functions of a system, and therefore can be programmed in an agnostic way, hence the origin of it's name.

This thesis presents a well tested and extensible Python implementation of the KIFMM, with investigations made into both mathematical and computational techniques for the acceleration of the software, using the N-Body electrostatic problem as model on which to test the implementation. Python is chosen as it has emerged as a standard for scientific and data intensive computing in recent years, with a huge increase in adoption, and a well supported ecosystem of libraries and tools available for accelerating numerical codes. The wider context of this thesis is an ongoing collaboration with the ExaFMM Project [7] to produce a Python implementation of the KIFMM that sacrifices as little performance as possible. This thesis introduces the relevant theoretical background, before proceeding to discuss the strategies used in the practical implementation of the software. Key bottlenecks in the implementation are examined and addressed, with the speed and accuracy of the software benchmarked with the Laplace kernel for electrostatic problems. Finally, future avenues of investigation and software development are discussed, in the context of the wider literature.

Contents

Glossary

far field An set of particles in the far field are considered to be far away enough from the particle of interest that they are suitably described by a convergent multipole expansion. . 5

FMM The Fast Multipole Method. . 1, 4, 5

KIFMM The 'Kernel Independent' Fast Multipole Method. . 1

near field An set of particles in the near-field are considered to violate the the criteria for describing them with a convergence multipole expansion. . 8

near neighbours Two boxes in computational tree are near neighbours if they are at the same level of refinement, and share a boundary point. . 16

well separated Two boxes in computational tree are near neighbours if they are at the same level of refinement, and are not near neighbours. . 16

Introduction

1.1 Overview of the Analytic FMM

1.1.1 Motivation

The paradigmatic problem of Fast Multipole Methods (**FMM**) [1] is the so called N-Body problem. This classic problem refers to the calculation of the pairwise interactions between N particles over a potentially long-range, for example in gravitational or electrostatic systems. The straightforward calculation can be written in the form of the following sum,

$$\Phi(x_j) = \sum_{i=1}^{N} w_i K(x_i, x_j) \tag{1.1}$$

Where $i, j \in [1, N]$ and $K(x, y)$ is called the Green's function, or equivalently a 'kernel function', where one is generally concerned with coordinates of particles in an $n = 2$ or 3 dimensional Hilbert space taking $x_i \in \mathbb{R}^n$. Additionally, each summand is weighted by w_i. For solving for electrostatic potential in three dimensions, which is used as the model problem throughout this thesis, this goes to,

$$\Phi(x_j) = \sum_{i=1}^{N} q_i K(x_i, x_j) \tag{1.2}$$

where q_i refers to a charge density with the kernel function,

$$K(x, y) = \frac{1}{4\pi\epsilon_0} \frac{1}{|x - y|} \tag{1.3}$$

the constant ϵ_0 is the permittivity of free space. It's easy to see how a naive direct application of this equation over N particles results in an algorithm of $O(N^2)$ complexity, therefore it's only practicable for systems of moderate size, whereas realistic systems, one may be interested in interactions involving 10^6 to 10^8 particles.

This chapter introduces the analytic, FMM, the kernel-independent version, which is the main focus of this thesis, is presented later. Though substantially different in implementation, the analytic FMM will provide the opportunity to exposit many of the key ideas behind all FMM-based algorithms, and provides a good starting point for understanding and developing upon these algorithms. First presented by Greengard [3], the analytic FMM represented a sea change for N-Body simulation. By trading off computations for error, it manages to achieve an asymptotic complexity of just $O(N)$. Additionally, it comes equipped with rigorous error

[1] The first usage of a technical term or abbreviation listed in the glossary is highlighted throughout the text for ease of reference.

bounds, making fast and accurate massive N-Body simulations feasible on available computing hardware. It's success has been such that it is regarded as one of the key developments in numerical algorithms in the twentieth century [2].

The original analytic FMM solves the electrostatic problem in two and three dimensions, this is equivalently known as the Poisson problem, represented by the differential equation,

$$\nabla^2 \phi = f \tag{1.4}$$

Where ϕ is some scalar potential to be determined, and f is a scalar source term which is usually known. For electrostatics the corresponding formulation can be derived from Gauss' law as [4],

$$\nabla^2 \phi = -\frac{q}{\epsilon_0} \tag{1.5}$$

where ϕ is the electrostatic potential, q is the charge density and ϵ_0 is the permittivity of free space. It can therefore be seen that the FMM is actually solving the Poisson problem by reformulating it as an integral equation. The ubiquity of problems of the form (1.1) in computational science has lead to diverse application of the FMM. For example, in the modeling the electrostatic interactions of charged particles in complex biological molecules at biologically relevant length scales [1]. The extension of FMMs to Helmholtz equations [8], has lead to even more applications, such as in seismic and acoustic scattering [5]. Though as the focus of this thesis is on solving the Laplace model problem for electrostatics, this is mentioned only for completeness.

The key insight that leads to the FMM's asymptotic complexity is the idea that if the field created by a distribution of charge (or mass) density is approximated to be relatively smooth in the **far field**, then it should be possible to apply some form of compression for the evaluation of contribution to local potentials due to particles in the far field. The FMM performs this compression by encoding the field contributions of particles in the far field using a multipole expansion.

For simple kernel functions and charge distributions, such as the model problem of this thesis, one can easily derive the expression for this multipole expansion by finding an series expansion of the system's Green's function. In order to generalise the discussion, an arbitrary continuous distribution of charge is considered as shown in figure 1.1, for which the potential is evaluated at some other evaluation point outside of the distribution. This can be written as follows,

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int \frac{1}{d} \rho(\mathbf{r}') d\tau' \tag{1.6}$$

where $\rho(\mathbf{r}')$ is a charge density, and the other symbols take their meanings from figure (1.1).

From law of the cosines,

$$d^2 = r^2 + (r')^2 - 2rr' \cos\alpha = r^2 \left[1 + \left(\frac{r'}{r} \right)^2 - 2 \left(\frac{r'}{r} \right) \cos\alpha \right] \tag{1.7}$$

$$d = r\sqrt{1+\epsilon} \tag{1.8}$$

Where,

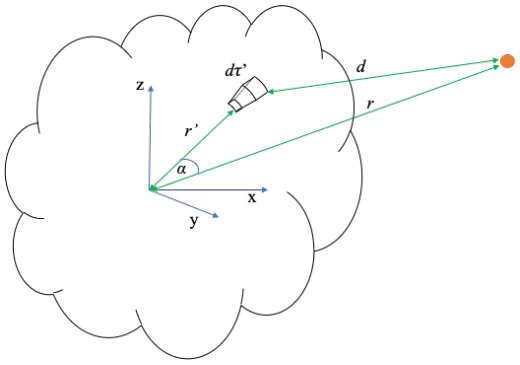

Figure 1.1: An arbitrary charge distribution, with an orange point to mark a point where the potential is being evaluated. Here, $\mathbf{r}$ is the the vector between the centre of the multipole expansion and the evaluation point, $\mathbf{r}'$ is the vector between the centre of expansion and a given volume element $d\tau'$, and d is a vector between the volume element $d\tau'$ and the evaluation point.

$$\epsilon \equiv \left(\frac{r'}{r}\right)\left(\frac{r'}{r} - 2\cos\alpha\right) \tag{1.9}$$

As ϵ is small far away from charge distribution one can expand $1/d$ binomially,

$$\frac{1}{d} = \frac{1}{r}(1+\epsilon)^{-1/2} = \frac{1}{r}\left(1 - \frac{1}{2}\epsilon + \frac{3}{8}\epsilon^2 - ...\right) \tag{1.10}$$

$$\frac{1}{d} = \frac{1}{r}\sum_{n=0}^{\infty}\left(\frac{r'}{r}\right)^n P_n(\cos\alpha) \tag{1.11}$$

Using this, the exact multipole expansion for this charge distribution is,

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_{n=0}^{\infty}\frac{1}{r^{n+1}}\int (r')^n P_n(\cos\alpha)\rho(\mathbf{r}')d\tau' \tag{1.12}$$

If instead we consider composed of N charges q_i this goes to,

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N}\sum_{n=0}^{\infty}\frac{(r')^n q_i}{r^{n+1}}P_n(\cos\alpha) \tag{1.13}$$

Using the addition theorem for Legendre polynomials [3],

$$P_n(\cos\gamma) = \sum_{m=-n}^{n} Y_n^{-m}(\alpha,\beta)Y_n^m(\theta,\phi) \tag{1.14}$$

Where we have written the Legendre polynomial in terms of spherical harmonics, where (r,θ,ϕ) and (ρ,α,β) define two spherical coordinates, and γ is the angle subtended between them. Therefore, the multipole expansion goes to,

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N}\sum_{n=0}^{\infty}\frac{(r')^n q_i}{r^{n+1}}P_n(\cos\alpha) \tag{1.15}$$

$$= \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}\frac{(r')^n q_i Y_n^{-m}(\alpha_i,\beta_i)}{r^{n+1}}Y_n^m(\theta,\phi) \tag{1.16}$$

$$= \sum_{i=1}^{N}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}\frac{M_n^m}{r^{n+1}}\cdot Y_n^m(\theta,\phi) \tag{1.17}$$

This is an exact expansion, and it converges for $\frac{r'}{r} < 1$. This convergence condition means estimating of the potential at a given evaluation point calculated using the multipole expansion is only possible in the far-field, the boundary of which is often tuned empirically for different systems as it's user defined. If instead the expansion is taken with centre at the evaluation point, one can rewrite as the multipole expansion as a 'local' expansion,

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N}\sum_{n=0}^{\infty}\sum_{m=-n}^{n}\frac{(r)^n q_i Y_n^{-m}(\alpha_i,\beta_i)}{r'^{n+1}}Y_n^m(\theta,\phi) \tag{1.18}$$

$$= \frac{1}{4\pi\epsilon_0}\sum_{i=1}^{N}\sum_{n=0}^{\infty}\sum_{m=-j}^{n}L_m^n\cdot Y_m^n(\theta,\phi)\cdot r^j \tag{1.19}$$

which converges when $\frac{r}{r'} < 1$. The region of convergence for both types of expansions are shown in figure (1.2).

The key point to note is that the multipole and local expansions are exact, and can be truncated as required to ensure that the asymptotic complexity of evaluating a multipole or local expansion at an evaluation point is bounded by $O(N)$.

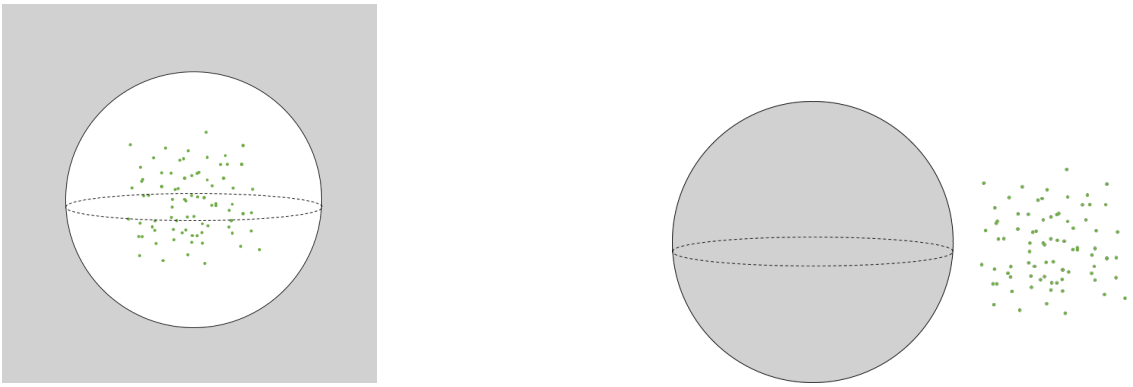

Figure 1.2: (A) A multipole expansion centered on charge distribution. (B) A local expansion, centered around a point of evaluation. Regions in which the expansions converge are shaded in grey. For the multipole expansion this is the entire domain outside of the region for $r > r'$, and for the local expansion this is the region for which $r < r'$

1.1.2 Algorithm structure and analysis

The convergence condition of the multipole expansion prohibits the compression of charges from particles in the **near field**. Therefore the FMM makes use of a tree structure in a recursive algorithm, known as an Octree in three dimensions and a Quadtree in two dimensions. This structure hierarchically partitions space such that each level, l, of the tree is equally partioned into $(2^n)^l$ boxes over the domain of the tree, where n is the dimension, i.e. $n = 3$ in three dimensions. If one were to simply traverse the tree from the coarsest level to the finest, or 'leaf', level and find the multipole expansion of source particles in each box of each level, one could then evaluate these multipole expansions at each particle to solve the N-Body problem. As there are $O(\log(N))$ boxes in the tree, and N particles this results in a $O(N \log(N))$ asymptotic complexity.

However the FMM reduces computational complexity further by making use of local expansions. There are analytic expressions to shift the multipole expansion coefficients M_n^m to local expansion coefficients L_n^m [2]

Notes from [6]

FMM makes use of these representations in a recursive algorithm. computational domain is a box containing all particles, sources and targets. Hierarchically partitioned into a tree structure, called Octree in 3D. With each level l of the tree partitioned into 8^l geometric boxes. For each box, the potential induced by it's source densities is represented by a multipole expansion centered around the box, while the potential induced by the sources from non-adjacent boxes is encoded in a local expansion.

Number of expansion terms p is chosen for a prescribed relative error ϵ, using $p = \log_c \epsilon$ where $c = \frac{4-\sqrt{3}}{\sqrt{3}}$ in 3D (numerically optimal?). The trucation error has rigorous bounds.

The availability of analytic translations enable the $\mathcal{O}(n)$ algorithm. In particular the following translations; M2M, L2L, M2L.

Two basic steps of FMM using tree structure.

1. *Upward Pass* The tree is traversed, post-order. S2M at leaves - compute multipole expansion of leaf sources at leaf node. Shift multipole expansion to parent node, and sum together.

2. *Downward Pass* The tree is traversed pre-order. The local expansion for each box is the sum of two parts: (1) the L2L transformation collects the local expansion of B's parent (compressing the information for boxes non-adjacent to B's parent) (2) the M2l translation, for each multipole expansion of boxes which are the children of the neighbors of B's parent but not adjacent to B itself.At the end of the Downward Pass, the 'far' interaction which is evaluated using the local expansion at this box (L2T operation) at each target particle. Combined with the 'near' interaction by direct computation of potential over all source points in near field (within the box itself, and it's direct neighbors).

1.1.3 Analysis

Begin with $O(N \log(N))$ algorithm variant; here post order traversal (from coarsest to finest level), compute multipole expansion for each box at each level, this has the

[2] see appendix A.2 for expressions for these shift operators in three dimensions.

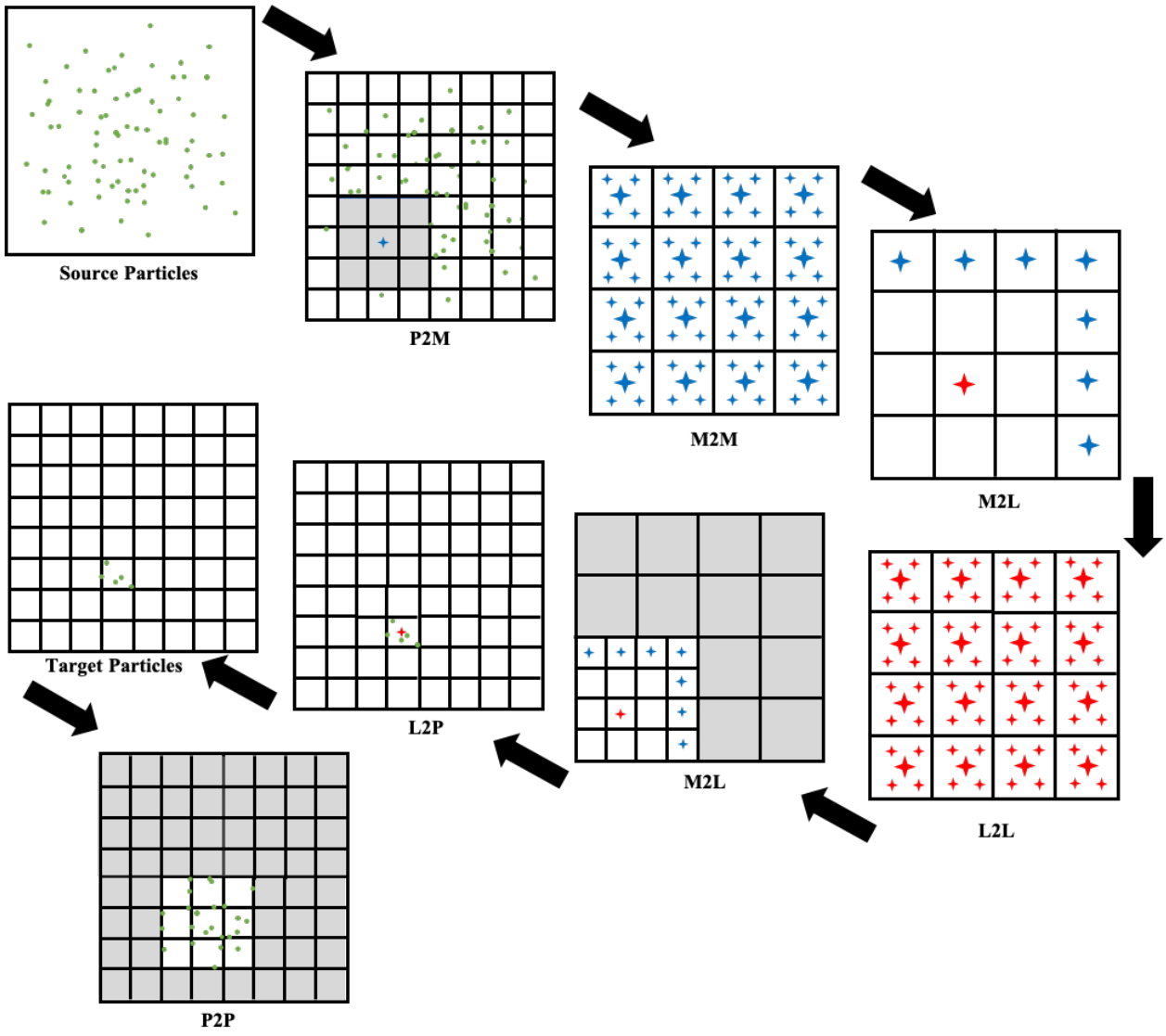

Figure 1.3: foo bar

Np^2 bound, at finest level assume $O(1)$ number of particles, so direct computation with nearest neighbour particles leads to $O(N)$ bound.

Full analysis defer to [3]. Enough to understand that the translation operators are what lead to the $\mathcal{O}(n)$ complexity. Beginning with upward pass, at leaf level each particle contributes to one expansion so S2M of the order Np^2 - p^2 is the order of operations for the multipole expansion, can see this from the equation. M2M/L2L/M2L shifts require p^4 operations, so all are bounded by $O(Np^4)$ (consider last level), precise bound dictated by number of boxes in interaction list. Evaluating the p^th degree local expansions at each target particle (L2T) bounded by $O(Np^2)$, small constant κ particles enforced at leaf level leads to $O(\kappa N)$ complexity for direct calculations at leaf level. We see that the whole algorithm is bounded by $O(N)$.

1.2 Overview of the Kernel-Independent FMM

1.2.1 Motivation

Analytic FMM implementation is Kernel specific, as we need to compute the multipole expansion coefficients. It's instead based solely on kernel evaluations, therefore can be programmed in an agnostic way.

1.2.2 Algorithm Structure

1.2.3 Analysis

Strategy for Practical Implementation

2.1 Bottleneck Analysis

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

2.2 Space-Filling Curves

2.3 Operator Caching

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

2.4 SVD Compression

2.5 Software Design

Experiments & Results

3.1 Section 1

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam

vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. å

Conclusion

4.1 Section 1

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Appendix

A.1 FMM Algorithm Specification

textbfNotes from [3]

Initialisation: Choose refinement of $n \approx \log_8 N$ and precision ϵ, set $p = \lceil -\log_2(\epsilon) \rceil$.

Upward Pass

Step 1

From multipole expansions of potential field due to particles in each box at leaf level.

do $ibox = 1, .., 8^n$
Form p^{th} degree multipole expansion
end

Step 2

Translate Multipole expansion to coarser levels from the bottom up.

do $l = n - 1, .., 0$
do $ibox = 1, .., 8^l$
M2M shifting
end
end

Downward Pass

Computations at the coarsest possible level. For a given box, done by including interactions with those boxes which are well separated, and whose interactions have not been accounted for at the parent level.

Step 3

Form local expansion about center of each box at each level $l \leq n-1$, describes field due to all particles that are not contained in the current box, it's near neighbours or it's secondary near neighbors.
do $l = 1, ..., n - 1$
do $ibox = 1, ..., 8^l$
Do M2L OPERATIONS
end

do $1, ..., 8^l$
Do L2L OPERATIONS
end
end

Step 4

After this step, leaf local expansions are available, can use this to evaluate potential at leaves from all particles in the far field.

do $ibox = 1, ..., 8^n$
Find local expansion at leaf level, by doing M2L from interaction list.
end

Step 5

Evaluate local expansions at particle positions in all leaves

do $ibox = 1, ..., 8^n$
For every particle in *ibox*'th box, evaluate local expansion.
end

Step 6

Compute nearest neighbors directly,
do $ibox = 1, ..., 8^n$
For every particle in *ibox*'th box, compute potential directly with nearest neighbors.
end

Step 7

do $ibox = 1, ..., 8^n$
Add direct and far field terms together for every particle in *ibox*
end

A.2 Analytic FMM Operators for 3D Laplace Kernel

Expressions derived in [3]

For l charges of strengths $q_1, ..., q_l$ located inside sphere D of radius a center at $Q = (\rho, \alpha, \beta)$, and that for points $P = (r, \theta, \phi)$ outside D potential given by multipole expansion
Multipole Expansion,

$$\Phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{O_n^m}{r'^{n+1}} \cdot Y_n^m(\theta', \phi')) \tag{A.1}$$

$P - Q = (r', \theta', \phi')$. Then for any point P outside sphere D_1 of radius $a + \rho$, M2M shifted

$$\Phi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} \frac{M_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi) \tag{A.2}$$

Defined,

$$M_j^k = \sum_{n=0}^{j} \sum_{m=-n}^{n} \frac{O_{j-n}^{k-m} \cdot J_m^{k-m} \cdot A_n^m \cdot A_{j-n}^{k-m} \cdot \rho^n \cdot Y_n^{-m}(\alpha, \beta)}{A_j^k} \tag{A.3}$$

and,

$$J_m^{m'} = \begin{cases} (-1)^{\min(|m'|,|m|)} & \text{if } m \cdot m' < 0 \\ 1 & \text{otherwise} \end{cases} \tag{A.4}$$

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)! \cdot (n+1)!}} \tag{A.5}$$

M2L shifted

For l charges of strengths $q_1, ..., q_l$ located inside sphere D_Q of radius a center at $Q = (\rho, \alpha, \beta)$, and that $\rho > (c + 1)a$ with $c > 1$. The corresponding multipole expansion, converges inside sphere D_0 of radius a centered at origin. Inside D_0 the potential due to charges has the local expansion,

$$\Phi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j \tag{A.6}$$

where,

$$L_j^k = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \frac{O_n^m \cdot J_k^m \cdot A_n^m \cdot A_j^k \cdot Y_{j+n}^{m-k}(\alpha, \beta)}{A_{j+n}^{m-k} \cdot \rho^{j+n+1}} \tag{A.7}$$

where A_n^m same as above,
but,

$$J_m^{m'} = \begin{cases} (-1)^{n'}(-1)^{\min(|m'|,|m|)} & \text{if } m \cdot m' < 0 \\ (-1)^{n'} & \text{otherwise} \end{cases} \tag{A.8}$$

n' refers to j, m' refers to k
and that for points $P = (r, \theta, \phi)$ outside D potential given by multipole expansion

L2L shifted

Let $Q = (\rho, \alpha, \beta)$ be the origin of a local expansion,

$$\Phi(P) = \sum_{n=0}^{p} \sum_{m=-n}^{n} O_n^m \cdot Y_n^m(\theta', \phi') \cdot r'^n \tag{A.9}$$

Where $P = (r, \theta, \phi)$ and $P - Q = (r', \theta', \phi')$.

$$\Phi(P) = \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j \tag{A.10}$$

where,

$$L_j^k = \sum_{n=j}^{p} \sum_{m=-n}^{n} \frac{O_n^m \cdot J_{n-j,m-k}^m \cdot A_{n-j}^{m-k} \cdot A_j^k \cdot Y_{n-j}^{m-k}(\alpha, \beta) \cdot \rho^{n-j}}{A_j^k} \tag{A.11}$$

where A_n^m same as above,

$$J_{n,m}^{m'} = \begin{cases} (-1)^n(-1)^m & \text{if } m \cdot m' < 0 \\ (-1)^n(-1)^{m'-m} & \text{if } m \cdot m' > 0 \text{ and } |m'| < |m| \\ (-1)^n & \text{otherwise} \end{cases} \tag{A.12}$$

Bibliography

[1] John A. Board et al. "Accelerated molecular dynamics simulation with the parallel fast multipole algorithm". In: *Chemical Physics Letters* 198.1 (1992), pp. 89 –94. ISSN: 0009-2614. DOI: https://doi.org/10.1016/0009-2614(92)90053-P. URL: http://www.sciencedirect.com/science/article/pii/000926149290053P.

[2] Barry A. Cipra. "The Best of the 20th Century: Editors Name Top 10 Algorithms". In: *SIAM News* 33.4 (2000).

[3] Leslie Greengard. "The Rapid Evaluation of Potential Fields in Particle Systems". PhD thesis. Yale University, 1987.

[4] David J. Griffiths. *Introduction to Electrodynamics*. 4th ed. Cambridge University Press, 2017. DOI: 10.1017/9781108333511.

[5] Wen-Mei W. Hwu. *GPU Computing Gems Emerald Edition*. 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN: 0123849888.

[6] Denis Zorin Lexing Ying George Biros. "A kernel-independent adaptive fast multipole algorithm in two and three dimensions". In: *Journal of Computational Physics* 196.2 (2004), pp. 591–626. DOI: http://dx.doi.org/10.1016/j.jcp.2003.11.021.

[7] Lorena A. Barba Rio Yokota. *ExaFMM User's Manual*. 2011. URL: http://www.bu.edu/exafmm/files/2011/06/ExaFMM-UserManual1.pdf.

[8] Vladimir Rokhlin. "Rapid Solution of Integral Equations of Theory in Two Dimensions". In: *Journal of Computational Physics* 86.2 (Feb. 1990), pp. 414–439. DOI: 10.1016/0021-9991(90)90107-C.