Approximating the Gravitational Potential Using Multipole Expansions

PHYS 590 Thesis, Queen's University

Name: Scott Thomas
Supervisor: Professor Widrow

March 20, 2019

Abstract

N-body simulations require the force in order to calculate the dynamics of the system. The force can be calculated from direct summation of the Newtonian gravitational force equation, requiring N^2 operations per time step. The force can also be calculated from a tree algorithm which group the force from far bodies to reduce the amount of steps, requiring $\sim N\log(N)$ calculations per time step. Both these methods are computationally expensive for large N. In order to ease computational burden one can use a method of Multipole Expansion (MEX) to find the gravitational potential, and then subsequently the force. MEX is based on series expansion of Poisson's equation.

A code was developed using Python to calculate the MEX approximation and was found to be in good agreement with the forces produced by a uniform density analytic toy model and that of a full N-body code Gadget. Future possibilities include the introduction of a softening parameter in MEX to stop the blow up of forces in high density regions, as well as creating forces as functions of time and beginning to evolve N-body systems using the MEX approximation.

Introduction

A galaxy is a system of stars, gas, dust and dark matter held together by gravitational attraction, governed by Newton's laws (Binney & Tremaine 1987). One of the main techniques used to study gravitational systems, in particular galaxies, is by direct N-body simulation. N-body simulations require the gravitational force in order to calculate the dynamics of the system. The gravitational force can be calculated by direct summation of the Newtonian gravitational force equation, requiring N^2 operations per time step, which is consequently expensive for large N. Another method to calculate gravitational force is to implement a tree algorithm, which makes use of neglecting individual bodies' that are far away from the body of interest, then calculating the net force from the collection of bodies centre of mass. A tree algorithm requires $\sim N\log(N)$ operations per time step. More information on tree algorithms can be found in Bode et al. 2000, and its references. To contrast with direct calculation, the gravitational potential, which can be used to find forces, can be approximated by a Self-Consistent Field (SCF) method or Multipole Expansion (MEX). SCF involves an expansion in the angular and radial parts while MEX is just an expansion in the angular part (Meiron et al. 2014). Details about SCF can be found in Hernquist & Ostriker 1992, while details about MEX can be found in Meiron et al 2014 and Binney & Tremaine 1987. MEX has been advocated for as it is able to easily capture radial structure as there is only an expansion in the angular part (Meiron et al. 2014). It also has the benefit of being able to capture the potentials of arbitrary distributions of matter (Binney & Tremaine 1987), meaning it is very useful in many cases, such as the interactions stellar clusters have on the structure of a galactic disk.

MEX seeks the solutions of Poisson's equation, $\nabla^2\Phi = 4\pi G\rho$, for a thin spherical shell of variable surface density. If the shell is sufficiently thin, the problem reduces to solving Laplace's equation, $\nabla^2\Phi = 0$, on both sides of the shell, with suitable boundary conditions at the origin, on the shell, and at infinity. Laplace's equation can be solved by separation of variables. The potential for the system is then found by the contribution of all the spherical shells. More information on MEX, and for a thorough derivation of the potential, see Binney & Tremaine 1987. The specific potential (gravitational potential per unit mass) is described by equation (2-122) of Binney & Tremaine 1987,

$$\Phi(r,\theta,\phi) = -4\pi G \times \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} \frac{Y_l^m(\theta,\phi)}{2l+1}\left[\frac{1}{r^{(l+1)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da + r^l\int_r^\infty \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da\right] \tag{1}$$

where ρ_{lm} are radial basis functions. l_{max} can be seen as distribution arbitrariness, the more arbitrary the distrubtion the higher l_{max} needs to be to accurately capture the potential. To get the true potential, regardless of distribution, l_{max} needs to go to infinity.

Equation (1) can be written in a more suggestive form (the motivation for rewriting equation (1) will become clear later in the report):

$$\Phi(r,\theta,\phi) = -4\pi G \times \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} \frac{Y_l^m(\theta,\phi)}{2l+1}\left[\mathrm{Int}_{lm}(r)\right] \tag{2}$$

where,

$$\mathrm{Int}_{lm}(r) = \frac{1}{r^{(l+1)}} \int_0^r \rho_{lm}(a) a^{(l+2)}\, da + r^l \int_r^\infty \rho_{lm}(a) \frac{1}{a^{(l-1)}}\, da \tag{3}$$

Using this model, the potential of arbitrary N-body distributions can be found in a very algorithmic way.

Since the potential using MEX can be found in a very algorithmic way, by taking the negative gradient of (1) the forces per unit mass can be found, that is $\boldsymbol{f} = -\boldsymbol{\nabla}\Phi$ (Thornton S., Marion J., 2003). The forces per unit mass are then given by,

$$f_r = 4\pi G \times \sum_{l,m} \frac{Y_l^m}{2l+1} \left[\frac{-(l+1)}{r^{(l+2)}} \int_0^r \rho_{lm}(a) a^{(l+2)}\, da + l r^{(l-1)} \int_r^\infty \rho_{lm}(a) \frac{1}{a^{(l-1)}}\, da \right] \tag{4}$$

$$f_\phi = 4\pi G \times \sum_{l,m} im \frac{Y_l^m}{2l+1} \frac{1}{r\sin(\theta)} \left[\frac{1}{r^{(l+1)}} \int_0^r \rho_{lm}(a) a^{(l+2)}\, da - r^l \int_\infty^r \rho_{lm}(a) \frac{1}{a^{(l-1)}}\, da \right] \tag{5}$$

$$f_\theta = 4\pi G \times \sum_{l,m} \frac{\frac{\partial Y_l^m}{\partial \theta}}{2l+1} \frac{1}{r} \left[\frac{1}{r^{(l+1)}} \int_0^r \rho_{lm}(a) a^{(l+2)}\, da - r^l \int_\infty^r \rho_{lm}(a) \frac{1}{a^{(l-1)}}\, da \right] \tag{6}$$

Where $\frac{\partial Y_l^m}{\partial \theta}$ is defined to be[1]

$$\frac{\partial Y_l^m}{\partial \theta} = m\ \cot(\theta)\, Y_l^m + \sqrt{(l-m)(l+m+1)}\, e^{-i\phi}\, Y_l^{m+1} \tag{7}$$

and by applying an appropriate coordinate transformation $\boldsymbol{f}$ can be found in cartesian coordinates

$$f_x = f_r \sin(\theta)\cos(\phi) + f_\theta \cos(\theta)\cos(\phi) - f_\phi \sin(\phi) \tag{8}$$

$$f_y = f_r \sin(\theta)\sin(\phi) + f_\theta \cos(\theta)\sin(\phi) + f_\phi \cos(\phi) \tag{9}$$

$$f_z = f_r \cos(\theta) - f_\theta \sin(\theta) \tag{10}$$

The derivation of the forces from the potential can be found in the appendix.

As equations (4), (5) and (6) can be written in the same form as equation (1), they too can be found in a very algorithmic way.

As MEX allows the potential and forces to be found in a very algorithmic way, this thesis aims to develop a code successful in calculating the gravitational potential and successfully calculating the forces per unit mass using MEX.

[1] Link to definition of spherical harmonic derivatives: http://functions.wolfram.com/Polynomials/SphericalHarmonicY/20/01/01/0001/

Methodology

The code that was developed to compute the MEX potential was written in Python. The code was made using two pre existing Python packages; NumPy[2] and SciPy[3]. NumPy was used for it's N-dimensional array and it's math functions, such as sine and cosine, while SciPy was used for numerical integration and interpolation.

From examining equation (1), it is clear that the approximation should be calculated in separate steps. The steps in which the MEX approximation is calculated is as follows:

Step 1: Radially bin the particles

Step 2: Calculate the radial basis functions, ρ_{lm}.

Step 3: Calculate the integral over each combination of lm, $\text{Int}_{lm}(r)$.

Step 4: Compute the sum over each particle position for each combination of lm.

In order to calculate the potential, the particles of the system need to be binned radially. Choosing to use bins spaced linearly allows sub structure and sharp edges in halos to be captured.

After the particles are binned radially, the radial basis functions, ρ_{lm}, need to be calculated:

$$\rho_{lm}(r) = \sum_i \frac{m_i Y_l^{m^*}(\theta_i, \phi_i)}{r_i^2 \Delta r} \tag{11}$$

Where r_i, m_i, θ_i, ϕ_i is the position and mass of the i^{th} particle in the logarithmic bin described by bin-centre r and bin width Δr. ρ_{lm} is a function over the centres of the radial bins. The derivation of the ρ_{lm} basis functions can be seen in the appendix.

Once the ρ_{lm}'s have been found for each combination of l and m, then by integrating over the radial bin centres of equation (3), the Int_{lm}'s can be found.

After finding the Int_{lm}'s, the potential at the particle points can be found. By specifying the position of the particle in spherical coordinates, (r, θ, ϕ), the potential is then returned at that position. The sum to find the potential at any given specified point is mass independent and does not need to be an actual particle position. This means that the potentials of multi mass systems can be found simply by using this scheme with high resolution.

Similarly, to find the forces, the same steps are taken with a few exceptions in steps 3 and 4. By examining equations (4), (5) and (6) we see that there are two unique $\text{Int}_{lm}(r)$ that needs to be calculated, one for f_r and one for f_θ and f_ϕ. Once the two $\text{Int}_{lm}(r)$'s are found, step 4 needs to modified to account for the fact that the sum will need to be computed for each component of the force separately and then transformed into cartesian coordinates.

The algorithm complexity for each step of the potential and force calculation is the same. The algorithm complexity for each step is as follows:

Step 1: $\sim m_r \, N$

Step 2: $\sim m_r \, N \, l_{max}^2$

Step 3: $\sim m_r \, l_{max}^2$

[2] Link to NumPy documentation: `https://docs.scipy.org/doc/numpy/`

[3] Link to SciPy documentation: `https://docs.scipy.org/doc/scipy/reference/`

Step 4: $\sim\ n\, l_{max}^2$

Where N is the number of particles, m_r is the number of radial bins, n is the number of points to find the force at and l_{max} is the highest order expansion term.

Results

As the forces are directly related to the potential, a toy model will be used to compare the forces that are calculated by MEX. The toy model is a uniform density halo. The expected force for the toy model can be found by solving Poisson's equation inside the halo and Laplace's equation outside the halo for the potential and then applying the negative gradient to find the force.

Starting with inside the halo,

$$\nabla^2\Phi = 4\pi G\rho \tag{12}$$

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\Phi}{\partial r}\right) = 4\pi G\rho \tag{13}$$

$$\frac{\partial}{\partial r}\left(r^2\frac{\partial\Phi}{\partial r}\right) = 4\pi G\rho r^2 \tag{14}$$

$$r^2\frac{\partial^2\Phi}{\partial r^2} + 2r\frac{\partial\Phi}{\partial r} = 4\pi G\rho r^2 \tag{15}$$

$$\frac{\partial^2\Phi}{\partial r^2} + \frac{2}{r}\frac{\partial\Phi}{\partial r} = 4\pi G\rho \tag{16}$$

$$\Phi(r) = \frac{4\pi G\rho r^2}{6} + \frac{C_1}{r} + C_2 \tag{17}$$

Now as the potential must be finite at the origin, so $C_1 = 0$, now by taking the negative gradient of (17) the force can be found.

$$\boldsymbol{f} = -\boldsymbol{\nabla}\Phi(r) \tag{18}$$

As $\Phi(r)$ is only a function of r, $f_\theta = f_\phi = 0$.

$$f_r = -\frac{\partial}{\partial r}\left(\frac{4\pi G\rho r^2}{6} + C_2\right) \tag{19}$$

$$\boxed{f_{r,\text{ inside}} = -\frac{8\pi G\rho r}{6}} \tag{20}$$

Now outside of the halo $\rho = 0$,

$$\nabla^2\Phi = 0 \tag{21}$$

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\Phi}{\partial r}\right) = 0 \tag{22}$$

$$\frac{\partial}{\partial r}\left(r^2\frac{\partial\Phi}{\partial r}\right) = 0 \tag{23}$$

$$r^2\frac{\partial^2\Phi}{\partial r^2} + 2r\frac{\partial\Phi}{\partial r} = 0 \tag{24}$$

$$\frac{\partial^2\Phi}{\partial r^2} + \frac{2}{r}\frac{\partial\Phi}{\partial r} = 0 \tag{25}$$

$$\Phi(r) = \frac{C_3}{r} + C_4 \tag{26}$$

Again, $f_\theta = f_\phi = 0$,

$$f_r = -\frac{\partial}{\partial r}\left(\frac{C_3}{r} + C_4\right) \tag{27}$$

$$\boxed{f_{r,\text{ outside}} = \frac{C_3}{r^2}} \tag{28}$$

C_3 can be found by equating the radial forces at the halo edge, r_{halo}

$$\frac{C_3}{r_{halo}^2} = -\frac{8\pi G\rho r_{halo}}{6} \tag{29}$$

$$\boxed{C_3 = -\frac{8\pi G\rho r_{halo}^3}{6}} \tag{30}$$

A uniform density halo was made consisting of 85302 particles each having a mass of $8.16 \cdot 10^5$ M$_\odot$ and an $r_{halo} = 100$kpc, with an overall density of $1.66 \cdot 10^4$ M$_\odot$/kpc. The forces produced by MEX will be compared to that of the analytic model along the positive z axis for the halo centred on the origin.

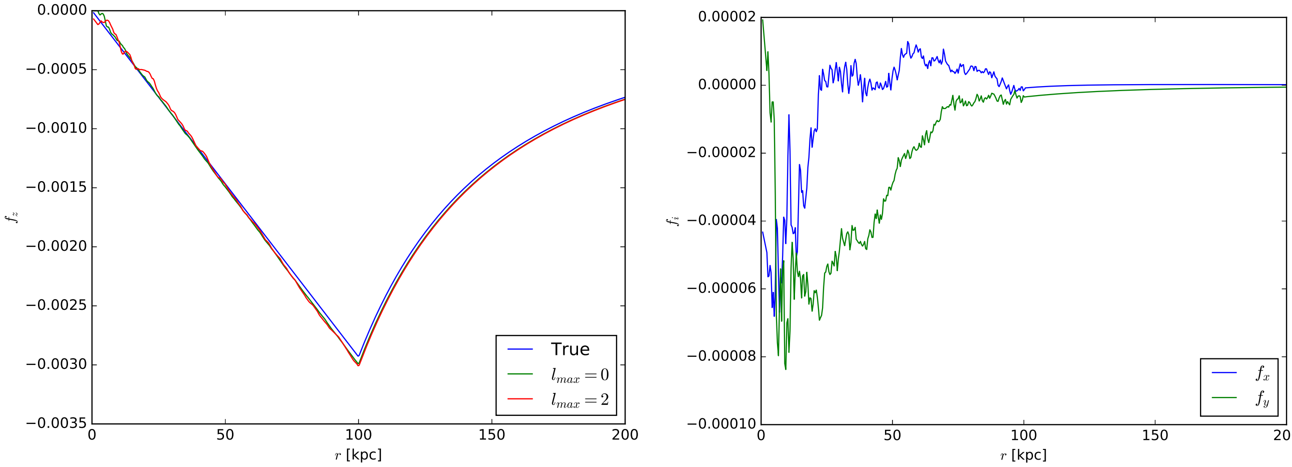


Figure 1: Force per unit mass of MEX compared to that of the toy model centred on the origin. The plot on the left is f_z for varying l_{max} with the analytic forces, the plot on the right is of f_x and f_y for $l_{max} = 2$. 500 radial bins are being used.

Looking at Fig. 1, both l_{max} of 0 and 2 capture the force well with minimal differences between them as the distribution is spherically symmetric. Looking at the plot on the right, both f_x and f_y are noisy for $r < r_{halo}$ and converge to zero for $r > r_{halo}$. This is expected behaviour as inside it is not perfectly uniform as the particles are randomly distributed.

The halo will now be shifted along the positive z axis so the centre of the halo will have a position of $z = 400$kpc. By doing this the distribution will no longer be spherically symmetric and there should now be noticeable differences by varying l_{max}.

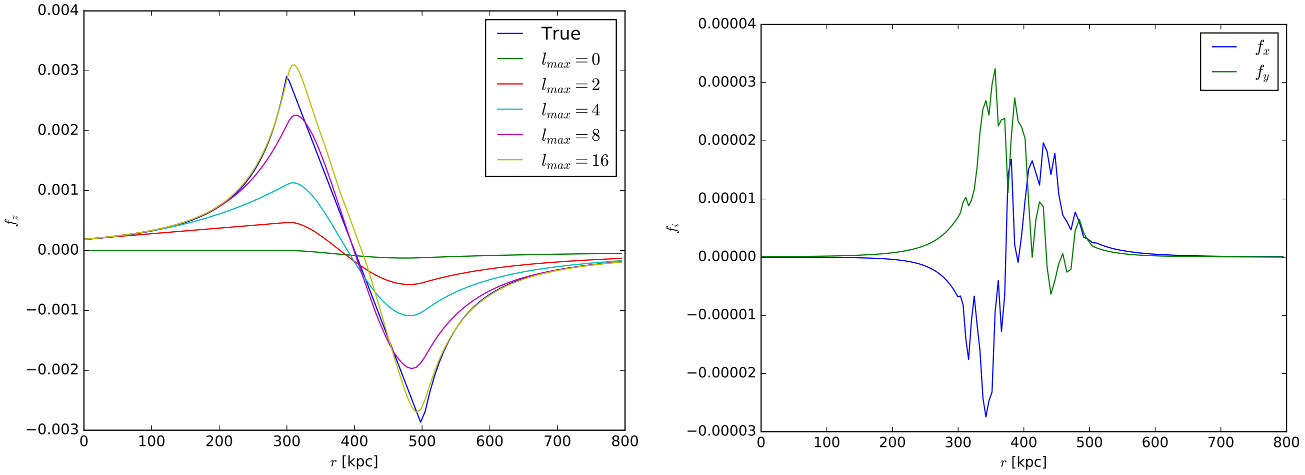


Figure 2: Force per unit mass of MEX compared to that of the toy model centred at $z = 400$kpc. The plot on the left is f_z for varying l_{max} with the analytic forces, the plot on the right is of f_x and f_y for $l_{max} = 16$. 500 radial bins are being used.

Looking at Fig. 2, it can be seen that as l_{max} increases the force approaches the analytic model. Looking at the plot on the right, both f_x and f_y are noisy for $300 < r < 500$ and converge to zero for $r < 300$ and $r > 500$. This is again because the halo is not perfectly uniform as the particles are randomly distributed inside.

In order to better determine how varying l_{max} and the number of radial bins impacts the accuracy of the approximation the halo will now be centred on the position $(x, y, z) = (150\text{kpc}, 150\text{kpc}, 150\text{kpc})$. The forces will then be calculated at 5000 random particle positions uniformly distributed in a sphere of radius 500kpc while varying l_{max} and the number of radial bins used. In order to qualitatively judge how well a certain pairing of l_{max} and bins captures the forces, the root mean square (RMS) values will be calculated between the analytic model and MEX for each component of force, that is,

$$\text{RMS}_i = \frac{1}{N} \sum_{j}^{N} \sqrt{(f_{i,j\ \text{MEX}} - f_{i,j\ \text{analytic}})^2} \tag{31}$$

The RMS values were calculated for each combination of $l_{max} = 0, 2, 4, 6, 8, 10$ and number of radial bins of 200,250,300,350,400. By plotting the RMS values against l_{max} and the number of radial bins it will be clear how changing these parameters impacts the accuracy of the MEX approximation. The RMS values can be seen in Table 1 in the appendix.

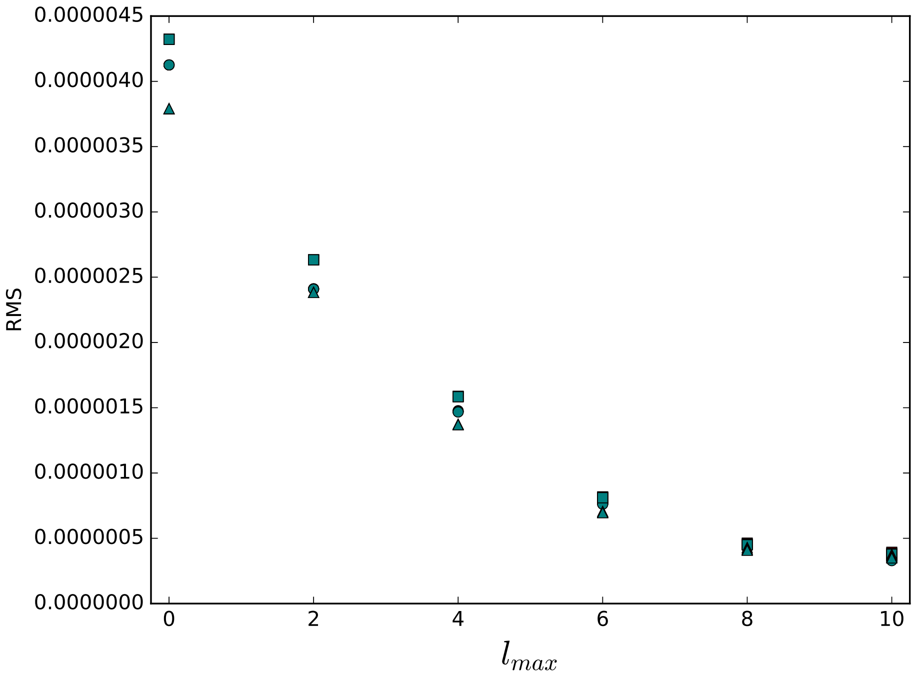
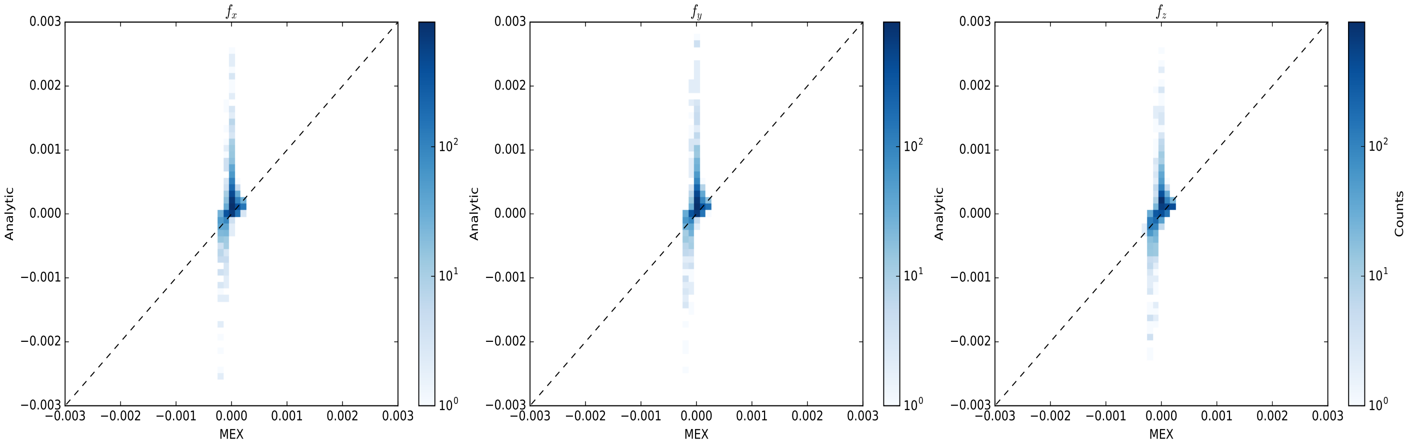


Figure 3: RMS vs. l_{max} for the toy model, circle points are RMS_x, square points are RMS_y and triangular points are RMS_z.

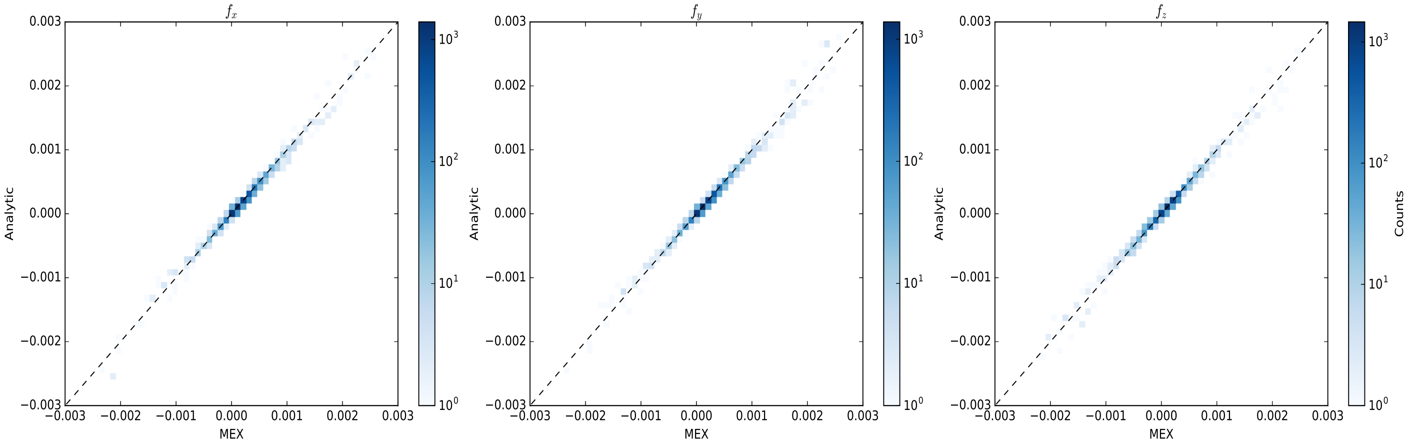
Looking at Fig. 3 it is clear that as l_{max} increases there is a strong improvement to the RMS value for each component. As the number of bins increases there is no discernible difference to the performance of the algorithm, but looking at the values in Table 1 there is a slight improvement as the number of bins increases. The lack of substantial improvement observed as the number of bins is increased may be explained by the toy model being used is uniform in nature and lacks any substructure that would have a substantial impact on the forces calculated.

To visually see the improvement between the lowest and the highest RMS values, $\boldsymbol{f}_{\text{MEX}}$ and $\boldsymbol{f}_{\text{Analytic}}$ can be plotted against each other for each component of force. The highest RMS value is for the combination $l_{max} = 0$ with 200 radial bins, while the lowest is for $l_{max} = 10$ with 400 radial bins.



(a) x component (b) y component (c) z component

Figure 4: $\boldsymbol{f}_{\text{MEX}}$ Vs. $\boldsymbol{f}_{\text{Analytic}}$ for an $l_{max} = 0$ with 200 radial bins. $\text{RMS}_x = 4.13 \cdot 10^{-6}, \text{RMS}_y = 4.32 \cdot 10^{-6}, \text{RMS}_z = 3.79 \cdot 10^{-6}$.



(a) x component (b) y component (c) z component

Figure 5: $\boldsymbol{f}_{\text{MEX}}$ Vs. $\boldsymbol{f}_{\text{Analytic}}$ for an $l_{max} = 10$ with 400 radial bins. $\text{RMS}_x = 3.27 \cdot 10^{-7}, \text{RMS}_y = 3.77 \cdot 10^{-7}, \text{RMS}_z = 3.49 \cdot 10^{-7}$.

The MEX approximation has very good agreement to that of the analytic toy model. To fully determine how well the program is performing it is important to look at the runtimes. As the program calculates the force in an algorithmic way through the steps listed above, each step will have its runtime value calculated independently. The runtimes were calculated on a 2015 MacBook Pro with quad core Intel i7 2.5 GHz processors for each combination of $l_{max} = 0, 2, 4, 6, 8, 10$ and number of bins of 200,250,300,350,400. The runtime values can be seen in Table 2 in the appendix.

By plotting the values in Table 2, it can be seen that step 1 has constant runtime over l_{max} and is proportional to N and m_r, step 2 is proportional to l_{max} to some power and N and m_r, step 3 is proportional to l_{max} to some power and m_r and step 4 is proportional to l_{max} to some power and n, where n is the number of points where the force was calculated.

The runtimes for each step will be fit with the following models, then normalized with respect to

N, n and m_r as appropriate. The fits will be made using the runtimes for 300 radial bins.

Step 1,

$$t_1 = C_1 \tag{32}$$

Steps 2, 3 and 4,

$$t_{2,3,4} = {l_{max}}^{a} \cdot b \tag{33}$$

After normalizing each step with the appropriate parameters the approximate runtimes for each step of the algorithm are given by,

$$t_1 = (2.80 \cdot 10^{-7})\, N\, m_r \tag{34}$$

$$t_2 = (5.86 \cdot 10^{-7})\, N\, m_r\, {l_{max}}^{1.7} \tag{35}$$

$$t_3 = (3.43 \cdot 10^{-4})\, m_r\, {l_{max}}^{1.62} \tag{36}$$

$$t_4 = (1.86 \cdot 10^{-4})\, n\, {l_{max}}^{1.7} \tag{37}$$

Now as t_1 and t_3 are small compared to t_2 and t_4,

$$t_{\text{total}} \approx t_2 + t_4 \tag{38}$$

The following plots are the runtimes for the toy model for steps 1-4 with time approximation fits. The plot for step 4 takes $n = 1$.

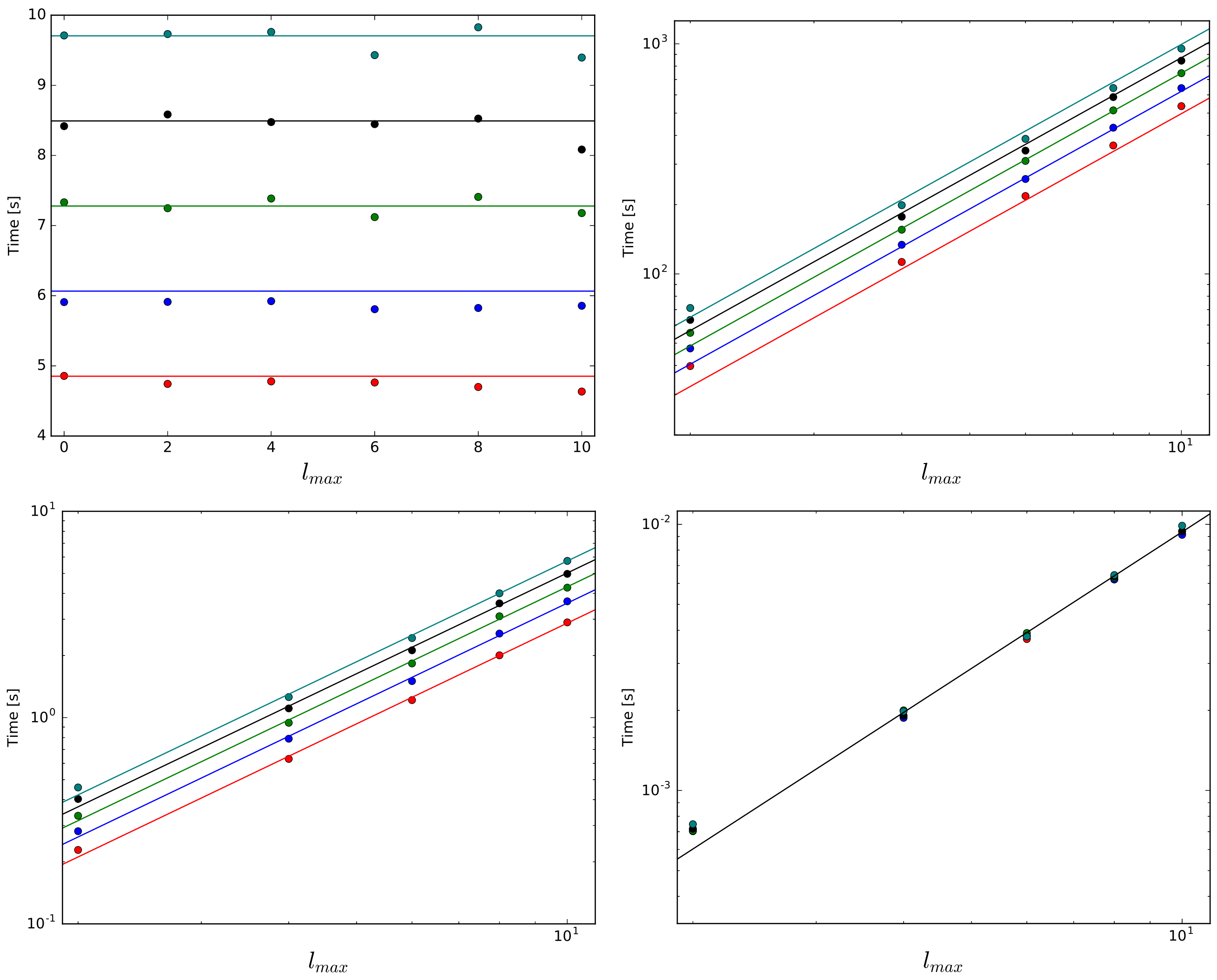


Figure 6: Runtime Vs. l_{max} for the toy model for steps 1-4. The top left is step 1, top right is step 2, bottom left is step 3 and bottom right is step 4. The red points/line is $m_r = 200$, blue points/line is $m_r = 250$, green points/line is $m_r = 300$, black points/line is $m_r = 350$ (except for fit in step 4, m_r independent), teal points/line is $m_r = 400$.

Looking at the fits of the time approximation in Fig. 6, there is good agreement between the approximated runtimes and the actual measured runtimes. Looking back at the expected algorithm complexity, steps 2-3 were proportional to ${l_{max}}^2$ while actual runtime measurements suggests its proportional to ${l_{max}}^{1.7}$. The runtime for steps 2-3 may be machine dependant and the ${l_{max}}^2$ gives an upper bound for runtime. As the ρ_{lm} basis functions calculated in step 2 are all independent of one another, that process could be threaded to improve runtime, as well as step 4 as the sum over each position does not depend on the other positions.

As there is good agreement between MEX and the analytic toy model, the next step is to compare the forces produced by MEX to that of a full N-body cody with a realistic cosmological halo. The N-body code used will be Gadget (Springel 2005). Gadget evolves self-gravitating collisionless systems using a tree code to compute the gravitational force. The cosmological halo used to compare Gadget with MEX has 684137 particles, each with a mass of $4.23 \cdot 10^5$ $\mathrm{M}_\odot$ and has visible sub structure. The plots of the halo distribution can be seen in Fig. 11 in the appendix. As MEX in its current state is only able to produce forces for a single snapshot, the forces for Gadget for a single snapshot will be exported along with the particle positions to compare with the forces produced by MEX.

To compare how varying l_{max} and m_r impact the accuracy of MEX, the RMS values will be calculated for each combination of $l_{max} = 0, 2, 4, 6$ and $m_r = 200, 250, 300, 350, 400$. The RMS values can be seen in Table 3 in the appendix.

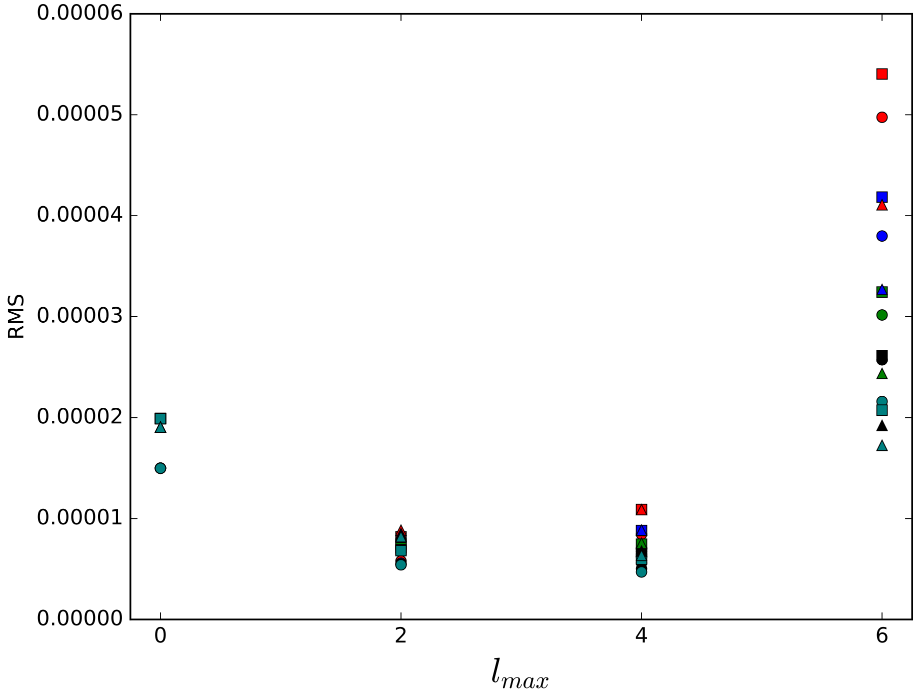
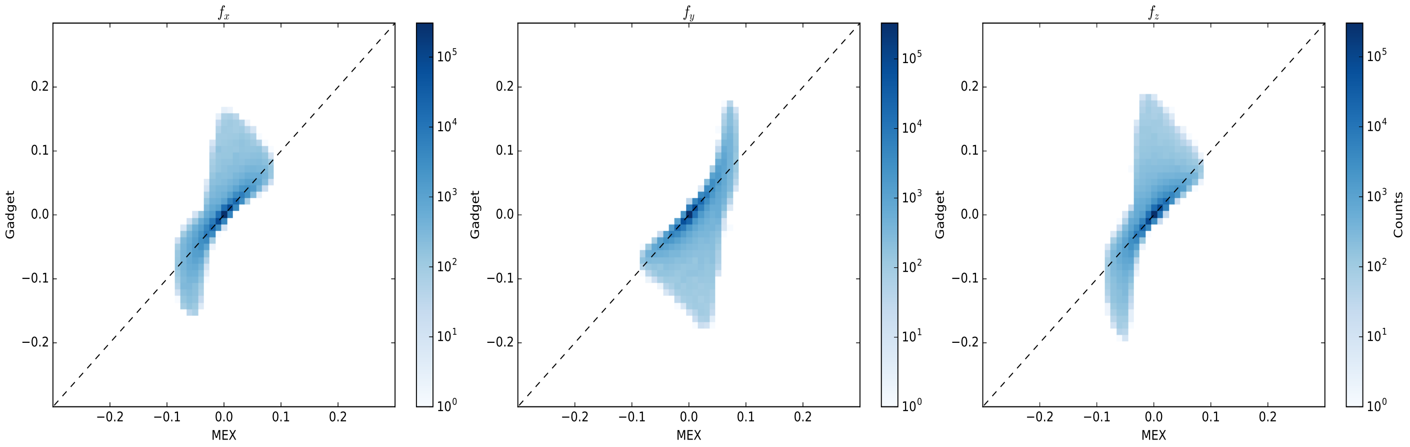


Figure 7: RMS vs. l_{max} for the cosmological halo, circle points are RMS_x, square points are RMS_y and triangular points are RMS_z. The red points are $m_r = 200$, blue points are $m_r = 250$, green points are $m_r = 300$, black points are $m_r = 350$, teal points are $m_r = 400$.

Looking at Fig. 7, it is clear that increasing l_{max} and the number of bins improves accuracy to a certain point, after that point it begins to get less accurate. By scattering $\sqrt{(|\boldsymbol{f}_{\mathrm{MEX}}| - |\boldsymbol{f}_{\mathrm{Gadget}}|)^2 / |\boldsymbol{f}_{\mathrm{Gadget}}|}$ versus r, which can be seen in Fig. 12 in the appendix, it is clear that the forces are blowing up for larger l_{max} as r goes to zero, then at further r it slightly improves for higher l_{max}. There are also peaks present in the scatter suggesting that the substructure is causing the forces in high density areas to be inaccurate. This suggests that MEX may need the introduction of some softening parameter to prevent the forces from blowing up in high density regions.

To visually see the improvement between $l_{max} = 0$ and $l_{max} = 4$, $\boldsymbol{f}_{\mathrm{MEX}}$ and $\boldsymbol{f}_{\mathrm{Gadget}}$ can be plotted against each other for each component of force.

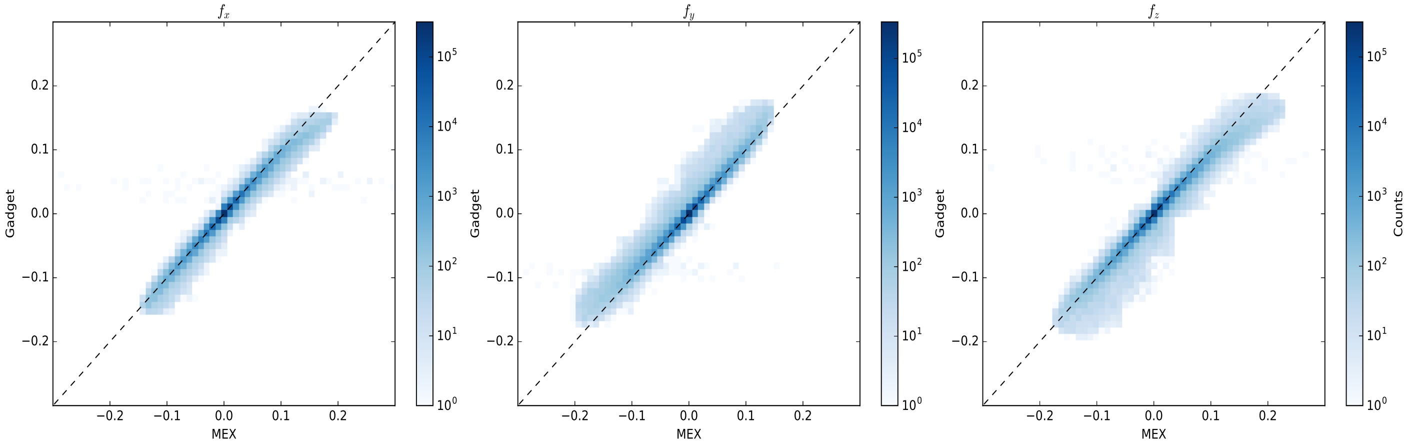


(a) x component

(b) y component

(c) z component

Figure 8: $\boldsymbol{f}_{\mathrm{MEX}}$ Vs. $\boldsymbol{f}_{\mathrm{Gadget}}$ for an $l_{max} = 0$ with 200 radial bins.$\mathrm{RMS}_x = 1.50 \cdot 10^{-5}, \mathrm{RMS}_y = 1.99 \cdot 10^{-5}, \mathrm{RMS}_z = 1.91 \cdot 10^{-5}$.



(a) x component

(b) y component

(c) z component

Figure 9: $\boldsymbol{f}_{\mathrm{MEX}}$ Vs. $\boldsymbol{f}_{\mathrm{Gadget}}$ for an $l_{max} = 4$ with 400 radial bins. $\mathrm{RMS}_x = 4.71 \cdot 10^{-6}, \mathrm{RMS}_y = 5.98 \cdot 10^{-6}, \mathrm{RMS}_z = 6.34 \cdot 10^{-6}$.

Looking at Fig. 8 and 9, it is clear that the forces are converging as l_{max} is increased. Looking closely at Fig. 9, the presence of particles that are wildly off the line can be seen, this again suggests the need for some softening parameter to prevent the blow up of forces in high density regions.

The runtimes were calculated again using the same hardware as the toy model, the runtimes can be seen in Table 4 in the appendix.

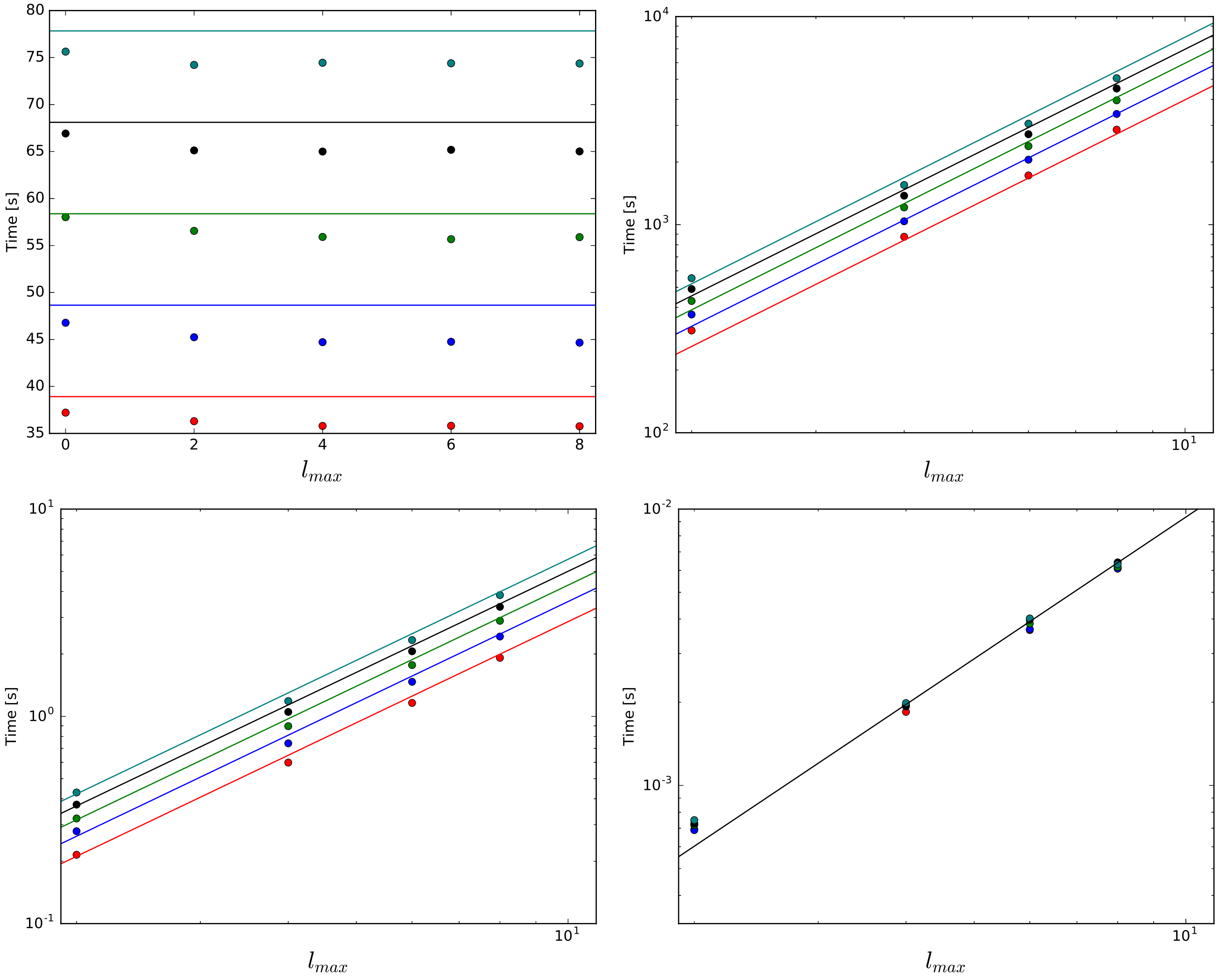


Figure 10: Runtime Vs. l_{max} for the cosmological halo for steps 1-4. The top left is step 1, top right is step 2, bottom left is step 3 and bottom right is step 4. The red points/line is $m_r = 200$, blue points/line is $m_r = 250$, green points/line is $m_r = 300$, black points/line is $m_r = 350$ (except for fit in step 4, m_r independent), teal points/line is $m_r = 400$,

Looking at Fig. 10, it is clear that the runtime models for each step from the toy model accurately scale with the number of bins and number of particles in the N-body distribution.

Discussion

Looking at the results from the toy model and cosmological halo, it is clear that the MEX approximation is working to accurately produce the forces in reasonable runtime. In order to improve the MEX approximation, a softening parameter could be introduced to eliminate the blow up of forces in high density regions.

Once a softening parameter was successfully introduced into the MEX approximation, one could begin to construct forces as functions of time by interpolating several time steps of the force and then ultimately evolving N-body systems through this force as a function of time.

Conclusion

In conclusion, the MEX approximation is able to produce the forces for N-body distributions. In order to improve accuracy a softening parameter could be introduced to eliminate error in high density regions. Once the softening parameter is successfully introduced, forces as functions of time could be created to evolve N-body systems.

References

Binney J., Tremaine S., 1987, Galactic Dynamics: First Edition. Princeton University Press.

Bode, P., Ostriker, J. P., Xu, G., 2000, ApJS, 128, 561

Griffiths D., 2012, Introduction to Electrodynamics: Fourth Edition. Pearson.

Hernquist, L., Ostriker, J. P. 1992, ApJ, 386, 375

Meiron, Y., Li, B., Holley-Bockelmann, K., Spurzem, R. 2014, (arXiv:1406.4254)

Springel V., 2005, MNRAS, 364, 1105

Stewart J., 2015, Calculus, Early Transcendentals: Eighth Edition. Brooks Cole.

Thornton S., Marion J., 2003, Classical Dynamics of Particles and Systems: Fifth Edition. Brooks Cole.

Appendix (In Order of Reference)

Derivation of Force from Potential

The forces per unit mass are related to the potential per unit mass by (Thornton S., Marion J., 2003)

$$\boldsymbol{f} = -\boldsymbol{\nabla}\Phi \tag{39}$$

Where $\boldsymbol{\nabla}$ in spherical coordinates is given by (Griffiths D., 2012)

$$\boldsymbol{\nabla} = \frac{\partial}{\partial r}\hat{\boldsymbol{r}} + \frac{1}{r}\frac{\partial}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{1}{r\sin(\theta)}\frac{\partial}{\partial \phi}\hat{\boldsymbol{\phi}} \tag{40}$$

Applying this directly to (1) will yield the forces on the particles in the $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\phi}}$ directions.

Starting with the force in the radial direction, we will need to exploit the second fundamental theorem of calculus (Stewart J., 2015)

$$\frac{\mathrm{d}}{\mathrm{d}x}\int_a^x f(t)\mathrm{d}t = f(x) \tag{41}$$

Re-writing equation (1) as

$$\Phi(r,\theta,\phi) = -4\pi G \times \sum_{l,m} \frac{Y_l^m}{2l+1}\left[\frac{1}{r^{(l+1)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da - r^l \int_\infty^r \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da\right] \tag{42}$$

Now applying the $\hat{\boldsymbol{r}}$ component of $\boldsymbol{\nabla}$

$$f_r = -\frac{\partial}{\partial r}\Phi(r,\theta,\phi) = 4\pi G \times \sum_{l,m} \frac{Y_l^m}{2l+1}\Bigg[\frac{-(l+1)}{r^{(l+2)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da \\ + r\rho_{lm}(r) + lr^{(l-1)}\int_r^\infty \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da - r\rho_{lm}(r)\Bigg] \tag{43}$$

Which then simplifies too

$$f_r = 4\pi G \times \sum_{l,m} \frac{Y_l^m}{2l+1}\left[\frac{-(l+1)}{r^{(l+2)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da + lr^{(l-1)}\int_r^\infty \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da\right] \tag{44}$$

Applying the $\hat{\boldsymbol{\phi}}$ component of $\boldsymbol{\nabla}$ to (1) allows f_ϕ to be found.

$$f_\phi = -\frac{1}{r\sin(\theta)}\frac{\partial}{\partial\phi}\Phi(r,\theta,\phi) = 4\pi G \times \sum_{l,m} \frac{\frac{\partial Y_l^m}{\partial\phi}}{2l+1}\frac{1}{r\sin(\theta)}\left[\frac{1}{r^{(l+1)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da - r^l\int_\infty^r \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da\right] \tag{45}$$

Now as $\frac{\partial Y_l^m}{\partial\phi} = im\,Y_l^m$

$$f_\phi = 4\pi G \times \sum_{l,m} im\frac{Y_l^m}{2l+1}\frac{1}{r\sin(\theta)}\left[\frac{1}{r^{(l+1)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da - r^l\int_\infty^r \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da\right] \tag{46}$$

Applying the $\hat{\boldsymbol{\theta}}$ component of $\boldsymbol{\nabla}$ to (1) allows f_θ to be found.

$$f_\theta = -\frac{1}{r}\frac{\partial}{\partial\theta}\Phi(r,\theta,\phi) = 4\pi G \times \sum_{l,m} \frac{\frac{\partial Y_l^m}{\partial\theta}}{2l+1}\frac{1}{r}\left[\frac{1}{r^{(l+1)}}\int_0^r \rho_{lm}(a)a^{(l+2)}\,da - r^l\int_\infty^r \rho_{lm}(a)\frac{1}{a^{(l-1)}}\,da\right] \tag{47}$$

Where $\frac{\partial Y_l^m}{\partial\theta}$ is defined to be [4]

$$\frac{\partial Y_l^m}{\partial\theta} = m\,\cot(\theta)\,Y_l^m + \sqrt{(l-m)(l+m+1)}\,e^{-i\phi}\,Y_l^{m+1} \tag{48}$$

Now $\boldsymbol{f}$ is defined as

$$\boldsymbol{f} = f_r\hat{\boldsymbol{r}} + f_\theta\hat{\boldsymbol{\theta}} + f_\phi\hat{\boldsymbol{\phi}} \tag{49}$$

Applying the following coordinate transformation the force per unit mass can be found in cartesian coordinates (Griffiths D., 2012)

$$\begin{cases} \hat{\boldsymbol{r}} = \sin(\theta)\cos(\phi)\hat{\boldsymbol{x}} + \sin(\theta)\sin(\phi)\hat{\boldsymbol{y}} + \cos(\theta)\hat{\boldsymbol{z}} \\ \hat{\boldsymbol{\theta}} = \cos(\theta)\cos(\phi)\hat{\boldsymbol{x}} + \cos(\theta)\sin(\phi)\hat{\boldsymbol{y}} - \sin(\theta)\hat{\boldsymbol{z}} \\ \hat{\boldsymbol{\phi}} = -\sin(\phi)\hat{\boldsymbol{x}} + \cos(\phi)\hat{\boldsymbol{y}} \end{cases} \tag{50}$$

[4] Link to definition of spherical harmonic derivatives: `http://functions.wolfram.com/Polynomials/SphericalHarmonicY/20/01/01/0001/`

Substituting in for $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\phi}}$, the components of the force per units mass in cartesian coordinates are:

$$f_x = f_r \sin(\theta)\cos(\phi) + f_\theta \cos(\theta)\cos(\phi) - f_\phi \sin(\phi) \tag{51}$$

$$f_y = f_r \sin(\theta)\sin(\phi) + f_\theta \cos(\theta)\sin(\phi) + f_\phi \cos(\phi) \tag{52}$$

$$f_z = f_r \cos(\theta) - f_\theta \sin(\theta) \tag{53}$$

Derivation of ρ_{lm} Basis Functions

The derivation for the ρ_{lm} basis functions is as follows.

Starting with:

$$\rho(r,\theta,\phi) = \sum_{l,m} \rho_{lm}(r) Y_l^m(\theta,\phi) \tag{54}$$

$$\rho(r,\theta,\phi) = \sum_i m_i \delta(\vec{r_i} - \vec{r}) \tag{55}$$

by equating the above densities,

$$\sum_i \frac{m_i}{r^2} \delta(r_i - r)\delta(\cos(\theta_i) - \cos(\theta))\delta(\phi_i - \phi) = \sum_{l,m} \rho_{lm}(r) Y_l^m(\theta,\phi) \tag{56}$$

by integrating both sides and by multiplying $Y_{l'}^{m'^*}(\theta,\phi)$

$$\int_0^{2\pi}\int_0^{\pi}\int_{r_i}^{r_i+\Delta r} r^2 dr d\cos(\theta) d\phi \left[\sum_i \frac{m_i}{r^2}\delta(r_i - r)\delta(\cos(\theta_i) - \cos(\theta))\delta(\phi_i - \phi)\right] Y_{l'}^{m'^*}(\theta,\phi)$$
$$= \int_0^{2\pi}\int_0^{\pi}\int_{r_i}^{r_i+\Delta r} r^2 dr d\cos(\theta) d\phi \sum_{l,m} \rho_{lm}(r) Y_l^m(\theta,\phi) Y_l^{m^*}(\theta,\phi) \tag{57}$$

from orthogonality of spherical harmonics and delta function integral collapse

$$\sum_i m_i Y_{l'}^{m'^*}(\theta_i,\phi_i) = \int_{r_i}^{r_i+\Delta r} r^2 dr \rho_{l'm'}(r) \tag{58}$$

the integral can be approximated using the Mean Value Theorem as

$$\int_{r_i}^{r_i+\Delta r} r^2 dr \rho_{l'm'}(r) \approx r_i^2 \Delta r \rho_{l'm'}(r) \tag{59}$$

giving the result

$$\sum_i m_i Y_{l'}^{m'^*}(\theta_i,\phi_i) = r_i^2 \Delta r \rho_{l'm'}(r) \tag{60}$$

rearranging and dropping the primes give

$$\rho_{lm}(r) = \sum_i \frac{m_i Y_l^{m^*}(\theta_i,\phi_i)}{r_i^2 \Delta r} \tag{61}$$

Where r_i, m_i, θ_i, ϕ_i is the position and mass of the i^{th} particle in the logarithmic bin described by bin-centre r and bin width Δr.

Table 1: RMS values for toy model varying l_{max} and the number of radial bins for each component of force.

Bins	x	y	z
$l_{max} = 0$			
200	$4.13 \cdot 10^{-6}$	$4.32 \cdot 10^{-6}$	$3.79 \cdot 10^{-6}$
250	$4.13 \cdot 10^{-6}$	$4.32 \cdot 10^{-6}$	$3.79 \cdot 10^{-6}$
300	$4.13 \cdot 10^{-6}$	$4.32 \cdot 10^{-6}$	$3.79 \cdot 10^{-6}$
350	$4.13 \cdot 10^{-6}$	$4.32 \cdot 10^{-6}$	$3.79 \cdot 10^{-6}$
400	$4.13 \cdot 10^{-6}$	$4.32 \cdot 10^{-6}$	$3.79 \cdot 10^{-6}$
$l_{max} = 2$			
200	$2.41 \cdot 10^{-6}$	$2.63 \cdot 10^{-6}$	$2.38 \cdot 10^{-6}$
250	$2.41 \cdot 10^{-6}$	$2.63 \cdot 10^{-6}$	$2.38 \cdot 10^{-6}$
300	$2.41 \cdot 10^{-6}$	$2.63 \cdot 10^{-6}$	$2.38 \cdot 10^{-6}$
350	$2.41 \cdot 10^{-6}$	$2.63 \cdot 10^{-6}$	$2.38 \cdot 10^{-6}$
400	$2.41 \cdot 10^{-6}$	$2.63 \cdot 10^{-6}$	$2.38 \cdot 10^{-6}$
$l_{max} = 4$			
200	$1.48 \cdot 10^{-6}$	$1.59 \cdot 10^{-6}$	$1.37 \cdot 10^{-6}$
250	$1.47 \cdot 10^{-6}$	$1.59 \cdot 10^{-6}$	$1.37 \cdot 10^{-6}$
300	$1.47 \cdot 10^{-6}$	$1.59 \cdot 10^{-6}$	$1.37 \cdot 10^{-6}$
350	$1.47 \cdot 10^{-6}$	$1.58 \cdot 10^{-6}$	$1.37 \cdot 10^{-6}$
400	$1.47 \cdot 10^{-6}$	$1.58 \cdot 10^{-6}$	$1.37 \cdot 10^{-6}$
$l_{max} = 6$			
200	$7.71 \cdot 10^{-7}$	$8.18 \cdot 10^{-7}$	$7.07 \cdot 10^{-7}$
250	$7.67 \cdot 10^{-7}$	$8.14 \cdot 10^{-7}$	$7.02 \cdot 10^{-7}$
300	$7.64 \cdot 10^{-7}$	$8.13 \cdot 10^{-7}$	$7.00 \cdot 10^{-7}$
350	$7.62 \cdot 10^{-7}$	$8.11 \cdot 10^{-7}$	$6.97 \cdot 10^{-7}$
400	$7.61 \cdot 10^{-7}$	$8.11 \cdot 10^{-7}$	$6.96 \cdot 10^{-7}$
$l_{max} = 8$			
200	$4.39 \cdot 10^{-7}$	$4.63 \cdot 10^{-7}$	$4.30 \cdot 10^{-7}$
250	$4.32 \cdot 10^{-7}$	$4.57 \cdot 10^{-7}$	$4.21 \cdot 10^{-7}$
300	$4.29 \cdot 10^{-7}$	$4.53 \cdot 10^{-7}$	$4.15 \cdot 10^{-7}$
350	$4.26 \cdot 10^{-7}$	$4.51 \cdot 10^{-7}$	$4.11 \cdot 10^{-7}$
400	$4.24 \cdot 10^{-7}$	$4.50 \cdot 10^{-7}$	$4.08 \cdot 10^{-7}$
$l_{max} = 10$			
200	$3.44 \cdot 10^{-7}$	$3.95 \cdot 10^{-7}$	$3.77 \cdot 10^{-7}$
250	$3.37 \cdot 10^{-7}$	$3.85 \cdot 10^{-7}$	$3.65 \cdot 10^{-7}$
300	$3.33 \cdot 10^{-7}$	$3.82 \cdot 10^{-7}$	$3.58 \cdot 10^{-7}$
350	$3.30 \cdot 10^{-7}$	$3.79 \cdot 10^{-7}$	$3.53 \cdot 10^{-7}$
400	$3.27 \cdot 10^{-7}$	$3.77 \cdot 10^{-7}$	$3.49 \cdot 10^{-7}$

Table 2: Runtime values for toy model varying l_{max} and the number of radial bins for each step of the program, step 4 values are for one iteration ($n = 1$). All numerical values are in seconds [s].

Bins	Step 1	Step 2	Step 3	Step 4
$l_{max} = 0$				
200	4.86	4.04	$2.42 \cdot 10^{-2}$	$1.23 \cdot 10^{-4}$
250	5.91	4.91	$3.05 \cdot 10^{-2}$	$1.21 \cdot 10^{-4}$
300	7.33	5.79	$3.89 \cdot 10^{-2}$	$1.56 \cdot 10^{-4}$
350	8.42	6.67	$4.61 \cdot 10^{-2}$	$1.22 \cdot 10^{-4}$
400	9.71	7.56	$4.87 \cdot 10^{-2}$	$1.25 \cdot 10^{-4}$
$l_{max} = 2$				
200	4.74	$3.98 \cdot 10^{1}$	$2.28 \cdot 10^{-1}$	$7.16 \cdot 10^{-4}$
250	5.91	$4.75 \cdot 10^{1}$	$2.81 \cdot 10^{-1}$	$7.03 \cdot 10^{-4}$
300	7.25	$5.55 \cdot 10^{1}$	$3.34 \cdot 10^{-1}$	$7.06 \cdot 10^{-4}$
350	8.58	$6.31 \cdot 10^{1}$	$4.03 \cdot 10^{-1}$	$7.18 \cdot 10^{-4}$
400	9.73	$7.11 \cdot 10^{1}$	$4.58 \cdot 10^{-1}$	$7.47 \cdot 10^{-4}$
$l_{max} = 4$				
200	4.78	$1.13 \cdot 10^{2}$	$6.30 \cdot 10^{-1}$	$1.89 \cdot 10^{-3}$
250	5.92	$1.34 \cdot 10^{2}$	$7.90 \cdot 10^{-1}$	$1.87 \cdot 10^{-3}$
300	7.37	$1.56 \cdot 10^{2}$	$9.44 \cdot 10^{-1}$	$2.00 \cdot 10^{-3}$
350	8.48	$1.77 \cdot 10^{2}$	1.11	$1.91 \cdot 10^{-3}$
400	9.76	$1.99 \cdot 10^{2}$	1.26	$1.98 \cdot 10^{-3}$
$l_{max} = 6$				
200	4.76	$2.18 \cdot 10^{2}$	1.22	$3.71 \cdot 10^{-3}$
250	5.81	$2.59 \cdot 10^{2}$	1.50	$3.80 \cdot 10^{-3}$
300	7.12	$3.10 \cdot 10^{2}$	1.83	$3.91 \cdot 10^{-3}$
350	8.45	$3.44 \cdot 10^{2}$	2.12	$3.81 \cdot 10^{-3}$
400	9.43	$3.87 \cdot 10^{2}$	2.43	$3.79 \cdot 10^{-3}$
$l_{max} = 8$				
200	4.70	$3.62 \cdot 10^{2}$	2.00	$6.21 \cdot 10^{-3}$
250	5.83	$4.32 \cdot 10^{2}$	2.56	$6.21 \cdot 10^{-3}$
300	7.41	$5.14 \cdot 10^{2}$	3.10	$6.31 \cdot 10^{-3}$
350	8.53	$5.87 \cdot 10^{2}$	3.58	$6.38 \cdot 10^{-3}$
400	9.83	$6.43 \cdot 10^{2}$	4.00	$6.45 \cdot 10^{-3}$
$l_{max} = 10$				
200	4.63	$5.36 \cdot 10^{2}$	2.89	$9.19 \cdot 10^{-3}$
250	5.86	$6.42 \cdot 10^{2}$	3.66	$9.13 \cdot 10^{-3}$
300	7.18	$7.45 \cdot 10^{2}$	4.26	$9.42 \cdot 10^{-3}$
350	8.08	$8.45 \cdot 10^{2}$	4.97	$9.42 \cdot 10^{-3}$
400	9.40	$9.54 \cdot 10^{2}$	5.75	$9.88 \cdot 10^{-3}$

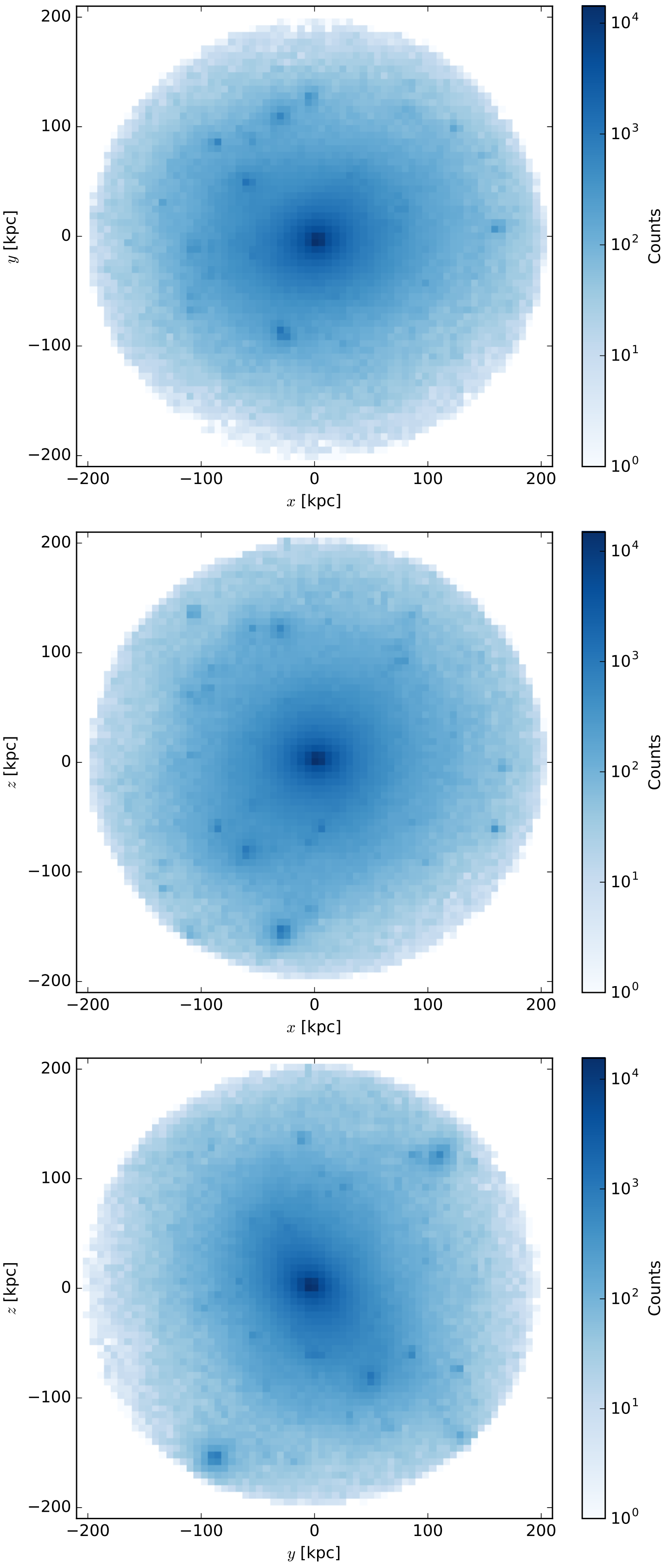


Figure 11: Cosmological halo distribution in the $x - y$, $x - z$ and $y - z$ planes.

Table 3: RMS values for the cosmological halo varying l_{max} and the number of radial bins for each component of force.

Bins	x	y	z
$l_{max} = 0$			
200	$1.50 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$
250	$1.50 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$
300	$1.50 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$
350	$1.50 \cdot 10^{-6}$	$1.99 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$
400	$1.50 \cdot 10^{-6}$	$1.99 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$
$l_{max} = 2$			
200	$5.79 \cdot 10^{-6}$	$8.19 \cdot 10^{-6}$	$8.85 \cdot 10^{-6}$
250	$5.58 \cdot 10^{-6}$	$7.47 \cdot 10^{-6}$	$8.50 \cdot 10^{-6}$
300	$5.48 \cdot 10^{-6}$	$7.11 \cdot 10^{-6}$	$8.33 \cdot 10^{-6}$
350	$5.43 \cdot 10^{-6}$	$6.91 \cdot 10^{-6}$	$8.25 \cdot 10^{-6}$
400	$5.41 \cdot 10^{-6}$	$6.81 \cdot 10^{-6}$	$8.20 \cdot 10^{-6}$
$l_{max} = 4$			
200	$8.44 \cdot 10^{-6}$	$1.09 \cdot 10^{-5}$	$1.09 \cdot 10^{-5}$
250	$6.74 \cdot 10^{-6}$	$8.81 \cdot 10^{-6}$	$8.85 \cdot 10^{-6}$
300	$5.60 \cdot 10^{-6}$	$7.46 \cdot 10^{-6}$	$7.55 \cdot 10^{-6}$
350	$5.03 \cdot 10^{-6}$	$6.62 \cdot 10^{-6}$	$6.77 \cdot 10^{-6}$
400	$4.71 \cdot 10^{-6}$	$5.98 \cdot 10^{-6}$	$6.34 \cdot 10^{-6}$
$l_{max} = 6$			
200	$4.98 \cdot 10^{-5}$	$5.40 \cdot 10^{-5}$	$4.11 \cdot 10^{-5}$
250	$3.80 \cdot 10^{-5}$	$4.18 \cdot 10^{-5}$	$3.27 \cdot 10^{-5}$
300	$3.02 \cdot 10^{-5}$	$3.24 \cdot 10^{-5}$	$2.43 \cdot 10^{-5}$
350	$2.57 \cdot 10^{-5}$	$2.61 \cdot 10^{-5}$	$1.92 \cdot 10^{-5}$
400	$2.16 \cdot 10^{-5}$	$2.07 \cdot 10^{-5}$	$1.75 \cdot 10^{-5}$

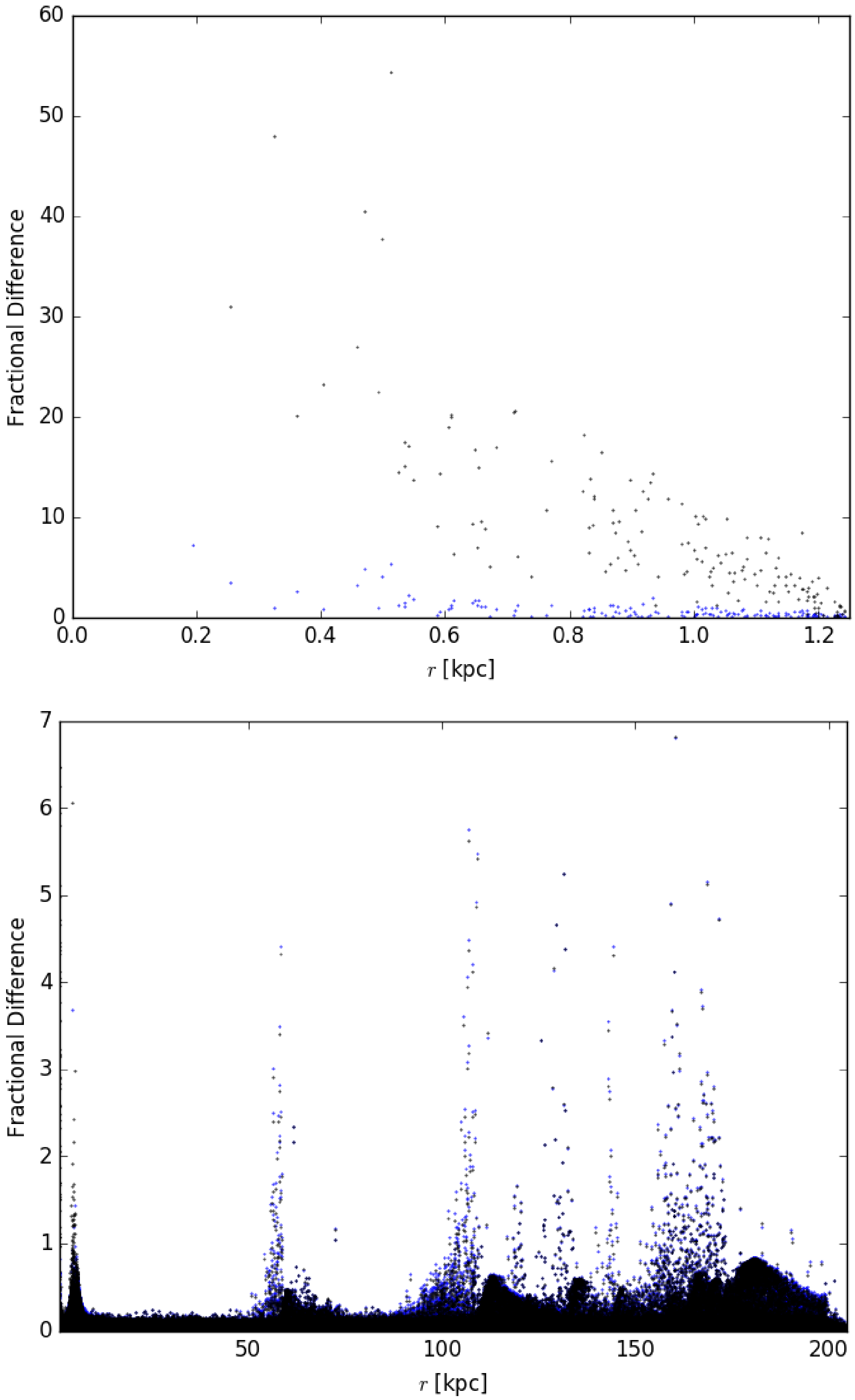


Figure 12: $\sqrt{(|\boldsymbol{f}_{\rm MEX}| - |\boldsymbol{f}_{\rm Gadget}|)^2}/|\boldsymbol{f}_{\rm Gadget}|$ versus r for the cosmological halo. $l_{max} = 4$ are the blue points while $l_{max} = 6$ are the black points. $m_r = 400$ for both. Top plot is $0 < r < 1.25$,while the bottom plot is $r > 1.25$.

Table 4: Runtime values for the cosmological halo varying l_{max} and the number of radial bins for each step of the program, step 4 values are for one iteration ($n = 1$). All numerical values are in seconds [s].

Bins	Step 1	Step 2	Step 3	Step 4
$l_{max} = 0$				
200	$3.72 \cdot 10^1$	$3.24 \cdot 10^1$	$2.42 \cdot 10^{-2}$	$1.26 \cdot 10^{-4}$
250	$4.67 \cdot 10^1$	$3.97 \cdot 10^1$	$3.00 \cdot 10^{-2}$	$1.19 \cdot 10^{-4}$
300	$5.80 \cdot 10^1$	$4.73 \cdot 10^1$	$3.96 \cdot 10^{-2}$	$1.52 \cdot 10^{-4}$
350	$6.69 \cdot 10^1$	$5.18 \cdot 10^1$	$4.13 \cdot 10^{-2}$	$1.21 \cdot 10^{-4}$
400	$7.56 \cdot 10^1$	$5.86 \cdot 10^1$	$4.84 \cdot 10^{-2}$	$1.23 \cdot 10^{-4}$
$l_{max} = 2$				
200	$3.63 \cdot 10^1$	$3.10 \cdot 10^2$	$2.15 \cdot 10^{-1}$	$6.93 \cdot 10^{-4}$
250	$4.52 \cdot 10^1$	$3.70 \cdot 10^2$	$2.79 \cdot 10^{-1}$	$6.89 \cdot 10^{-4}$
300	$5.66 \cdot 10^1$	$4.30 \cdot 10^2$	$3.21 \cdot 10^{-1}$	$7.19 \cdot 10^{-4}$
350	$6.51 \cdot 10^1$	$4.90 \cdot 10^2$	$3.75 \cdot 10^{-1}$	$7.28 \cdot 10^{-4}$
400	$7.42 \cdot 10^1$	$5.53 \cdot 10^2$	$4.29 \cdot 10^{-1}$	$7.49 \cdot 10^{-4}$
$l_{max} = 4$				
200	$3.58 \cdot 10^1$	$8.74 \cdot 10^2$	$5.98 \cdot 10^{-1}$	$1.85 \cdot 10^{-3}$
250	$4.47 \cdot 10^1$	$1.04 \cdot 10^3$	$7.41 \cdot 10^{-1}$	$1.93 \cdot 10^{-3}$
300	$5.59 \cdot 10^1$	$1.21 \cdot 10^3$	$8.97 \cdot 10^{-1}$	$1.94 \cdot 10^{-3}$
350	$6.50 \cdot 10^1$	$1.38 \cdot 10^3$	1.05	$1.95 \cdot 10^{-3}$
400	$7.44 \cdot 10^1$	$1.55 \cdot 10^3$	1.18	$1.99 \cdot 10^{-3}$
$l_{max} = 6$				
200	$3.58 \cdot 10^1$	$1.72 \cdot 10^3$	1.16	$3.65 \cdot 10^{-3}$
250	$4.48 \cdot 10^1$	$2.05 \cdot 10^3$	1.47	$3.67 \cdot 10^{-3}$
300	$5.57 \cdot 10^1$	$2.38 \cdot 10^3$	1.77	$3.85 \cdot 10^{-3}$
350	$6.52 \cdot 10^1$	$2.72 \cdot 10^3$	2.06	$3.98 \cdot 10^{-3}$
400	$7.44 \cdot 10^1$	$3.05 \cdot 10^3$	2.33	$4.02 \cdot 10^{-3}$
$l_{max} = 8$				
200	$3.58 \cdot 10^1$	$2.86 \cdot 10^3$	1.92	$6.12 \cdot 10^{-3}$
250	$4.47 \cdot 10^1$	$3.41 \cdot 10^3$	2.43	$6.09 \cdot 10^{-3}$
300	$5.59 \cdot 10^1$	$3.96 \cdot 10^3$	2.89	$6.17 \cdot 10^{-3}$
350	$6.50 \cdot 10^1$	$4.51 \cdot 10^3$	3.38	$6.42 \cdot 10^{-3}$
400	$7.44 \cdot 10^1$	$5.06 \cdot 10^3$	3.85	$6.31 \cdot 10^{-3}$