Developing a particle mesh code for Modified Newtonian Dynamics

by

Joost de Nijs

to obtain the degree of Bachelor of Science at the Delft University of Technology,
To be defended publicly on Thursday June 29, 2023.

Student number: 5270464

Thesis committee:	Dr. P.M. Visser	Supervisor
	Dr. S.W.H. Eijt	Supervisor
	Dr. A.R. Akhmerov	
	Dr. B. van den Dries	

TU Delft

Abstract

The rotational velocities of stars in galaxies indicate that either Newtonian gravity breaks down on this scale, or that there is extra mass in the universe that has not been observed. The former is known as Modified Newtonian Dynamics (MOND), whereas the latter option is known as the dark matter paradigm. In this thesis a version of MOND is considered where the field equation for gravity is modified and becomes nonlinear. To simulate MOND, new N-body codes are needed due to the field equation not being linear anymore and in this thesis a new particle mesh method is developed. This new particle mesh method follows the same steps as the normal particle mesh method, except at the step where the acceleration field on the grid is calculated. This step is replaced by an iterative method to find the MONDian acceleration field on the grid. The code was then tested in the deep MOND regime using four cases of which analytical solutions of the MOND Poisson equation are known. These cases are: linear motion, the two-body problem, a ring system and an isothermal sphere. Using these test cases, numerous conclusions were drawn. From the linear motion test it was found that in the code bodies can interact with themselves, resulting in non-physical accelerations. The two-body test showed that the magnitude of the relative error in the force as calculated by the particle mesh code is roughly on the same order as the same error when calculating the Newtonian force using a particle mesh code. Furthermore, all tests showed that energy is conserved quite well, even though momentum need not be conserved due to the self interactions, and that generally the conservation of angular momentum is violated by the code. The tests also showed that the analytical formulae that were derived for these systems gave largely the same predictions as the algorithm, hence verifying both the formulae as the algorithm. The code could be sped up by writing it in C, the accuracy can be improved by reducing self-interactions and it can be extended by adding a method to handle close encounters. Some applications of the code are given, including: simulating the solar system, wide binary stars, tidal dwarf galaxies and galaxy clusters.

Contents

1 Introduction

1.1. Dark matter and MOND

Figure 1.1: The Pinwheel galaxy, an example of a spiral galaxy. This picture was made NASA using 51 exposures of the Hubble telescope.

Rotational velocities of matter in orbit around a large mass in galaxies have challenged Newtonian gravity. Newtonian gravity namely, predicts that the rotational velocity of matter in orbit around a large mass should decrease with the distance between the two masses. To visualize this, one can plot the observed velocities versus the distance between the masses. Such a graph is called a rotation curve. In the 1970's V. Rubin made rotation curves for spiral galaxies, which are galaxies like the one shown in figure (1.1), in which star systems orbit around the center of the galaxy, and found that, contrary to expectations based on Newtonian dynamics, for large distances to the center, these curves became flat[21], as shown in figure (1.2). One possible solution to this problem is to introduce extra matter in a halo shaped region around the galaxy[18]. As this matter cannot be detected using means other than via its gravitational effect, it is called dark matter. The existence of this dark matter also solves other problems, such as the structure of the power spectrum of the cosmic microwave radiation[1]. However, there are also numerous problems with the dark matter solution, such as the missing satellite problem[5] and the cuspy halo problem[11]. Furthermore, after decades of searching for a dark matter particle, it has not been directly observed[3]. For these reasons, it is interesting to look at alternative explanations for the flat rotation curves. One such alternative is called MOdified Newtonian Dynamics (MOND). MOND was proposed in 1983 by M. Milgrom[12]. In this model, instead of adding extra mass to the universe, the laws of Newtonian dynamics are altered. This can be done in two ways. One can alter inertia, so Newton's second law: $\boldsymbol{F} = m\boldsymbol{a}$, or one can alter Newton's law of gravity:

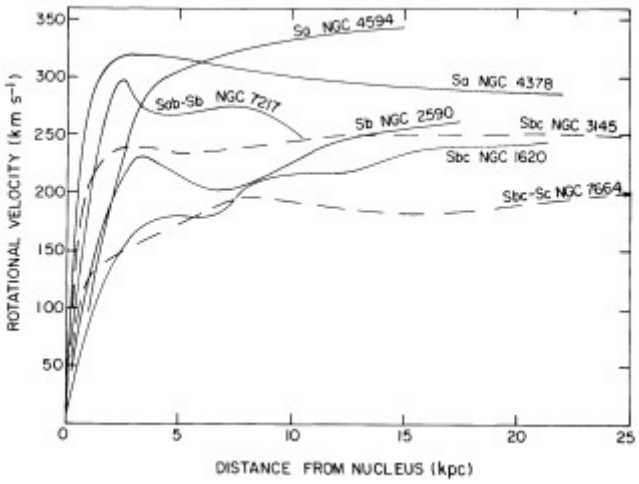


Figure 1.2: Rotation curves, showing rotational velocity of bodies as a function of radial distance to the center, for seven different galaxies[21]. It can be seen that for large distances to the center, the rotational velocity becomes approximately constant, hence the curve becomes ”flat”.

$\boldsymbol{F} = \frac{GMm}{r^2}\hat{r}$. In this research, the second approach was taken. In this approach gravity is altered at low accelerations, in contrast to general relativity, which alters gravity when the gravitational force is strong. Specifically, when a body has total acceleration much less than $a_0 = 1.2 \cdot 10^{-10}$ m s^{-2}, it is said to be in the deep MOND regime, where standard gravity is altered, and when the total acceleration is much more than a_0, the body is in the Newtonian regime and gravity is not altered. In between these regimes, standard gravity is altered in such a way that there is a continuous transition between these regimes. As the total acceleration is considered, the so called strong equivalence principal is violated in MOND. This principle states that the internal gravitational dynamics of a system do not depend on external gravitational fields. For example, the strong equivalence principle predicts that the Newtonian dynamics of the solar system does not depend on the gravitational pull to the center of the Milky Way. In MOND, this principle is violated, as now one needs to know the total acceleration of a body to calculate the gravitational field.

1.2. Simulation method

To be able to verify the MOND model and make predictions with it, the differential equations resulting from the altered laws of Newtonian dynamics need to be solved. This can in only a few cases be done analytically. Therefore code is needed which will solve these differential equations numerically. In this research such a code was developed, using an altered particle mesh method. Other codes which solve the MOND differential equations numerically already exist, such as N-MODY[9] and RAyMOND[6]. These codes typically discretize the derivatives to obtain a matrix equation for the potential and solve these using an existing method to solve matrix equations. This is quite different from the particle mesh method used in this research, which justifies the research into this different type of N-body code, as this code might have different advantages and faults compared to the other existing codes.

The Newtonian particle mesh method was introduced by Eastwood and Hockney and is described in their book[20]. Instead of only discretizing time, space is also discretised into a mesh. This allows one to solve the Poisson equation for Newtonian gravity on this mesh and find the potential in that way. From the potential, the forces on the bodies in system are readily found. For Newtonian gravity it is considered an obsolete method, as it cannot handle close encounters well, as will also be shown in this thesis. The particle mesh method is thus replaced by the particle-particle-particle-mesh (P^{3}M) method, also described in Eastwood’s and Hockney’s book[20], which calculates the force resulting from close encounters by directly summing the contributions from each nearby particle, which are found by using Newton’s equation for the gravity between two point masses. As the Poisson equation for MONDian gravity is nonlinear, one cannot sum the contributions of each particle to find the total force, and hence the (P^{3}M) method cannot be used for MONDian gravity.

Figure 1.3: Galaxy cluster Abell 370. The effects of gravitational lensing are shown by the smeared out arcs, which are the images of background galaxies distorted by the gravity of galaxies in the foreground. The picture was taken by the Hubble telescope as a part of the Frontier Fields program.

1.3. Three applications

A typical example of a system of which the observational data is thought to contradict MOND is a galaxy cluster, which is shown in figure (1.3). Such a galaxy cluster could be interesting to simulate using a particle mesh code, as the galaxies can be simulated as point masses, which would result in only 100 to 1000 particles, which could easily be simulated using a particle mesh code. This might lead to interesting results about the amount of dark matter required to explain the observational data.

In this thesis three systems will be analyzed. These will now be introduced and examples of these systems from astronomy will be given.

1.3.1. Two-body problem

First of all is the two-body problem. This system denotes any system consisting of two bodies. In Newtonian dynamics this system is completely solved and the trajectories of the bodies are Kepler orbits. In MOND, Milgrom derived a formula for the force between the two bodies in [13], which allows

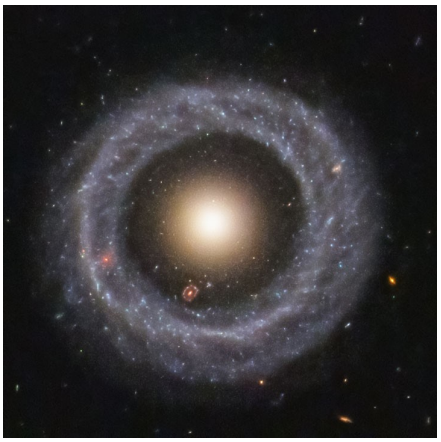


Figure 1.4: Ring galaxy called Hoag's object. The galaxy consists of an inner core consisting of stars, surrounded by an almost perfectly circularly symmetric ring of stars. A second ring galaxy can be seen inside of the ring. The picture was taken by NASA and ESA, using the Hubble telescope, and the image processing was done by Benoit Blanco.

the system to be easily analysed. The analysis of this system also has real world applications, as currently one of the most promising tests of MOND versus dark matter involves a two body system [19], namely a wide binary star system. This system consists of two stars, orbiting around their center of mass and with the orbital radius being larger than 2 kAU, such that the acceleration resulting from the force between the two stars is in the deep MOND regime. As the scale of this system is quite small in galactic terms, there is not expected to be any significant amount of dark matter affecting the orbits of these stars. Furthermore, as the stars interact with the external field generated by the center of the galaxy, MOND predicts a violation of strong equivalence principle. These systems therefore form a good testing ground for MOND, as one could model these systems both in MOND and Newtonian gravity, and compare these results with observational data to see if the strong equivalence principle has indeed been violated and if the strength of the gravitational force matches MOND or Newtonian gravity.

1.3.2. Astrophysical ring

Secondly is the ring system, which consists of a central mass with a ring of particles around it. Using the virial equation Milgrom derived a formula for the force per unit length of the ring and the central mass[15]. This system was again tested and compared with the analytical solution. A realization of a ring system in the universe is Hoag's object, shown in figure (1.4), of which it is unknown how it was created. Modeling this object using MONDian gravity might result in new insights about how this object was created, or about the validity of the MOND model, and one way to model it would be by using N-body code, such as the one described in this thesis.

1.3.3. Isothermal sphere

The last system analyzed is an isothermal sphere. This is a system of particles at fixed temperatures which are in hydrostatic equilibrium, which means that the gravity that pulls all of the particles together is exactly counteracted by the pressure of the particles. This pressure occurs because the particles are initially distributed via a certain radial mass and velocity distribution, which roughly means that the particles are packed together in a sphere, just like all of the molecules in the sun can roughly be said to be packed in a sphere. This packing in a sphere has lower entropy than a uniform distribution in space, and therefore the particles try to attain this uniform distribution, which causes pressure. This pressure is then counteracted by the gravity between the particles and thus the system is in equilibrium. This equilibrium was tested using the particle mesh code. A real world example of this system can be found in galaxy clusters. These are clusters of galaxies which are bound together by gravity. In between the galaxies, gas resides, which can be modeled as an hydrostatic isothermal gas. Currently it is unknown why this gas does not cool down, so modeling it use MOND would be interesting.

1.3.4. Non MOND use cases

As it turns out, the differential equation that will be solved using the particle mesh code, is also used to describe other physical phenomena than MONDian gravity. Milgrom lists some of these in [14]. One of these is nonlinear electric media [8]. Currently electric charges are not implemented in the code, but one might be able to alter the code to simulate nonlinear electronics as well.

1.4. Outline

To understand MOND, first some theory about MOND and how it solves the problem of flat rotation curves is described, together with a derivation of the Tully-Fisher relation and a more thorough analysis of the systems described above. Afterwards the Newtonian particle mesh method will be explained, and the changes to convert it to a MONDian particle mesh code are described. Next, the tests that were used to test the code are shown together with the results of those tests. Subsequently some recommendations are given for further research, and lastly the conclusion is given.

2 Theory

2.1. Newtonian gravity

Newton's law for gravity states that the gravitational field of a point mass can be described by:

$$\boldsymbol{g}(\boldsymbol{r}) = -\frac{GM}{r^2}\hat{r} \tag{2.1}$$

This law, however, does not hold for non-point masses. To describe situations like these, a different law can be used, called Gauss's law.

Lemma 2.1.1 (Gauss's law)**.**

$$\nabla \cdot \boldsymbol{g} = -4\pi G\rho$$

Proof. By the superposition principle, the gravitational field of a non-point mass can be found by adding up the contributions of all points in that mass, or more formally, by integrating over the object:

$$\boldsymbol{g}(\boldsymbol{r}) = -G\iiint \rho(\boldsymbol{s})\frac{(\boldsymbol{r}-\boldsymbol{s})}{|\boldsymbol{r}-\boldsymbol{s}|^3}d^3\boldsymbol{s}$$

Now the divergence operator with respect to $\boldsymbol{r}$ is applied to both sides, and using the identity

$$\nabla \cdot (\frac{\boldsymbol{r}}{|\boldsymbol{r}|^3}) = 4\pi\delta(\boldsymbol{r})$$

where δ represents the Dirac delta function, the following result is obtained:

$$\nabla \cdot \boldsymbol{g}(\boldsymbol{r}) = -4\pi G\iiint \rho(\boldsymbol{s})\delta(\boldsymbol{r}-\boldsymbol{s})d^3\boldsymbol{s}$$

which simplifies to

$$\nabla \cdot \boldsymbol{g}(\boldsymbol{r}) = -4\pi G\rho(\boldsymbol{r})$$

□

By now substituting $-\nabla\phi(\boldsymbol{r}) = \boldsymbol{g}(\boldsymbol{r})$, Poisson's equation for Newtonian gravity is obtained:

Theorem 2.1.2 (Poisson's equation)**.**

$$\nabla^2\phi(\boldsymbol{r}) = 4\pi G\rho(\boldsymbol{r})$$

Together with Newton's second law, $\boldsymbol{F} = m\boldsymbol{a}$, these two differential equations form the model of Newtonian dynamics.

2.2. Modified Newtonian dynamics

To obtain the model for MOND, Poisson's equation for gravity is changed and becomes:

$$\nabla \cdot (\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi) = 4\pi G\rho \tag{2.2}$$

In this equation ρ is the baryonic mass density, μ is a function that interpolates between the Newtonian regime, where $|\nabla\phi| \gg a_0$ and $\mu(\frac{|\nabla\phi|}{a_0}) = 1$, and the deep MOND regime, where $|\nabla\phi| \ll a_0$ and $\mu(\frac{|\nabla\phi|}{a_0}) = \frac{|\nabla\phi|}{a_0}$. This is the gravitational approach to MOND. The other differential equation, Newton's second law, stays the same in this MOND model, so inertia is not changed in this approach.

2.2.1. Rotation curves

To show that this modified Poisson's equation results in flat rotation curves for large distances to the nuclei of galaxies, it is assumed that the galaxy is spherically symmetric and the gravitational field that results from a spherically symmetric body will be derived. This will first be done for the Newtonian case, as it is needed for the MOND case. Let M be the total mass of the spherically symmetric body and let r be the distance from a point outside of the body to the center of the body, then:

Lemma 2.2.1 (Newtonian acceleration field for spherically symmetric bodies)**.**

$$\boldsymbol{g}(\boldsymbol{r}) = -\frac{GM\hat{r}}{r^2}$$

Proof. Starting with Gauss's law:

$$\nabla \cdot \boldsymbol{g} = -4\pi G\rho$$

Taking the volume integral over the volume of a ball V with radius r on both sides:

$$\iiint_V \nabla \cdot \boldsymbol{g} dV = \iiint_V -4\pi G\rho dV$$

Then the divergence theorem is applied to the left hand side and the volume integral of the density on the right hand side is recognised to simply be M

$$\iint_S \boldsymbol{g} \cdot \hat{n} dS = -4\pi GM$$

Now, as the body is spherically symmetric, the gravitational field has to be spherically symmetric as well, hence it is constant on the sphere S.

$$4\pi r^2 \boldsymbol{g} \cdot \hat{r} = -4\pi GM$$

$$\boldsymbol{g}(\boldsymbol{r}) = -\frac{GM\hat{r}}{r^2}$$

□

The Newtonian potential of a spherically symmetric body will now be called ϕ_N and the MOND potential of a spherically symmetric body will be derived.

Theorem 2.2.2 (MONDian acceleration field for spherically symmetric bodies)**.**

$$\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi = \nabla\phi_N = \frac{GM\hat{r}}{r^2}$$

Proof. The Poisson equation for MOND states:

$$\nabla \cdot \mu(\frac{|\nabla\phi|}{a_0})\nabla\phi = 4\pi G\rho = \nabla \cdot \nabla\phi_N$$

Rewriting this results in

$$\nabla \cdot (\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi - \nabla\phi_N) = 0$$

Now integrating over the volume V enclosed by a sphere with radius r:

$$\iiint_V \nabla \cdot (\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi - \nabla\phi_N)dV = 0$$

Using the divergence theorem and calling the sphere S:

$$\iint_S (\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi - \nabla\phi_N)\hat{r}dS = 0$$

As in the Newtonian case, as the body is spherically symmetric, the gravitational field has to be spherically symmetric as well, hence it is constant on the sphere and as the area of the sphere is nonzero, the integrand must be zero.

$$(\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi - \nabla\phi_N) \cdot \hat{r} = 0$$
$$\mu(\frac{|\nabla\phi|}{a_0})\nabla\phi = \nabla\phi_N = \frac{GM\hat{r}}{r^2}$$

□

In the deep MOND regime, so for $|\nabla\phi| \ll a_0$, this becomes:

$$g_{\text{MOND}}^2 = \frac{GMa_0}{r^2}$$
$$g_{\text{MOND}} = \frac{\sqrt{GMa_0}}{r} \tag{2.3}$$

To relate this to the flat rotation curves, first it is noted that the accelerations on the stars for which the rotation curve is flat are indeed much smaller than a_0. Secondly, the mass of a star is much lower than the mass of the black hole at the center of the galaxy around which it orbits, hence we can assume that the star does not alter the gravitational field. In Newtonian dynamics one does not have to worry about this at all due to the strong equivalence principle, but this does not hold anymore in MOND. Lastly, as the stars are in an orbit around the nucleus, the acceleration due to the gravity of the nucleus must be equal to the acceleration predicted by the centripetal force. Hence:

$$a_{\text{MOND}} = \frac{\sqrt{GMa_0}}{r} = \frac{v^2}{r}$$
$$v^4 = GMa_0 \tag{2.4}$$

As equation (2.4) shows that v does not depend on r, MOND indeed predicts that rotation curves become flat for large distances to the nuclei, as the accelerations on the rotating bodies are then small enough to be in the deep MOND regime.

2.2.2. Tully-Fisher relation

In [24], R. Tully and J. Fisher published a relation between the width of the hydrogen 21 cm emission line and the intrinsic luminosity of a spiral galaxy. Later this was relation was found to hold better, if instead of the intrinsic luminosity, the total baryonic matter in the galaxy was used. These two are related as an increase of baryonic matter often means an increase of the mass of stars in the galaxy, which in turn leads to an increase in luminosity. Furthermore, the width of the emission line is dependent on the Doppler effect, which is directly dependent on the asymptotic rotational velocity, the rotational velocity that the rotation curve tends to as it becomes flat.Hence the Tully-Fisher relation can also be written as a relation between the asymptotic rotational velocity and the total baryonic matter of a galaxy. The specific relation is:

$$M \propto v^\alpha \tag{2.5}$$

In this relation α has been empirically determined to be between 3.5 and 4[23]. This relation can however, be recognized from before. Indeed, equation (2.4) shows the Tully-Fisher relation for $\alpha = 4$. Thus MOND offers an explanation for this relation. The dark matter model, in contrast, has up to now not been able to explain the Tully-Fisher relation [10].

2.2.3. Virial equation

In this section the general virial equation will be proved, a result of Milgrom using the virial equation will be shown and using this some systems will be analyzed. In the following equations $\langle\rangle$ denotes a long time average, whereas the bar denotes a mass-weighted mean.

Lemma 2.2.3 (Virial equation for bound systems)**.**

$$\langle -\sum_i \boldsymbol{r}_i \cdot \boldsymbol{F}_i \rangle = \langle \sum_i m_i \dot{\boldsymbol{r}}_i^2 \rangle = \langle 2E_{\text{kin}} \rangle$$

Proof.

$$-\frac{1}{2}\frac{d^2}{dt^2}(\sum_i m_i \boldsymbol{r}_i^2) = -\frac{d}{dt}(\sum_i m_i \boldsymbol{r}_i \cdot \dot{\boldsymbol{r}}_i) = -\big(\sum_i m_i(\boldsymbol{r}_i \cdot \ddot{\boldsymbol{r}}_i + \dot{\boldsymbol{r}}_i^2)\big)$$

Rewriting this gives:

$$-\sum_i \boldsymbol{r}_i \cdot \boldsymbol{F}_i = -\frac{1}{2}\frac{d^2}{dt^2}(\sum_i m_i \boldsymbol{r}_i^2) + \sum_i m_i \dot{\boldsymbol{r}}_i^2 \tag{2.6}$$

Now for bound systems the long time average of the first term on the right vanishes, as will now be shown:

$$\langle -\frac{1}{2}\frac{d^2}{dt^2}(\sum_i m_i \boldsymbol{r}_i^2)\rangle = \lim_{t'\to\infty} \frac{\int_0^{t'} -\frac{1}{2}\frac{d^2}{dt^2}(\sum_i m_i \boldsymbol{r}_i^2)dt}{t'} = \lim_{t'\to\infty} \frac{-\frac{1}{2}\frac{d}{dt}(\sum_i m_i \boldsymbol{r}_i^2)|_{t'} + \frac{1}{2}\frac{d}{dt}(\sum_i m_i \boldsymbol{r}_i^2)|_0}{t'} = 0$$

The last equality holds as the system is bounded and therefore the numerator must be bounded. Therefore by taking the longtime average of equation (2.6), the proof is complete.

□

Now typically the left hand side of this equation can be rewritten in terms of the potential energy of the system, however, in the case of deep MOND it can only be rewritten in terms of the mass of the system. In [17], Milgrom did this and derived the following equation for bound systems of point masses in deep MOND:

$$\langle \overline{v^2} \rangle = \frac{2\sqrt{Ga_0 M}}{3}(1 - \sum_p (m_p/M)^{3/2}) \tag{2.7}$$

In this equation the sum over p is a sum over all the point masses p and $M = \sum_p m_p$ is the total mass. For systems in steady-state, this formula thus shows that the kinetic energy of the system is dependent only on the masses of the particles in the system. Now some results will be derived using the general virial equation and Milgrom's result.

Corollary 2.2.3.1 (Force between two point masses in circular orbits in deep MOND)**.** Suppose we have two point masses in circular orbit around the center of mass each other, with masses m_1 and m_2, with positions $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ and with $\boldsymbol{r_{12}} = \boldsymbol{r_2} - \boldsymbol{r_1}$. Then the force on point mass 1 is:

$$\boldsymbol{F}_{12} = \frac{2}{3}\frac{\sqrt{Ga_0}}{r_{12}}((m_1+m_2)^{3/2} - m_1^{3/2} - m_2^{3/2})\hat{r}_{12} \tag{2.8}$$

Proof. As the two point masses are in circular orbits the distance between them should be constant, therefore also the force between them as this can only depend on their distance, and the time averages are dropped. Now combining the general virial equation with Milgrom's result:

$$-\sum_i \boldsymbol{r}_i \cdot \boldsymbol{F}_i = -(\boldsymbol{r}_1 \cdot \boldsymbol{F}_{12} + \boldsymbol{r}_2 \cdot \boldsymbol{F}_{21}) = -\boldsymbol{r}_1 \cdot \boldsymbol{F}_{12} + \boldsymbol{r}_2 \cdot \boldsymbol{F}_{12} =$$

$$\boldsymbol{r}_{12} \cdot \boldsymbol{F}_{12} = M\overline{v^2} = \frac{2\sqrt{Ga_0}}{3}((m_1+m_2)^{3/2} - m_1^{3/2} - m_2^{3/2})$$

$$\boldsymbol{F}_{12} = \frac{2}{3}\frac{\sqrt{Ga_0}}{r_{12}}((m_1+m_2)^{3/2} - m_1^{3/2} - m_2^{3/2})\hat{r}_{12}$$

□

Now note that the force between two point masses in deep MOND cannot depend on their velocity, as the potential does not depend on their velocity. Therefore, for two arbitrary point masses, even if they are not in circular orbits, the force between them has to be the same as the same point masses except that their velocities are adjusted to be in circular orbits. Hence, equation (2.8) must hold for two arbitrary point masses. Now, if one calculates the acceleration on mass 2 in the limit that m_2 goes to zero, equation (2.3) will be found again.

Corollary 2.2.3.2 (Inwards force on ring of finite number of bodies)**.** Suppose we have a finite number N of bodies in a ring, with constant distance between them and equal masses m, with a central mass m_0 at the center. Let $M = Nm + m_0$ and assume that the force on the bodies in the ring points radially inward. By symmetry we can assume that the force on each body in the ring has equal magnitude. Then the force on a body in the ring is:

$$\boldsymbol{F} = -\frac{2\sqrt{Ga_0}}{3Nr}(M^{3/2} - m_0^{3/2} - Nm^{3/2})\hat{r}$$

Proof.

$$-\sum_i \boldsymbol{r}_i \cdot \boldsymbol{F}_i = -NrF = M\overline{v^2} = \frac{2\sqrt{Ga_0}}{3}(M^{3/2} - m_0^{3/2} - Nm^{3/2})$$

$$F = -\frac{2\sqrt{Ga_0}}{3Nr}(M^{3/2} - m_0^{3/2} - Nm^{3/2})$$

□

Corollary 2.2.3.3 (Velocity dispersion in thermodynamic limit)**.** Suppose we have N particles with equal masses m in hydrostatic equilibrium. Then $M = Nm$ and in the thermodynamic limit equation (2.7) reads:

$$\langle\overline{v^2}\rangle = \frac{2\sqrt{Ga_0M}}{3}$$

Proof. With N particles equation (2.7) reads:

$$\langle\overline{v^2}\rangle = \frac{2\sqrt{Ga_0M}}{3}(1 - \sum_p (1/N)^{3/2}) = \frac{2\sqrt{Ga_0M}}{3}(1 - N(1/N)^{3/2}) = \frac{2\sqrt{Ga_0M}}{3}(1 - (1/N)^{1/2})$$

. The result is now obtained by taking the limit $N \to \infty$. □

3 Method

In this chapter the methods used to do MOND simulations will be explained. First, the Newtonian particle mesh method will be covered.

3.1. Newtonian particle mesh method

The Newtonian particle mesh method solves the N-body problem by not only discretizing time, but also discretizing space, using a mesh. Then the masses in the space are distributed over the mesh points and the density is calculated. With this discretized density, the Poisson equation can be solved using Fast Fourier Transforms. The Poisson equation reads:

$$\nabla^2\phi = 4\pi G\rho \tag{3.1}$$

Then the acceleration is found by:

$$\boldsymbol{a} = -\nabla\phi \tag{3.2}$$

These accelerations are attributed to the masses, using the inverse of the interpolation used to distribute the masses over the mesh points. Once the accelerations on the masses are known, the positions and velocities of the bodies at the new time step are calculated using the Leapfrog integration method. All the steps mentioned above will now be explained in detail.

3.1.1. Distributing masses

As the Fast Fourier Transform is a Discrete Fourier Transform, the Fast Fourier Transform must be taken over a finite set of points, with equal spacings between the points. However, to confine the masses to this mesh would greatly reduce the resolution. Hence the masses must be allowed to be in between the mesh points. This means that a method must be chosen to distribute the masses over the finite mesh. At first, the cloud in cell (CIC) method was used in this research[20]. For every point mass, there are eight points in the mesh grid that together form a cube around this point mass. The cloud in cell method distributes the mass of this point mass over these eight points as follows. Let (x, y, z) be the coordinates of the point mass, let (a, b, c) be the coordinates of a vertex of the cube, let m be the mass of the point mass, and let Δx be the spacing between points in the grid. Then M will be the mass that is given to this vertex.

$$M((a,b,c),(x,y,z)) = m(1-\frac{|a-x|}{\Delta x})(1-\frac{|b-y|}{\Delta x})(1-\frac{|c-z|}{\Delta x}) \tag{3.3}$$

Using this the density grid is found by summing over all of the contributions of the point masses and then dividing by the volume of a unit cube, $(\Delta x)^3$. This method however, turned out to not smooth out the point masses enough, which led to problems with the FFT, as will be explained next.

3.1.2. Mass interpolation functions

When using low order interpolation functions, such as the one described in equation (3.3), applying the Fourier method to find the MOND accelerations, led to oscillations in the acceleration field, as shown

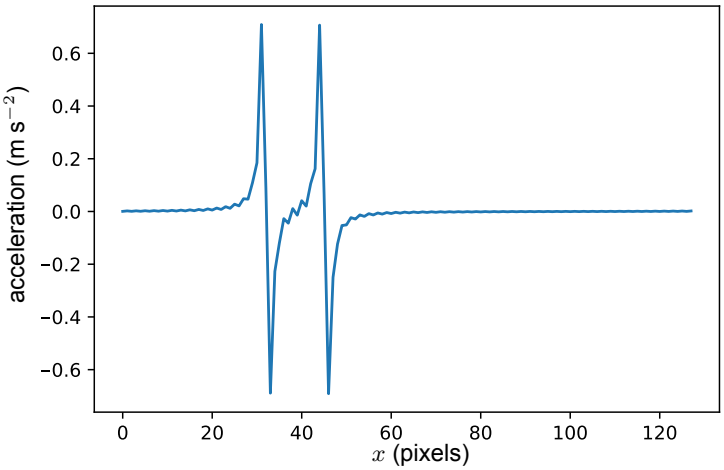


Figure 3.1: The acceleration field after partially applying the Fourier method to find the MOND accelerations is shown, along the line on which the bodies lie. The densities of the two bodies were smoothed out using the CIC method, which does not smooth out the densities enough to remove the oscillations in the acceleration field.

in figure (3.1). These oscillations are not physical, as for the two body case the accelerations should be monotone, except at the locations of the bodies, at which the resulting acceleration is not defined. The oscillations led to large errors when assigning the accelerations to the bodies. Hence a different approach was needed. One option is to use a higher order interpolation function. Then M will in general be:

$$M((a,b,c),(x,y,z)) = mW(\frac{|a-x|}{\Delta x})W(\frac{|b-y|}{\Delta x})W(\frac{|c-z|}{\Delta x}) \tag{3.4}$$

Here W is an interpolation function. As described in Eastwood and Hockney[20], we can find the W_{2n} as follows:

$$W_{2n} = \bigwedge * \bigwedge * ... * \bigwedge \tag{3.5}$$

In equation (3.5) * represents the convolution and the convolution is taken n times. $\bigwedge$ is given by:

$$\bigwedge = \begin{cases} 1-|x| & \text{if } |x|<1 \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

When using an order $2n$ interpolation function, the mass of the point mass must be distributed over the $(2n)^3$ surrounding vertices. Hence a higher order method will be more computationally expensive, as the mass has to be distributed to more vertices, which means extra computations. However, it turns out that these interpolation functions are asymptotically equal to a Gaussian, as proven in the appendix. This corresponds to the following weight function, with $\sigma^2 = \frac{n}{6}$:

$$W(x) = \sqrt{\frac{6}{2\pi n}}\exp(\frac{-x^2}{2n/6}) = \frac{1}{\sigma\sqrt{2\pi}}\exp(\frac{-x^2}{2\sigma^2}) \tag{3.7}$$

Here σ represents the standard deviation, which can be chosen freely, but should be taken larger than 1 to get sufficient smoothing. This approach using a Gaussian weight function is easier to implement and smooths the densities better than using a high order interpolation function and was hence used the most in the rest of the research.

As using (3.7) as a weight function corresponds to taking a Riemann sum of the Gaussian function, where the area under the Gaussian curve is divided into rectangles and the relative area of each rectangle is the weight corresponding to that grid point, a better approximation can be made by using the error function instead of rectangles. The difference in this approximation compared to just using a Gaussian is, however, negligible.

3.1.3. Fourier Transform

Although the Discrete Fourier Transform is used, some properties from the continuous version of the Fourier Transform are used. These will be explored first. The Fourier transform of a function f from $\mathbb{R}^3$ to $\mathbb{R}$ is defined by:

$$\mathcal{F}[f](\boldsymbol{k}) = \tilde{f}(\boldsymbol{k}) = (2\pi)^{-3/2} \iiint_{\mathbb{R}^3} f(\boldsymbol{r}) e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} \tag{3.8}$$

The inverse Fourier transform then becomes:

$$\mathcal{F}^{-1}[\tilde{f}](\boldsymbol{r}) = f(\boldsymbol{r}) = (2\pi)^{-3/2} \iiint_{\mathbb{R}^3} \tilde{f}(\boldsymbol{r}) e^{i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{k} \tag{3.9}$$

One particular identity that will be used a lot is:

Lemma 3.1.1.

$$\mathcal{F}[\nabla^2 f](\boldsymbol{k}) = -|\boldsymbol{k}|^2 \tilde{f}(\boldsymbol{r})$$

Proof.

$$\mathcal{F}[\nabla^2 f](\boldsymbol{k}) = \iiint_{\mathbb{R}^3} \nabla^2 f(\boldsymbol{r}) e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r}$$

We will now use partial integration twice and assume that $f(\boldsymbol{r})$ and $\nabla f(\boldsymbol{r})$ are zero on the boundary. Then:

$$\begin{aligned}
&\iiint_{\mathbb{R}^3} \nabla^2 f(\boldsymbol{r}) e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} = -\iiint_{\mathbb{R}^3} \nabla f(\boldsymbol{r}) \nabla e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} \\
&= \iiint_{\mathbb{R}^3} f(\boldsymbol{r}) \nabla^2 e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} = \iiint_{\mathbb{R}^3} f(\boldsymbol{r}) (-i|\boldsymbol{k}|)^2 e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} \\
&= -|\boldsymbol{k}|^2 \iiint_{\mathbb{R}^3} f(\boldsymbol{r}) e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} = -|\boldsymbol{k}|^2 \tilde{f}(\boldsymbol{k})
\end{aligned}$$

□

The Fourier transform of a function f from $\mathbb{R}^3$ to $\mathbb{R}^3$ will also be needed, and is defined by:

$$\mathcal{F}[\boldsymbol{f}](\boldsymbol{k}) = \tilde{\boldsymbol{f}}(\boldsymbol{k}) = (2\pi)^{-3/2} \iiint_{\mathbb{R}^3} \boldsymbol{f}(\boldsymbol{r}) e^{-i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{r} \tag{3.10}$$

The inverse Fourier transform is:

$$\mathcal{F}^{-1}[\tilde{\boldsymbol{f}}](\boldsymbol{r}) = \boldsymbol{f}(\boldsymbol{r}) = (2\pi)^{-3/2} \iiint_{\mathbb{R}^3} \tilde{\boldsymbol{f}}(\boldsymbol{r}) e^{i\boldsymbol{k}\cdot\boldsymbol{r}} d\boldsymbol{k} \tag{3.11}$$

3.1.4. Helmholtz decomposition

To find the MONDian accelerations in a certain system, the Helmholtz decomposition of a certain vector field will be needed. The Helmholtz decomposition of a vector field $\boldsymbol{F}$ decomposes $\boldsymbol{F}$ into its curl-free and divergence-free parts:

$$\boldsymbol{F} = \boldsymbol{F}_\mathsf{c} + \boldsymbol{F}_\mathsf{d} \tag{3.12}$$

such that:

$$\nabla \times \boldsymbol{F}_\mathsf{c} = 0 \qquad\qquad \nabla \cdot \boldsymbol{F}_\mathsf{d} = 0 \tag{3.13}$$

$\boldsymbol{F}_\mathsf{c}$ and $\boldsymbol{F}_\mathsf{d}$ can easily be found in the Fourier domain. The conditions (3.13) in the Fourier domain are namely:

$$i\boldsymbol{k} \times \tilde{\boldsymbol{F}}_\mathsf{c} = 0 \qquad\qquad i\boldsymbol{k} \cdot \tilde{\boldsymbol{F}}_\mathsf{d} = 0 \tag{3.14}$$

This can be derived in a similar fashion as the proof of lemma 3.1.1. As $\boldsymbol{F}_\mathsf{c}$ is thus parallel to $\boldsymbol{k}$:

$$\tilde{\boldsymbol{F}}_\mathsf{c} = \tilde{F}_\mathsf{c} \frac{\boldsymbol{k}}{k}$$

Furthermore, as $\boldsymbol{k} \cdot \tilde{\boldsymbol{F}}_{\mathsf{d}} = 0$:

$$\boldsymbol{k} \cdot \tilde{\boldsymbol{F}} = \boldsymbol{k} \cdot \tilde{\tilde{\boldsymbol{F}}}_{\mathsf{c}} = k\tilde{F}_{\mathsf{c}}$$

Combining this gives:

$$\tilde{\boldsymbol{F}}_{\mathsf{c}} = \frac{\boldsymbol{k} \cdot \tilde{\boldsymbol{F}}}{k^2}\boldsymbol{k} \tag{3.15}$$

Now as $\tilde{\boldsymbol{F}} = \tilde{\boldsymbol{F}}_{\mathsf{c}} + \tilde{\boldsymbol{F}}_{\mathsf{d}}$:

$$\tilde{\boldsymbol{F}}_{\mathsf{d}} = \tilde{\boldsymbol{F}} - \frac{\boldsymbol{k} \cdot \tilde{\boldsymbol{F}}}{k^2}\boldsymbol{k} \tag{3.16}$$

These two formula will be used in section 3.2 to find the MONDian accelerations in a system.

3.1.5. Fast Fourier transform

Calculating an arbitrary continuous Fourier transform is impossible, as an arbitrary integral has to be solved analytically. Therefore, the Fourier transform has to be approximated, for which the discrete Fourier transform can be used. The fast Fourier transform is an algorithm to calculate the discrete Fourier transform. Let $x_0, x_1, \ldots, x_{N-1}$ be a finite sequence of complex numbers, then its discrete Fourier transform is defined by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi k n}{N}}$$

Its inverse is:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{i\frac{2\pi k n}{N}}$$

However, as the Fourier transform will, among other things, be used to calculate the potential resulting from a certain 3 dimensional density, the Fourier transform of a sequence of 3 dimensional vectors is needed. This is defined by:

$$\boldsymbol{X_k} = \sum_{\boldsymbol{k}=0}^{\boldsymbol{N}-1} \boldsymbol{x_n} e^{-i\frac{2\pi \boldsymbol{k}\cdot\boldsymbol{n}}{N}} \tag{3.17}$$

Its inverse is:

$$\boldsymbol{x_n} = \frac{1}{N^3}\sum_{\boldsymbol{n}=0}^{\boldsymbol{N}-1} \boldsymbol{X_n} e^{i\frac{2\pi \boldsymbol{k}\cdot\boldsymbol{n}}{N}} \tag{3.18}$$

3.1.6. Acceleration field

When the masses have been assigned to the grid points, the Newtonian acceleration field can be calculated. The acceleration field is calculated from the potential, thus the potential must be found first. This is done by solving the Poisson equation in the Fourier domain. So if let the density be denoted by ρ and its Fourier transform by $\tilde{\rho}$, then the Fourier transform of the potential $\tilde{\phi}$ can be calculated as follows:

$$\tilde{\phi}(\boldsymbol{k}) = -4\pi G\frac{\tilde{\rho}}{|\boldsymbol{k}|^2} \tag{3.19}$$

Then ϕ is found by taking the inverse Fourier transform. Now the acceleration field is found by:

$$\boldsymbol{a} = -\nabla\phi \tag{3.20}$$

This can be done in the Fourier domain by multiplying by $\boldsymbol{k}$, but when using low order interpolation functions this leads to oscillations, like those shown in figure (3.1). Hence this derivative was taken using finite differences in the time domain.

3.1.7. Assigning accelerations

When the acceleration field is calculated, it is only calculated on the mesh points. Hence the accelerations on the actual bodies must still be found. This is done by using the same weights that were used to distribute the mass of the body to the neighboring grid points, as in the Newtonian case this will make sure a body does not interact with itself, due to the linearity of the Newtonian Poisson equation. So

if the mass was distributed to a cube C around the body, using an interpolation function W, then the acceleration on the body at location (x, y, z) is found by:

$$\boldsymbol{a}_{\text{body}}(x,y,z) = \sum_{(i,j,k)\in C} \boldsymbol{a}(i,j,k) W(\frac{|i-x|}{\Delta x}) W(\frac{|j-y|}{\Delta x}) W(\frac{|k-z|}{\Delta x}) \tag{3.21}$$

3.1.8. Leapfrog

To update the positions and velocities of the bodies after a time step, Leapfrog integration was used[20]. In Leapfrog integration, first the new positions are calculated based on the current velocities and accelerations, then the new accelerations are calculated and then using these new accelerations the new velocities are calculated. Let i be the current time step, then this looks as follows:

$$\begin{aligned} \boldsymbol{x_{i+1}} &= \boldsymbol{x_i} + \boldsymbol{v_i}\Delta t + \frac{1}{2}\boldsymbol{a_i}\Delta t^2 \\ \boldsymbol{v_{i+1}} &= \boldsymbol{v_i} + \frac{1}{2}(\boldsymbol{a_i} + \boldsymbol{a_{i+1}})\Delta t \end{aligned} \tag{3.22}$$

3.1.9. Complete Newtonian method

As all the parts of the Newtonian particle mesh method have now been explained, the complete method can be listed:

1. Assign the masses of the particles to grid points using equation (3.4), summing the contribution of each particle will result in the density ρ on the grid.
2. Use the FFT to Fourier transform ρ.
3. Calculate the Fourier transformed potential using equation (3.19).
4. Use the inverse FFT to transform the potential back.
5. Use finite difference to calculate the acceleration field $\boldsymbol{a}$.
6. Assign these accelerations on the grid points back to the particles using equation (3.21).
7. Calculate the new positions and velocities of the particles using equation (3.22).

3.2. MOND particle mesh method

The particle mesh method for MOND works largely the same, except for the part where the acceleration field must be calculated. Hence this part will be focused on. The Poisson equation for the MOND potential is:

$$\nabla \cdot \mu(\frac{|\nabla\phi|}{a_0})\nabla\phi = 4\pi G\rho \tag{3.23}$$

Here $\mu(x)$ is a function that interpolates between the deep MOND regime, where $\mu(x) \to x$ and the Newtonian regime, where $\mu(x) \to 1$. As this differential equation is nonlinear, it cannot be solved directly using Fourier transforms. Instead, P.M. Visser derived that the nonlinear partial differential equation can be transformed into 4 coupled linear partial differential equations and one ordinary equation. Let $\boldsymbol{g}_{\mathsf{M}} = -\nabla\phi$, then the following holds:

$$\begin{cases} \nabla \cdot \boldsymbol{g}_{\mathsf{N}} = -4\pi G\rho, & \nabla \times \boldsymbol{g}_{\mathsf{N}} = \boldsymbol{0}, & (3.24)\\ \boldsymbol{F} = \boldsymbol{g}_{\mathsf{N}} + \boldsymbol{H}, & & (3.25)\\ & \nabla \cdot \boldsymbol{H} = 0, & (3.26)\\ \boldsymbol{g}_{\mathsf{M}} = \nu\left(\dfrac{F}{a_0}\right)\boldsymbol{F}, & & (3.27)\\ & \nabla \times \boldsymbol{g}_{\mathsf{M}} = \boldsymbol{0}, & (3.28)\\ \boldsymbol{F} = \mu\left(\dfrac{g_{\mathsf{M}}}{a_0}\right)\boldsymbol{g}_{\mathsf{M}}. & & (3.29) \end{cases}$$

In this system of equations ν is the inverse interpolation function, as: $F = \mu(\frac{g_{\mathsf{M}}}{a_0})g_{\mathsf{M}} = \mu(\frac{g_{\mathsf{M}}}{a_0})\nu(\frac{F}{a_0})F$ That these are equivalent to the the original differential equation (3.23) can be seen as follows. First combine equation (3.25) and equation (3.29):

$$\boldsymbol{g}_{\mathsf{N}} + \boldsymbol{H} = \mu\left(\frac{g_{\mathsf{M}}}{a_0}\right)\boldsymbol{g}_{\mathsf{M}}$$

Then solve for $\boldsymbol{g}_{\mathsf{N}}$:

$$\boldsymbol{g}_{\mathsf{N}} = \mu\left(\frac{g_{\mathsf{M}}}{a_0}\right)\boldsymbol{g}_{\mathsf{M}} - \boldsymbol{H}$$

Now substitute this expression for $\boldsymbol{g}_{\mathsf{N}}$ in equation (3.24) and simplify using equation (3.26):

$$\nabla \cdot (\mu\left(\frac{g_{\mathsf{M}}}{a_0}\right)\boldsymbol{g}_{\mathsf{M}} - \boldsymbol{H}) = \nabla \cdot \mu\left(\frac{g_{\mathsf{M}}}{a_0}\right)\boldsymbol{g}_{\mathsf{M}} = -4\pi G\rho$$

Using $\boldsymbol{g}_{\mathsf{M}} = -\nabla\phi$, which, as the field must be conservative for a potential to exist, is only allowed if (3.28) holds, equation (3.23) is found.

Now this system of differential equations can be solved in the following manner:

1. First find the Newtonian acceleration field $\boldsymbol{g}_{\mathsf{N}}$
2. Create a second vector field filled with zero's, called $\boldsymbol{H}_d$
3. Iterate over the following steps, which is also called the main loop:
   (a) $\boldsymbol{F} = \boldsymbol{g}_{\mathsf{N}} + \boldsymbol{H}_d$
   (b) $\boldsymbol{g}_{\mathsf{M}} = \nu(\frac{F}{a_0})\boldsymbol{F}$
   (c) Let $\boldsymbol{g}_c$ be the curl-free part of $\boldsymbol{g}_{\mathsf{M}}$.
   (d) $\boldsymbol{F} = \mu(\frac{F}{a_0})\boldsymbol{g}_c$
   (e) $\boldsymbol{H} = \boldsymbol{F} - \boldsymbol{g}_{\mathsf{N}}$
   (f) Let $\boldsymbol{H}_d$ be the divergence-free part of $\boldsymbol{H}$
4. Now $\boldsymbol{g}_c$ is the MONDian acceleration field.

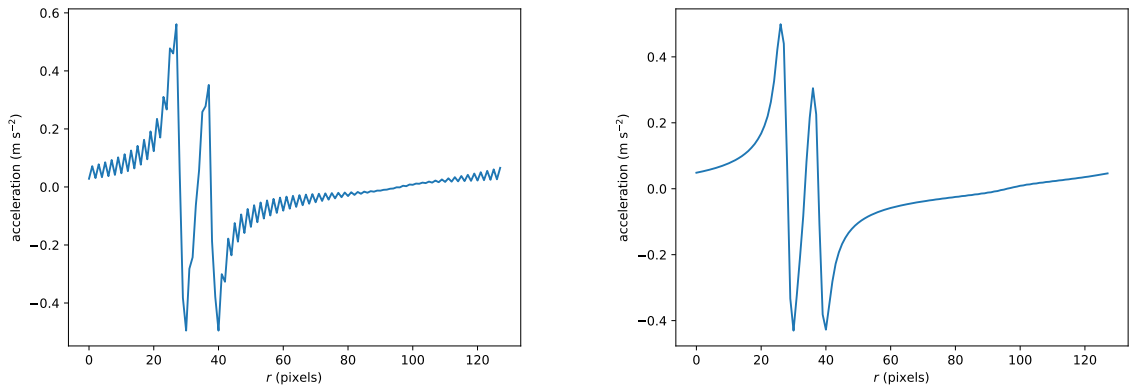
Note that the final iteration can be stopped after step (c) $\boldsymbol{g}_c$, as after this the MONDian acceleration field is not altered.

3.2.1. Additional steps

As shown in figure (3.2) iterating many times over the main loop will lead to oscillations in the acceleration field due to heavy usage of Fourier transforms. These oscillations can lead to errors when the accelerations are assigned back to the particles. To combat this the following steps can be used after the acceleration field has been found:

5. During the last iteration, stop after step (b)
6. Take the DFT of the acceleration field $\boldsymbol{g}_{\mathrm{M}}$
7. Calculate the MOND potential: $\tilde{\phi} = -i\boldsymbol{k} \cdot \tilde{\boldsymbol{g}}_{\mathrm{M}}$
8. Transform the MOND potential back
9. Calculate the acceleration field using finite difference

These additional steps will get rid of the oscillations and therefore give higher accuracy. In this research these additional steps were only used when analysing the isothermal sphere, due to time constraints.



(a) Ten iterations, not using the additional steps

(b) Ten iterations, using the additional steps

Figure 3.2: The MOND acceleration field of two bodies is shown after using ten iterations in the main loop. On the left the acceleration field is shown without using any additional steps to remove oscillations. On the right the acceleration field is shown after applying the steps mentioned in subsection (3.2.1), which clearly removes the oscillations.

4 Testing

To test the accuracy of this altered particle mesh method, some systems of which an analytical solution is known in MOND were simulated and compared to this analytical solution. Unless otherwise mentioned, the following parameters were used for the simulations:

- Distance between neighboring pixels: 10^{10} distance units
- $a_0 = 1$ distance units / time units squared
- $G = 1$ distance units cubed / (time units squared times mass units)
- Mass of lightest body: $m = 10^{20}$ mass units
- The simulations were done in deep MOND, so $\mu(x) = x$, $\nu(y) = \frac{1}{\sqrt{y}}$
- Gaussian weight functions were used, with $\sigma = 1$ and the cubes over which the masses were distributed had side lengths of 8 pixels.

4.1. One-body problem

Simulating only one body using this particle mesh shows some of the errors of this method. As there is only one point mass, there is no other object that could exert a force on this point mass, and the point mass should not accelerate. This is true in both Newtonian mechanics and MOND. When one tests this in Newtonian particle mesh code, one will indeed find that the accelerations on the point mass will be on the order of the machine precision, hence largely irrelevant. In the deep MOND regime though, the code calculates that the one body should accelerate significantly, with these accelerations being on the order of $10^{-3}\frac{\sqrt{MGa_0}}{\Delta x}$ pixels s^{-2} when Gaussian smoothing with standard deviation $\sigma = 1$ is used. Higher standard deviations, so smoother densities, lead to lower self-interactions.

4.2. Two-body problem

In contrast to the one body problem, in the two body problem, the two bodies interact, resulting in nonzero accelerations on each of the bodies. Hence, tests of the two body problem will show if the code can correctly calculate the MOND potential and the forces resulting from it.

4.2.1. Newtonian regime

The Newtonian particle mesh code was tested by reproducing Kepler orbits, as shown in figure (4.1). As the orbit produced by the particle mesh code agrees with the Kepler orbit, the Newtonian particle mesh code was deemed successful. As expected, the simulation with the smaller time step agrees better with the Kepler solution than the simulation with the larger time step.

4.2.2. Deep MOND regime

In chapter 2, a formula was derived for the force between two bodies in the deep MOND potential:

$$\boldsymbol{F}_{12} = \frac{2}{3}\frac{\sqrt{Ga_0}}{r_{12}}((m_1 + m_2)^{3/2} - m_1^{3/2} - m_2^{3/2})\hat{r}_{12} \tag{4.1}$$

In this equation $\boldsymbol{F}_{12}$ is the gravitational force from body 2 acting on body 1, r_{12} and $\hat{r}_{12}$ are the distance between the bodies and the unit vector from body 1 to body 2 respectively. Furthermore m_1 and m_2 are the masses of body 1 and body 2 respectively and lastly, G, a_0 are Newton's gravitational constant and Milgrom's constant respectively. For this force the potential energy is found to be:

$$E_{\text{pot}} = \frac{2}{3}\sqrt{Ga_0}((m_1+m_2)^{3/2} - m_1^{3/2} - m_2^{3/2})\log(r_{12}) \tag{4.2}$$

Note that this energy does not have the right units as the the logarithm of r_{12} is taken, even though r_{12} is not unitless. Dividing r_{12} by an arbitrary constant such that the logarithm is taken of a unitless variable is, however, pointless, as this only introduces an additive constant to the energy.

These exact formulae allows the numerical method to be tested and compared for the two body problem. This was done in a variety of ways. First of all, the accelerations on one of the bodies was calculated for various distances between the bodies. As this is a 1D problem, only the component of the acceleration that was in the direction of the other body was taken into account. The results of this are shown in figure (4.2). From figure (4.3) it can be seen that the dips in the relative errors are caused by the numerical solution crossing the exact solution. From these figures it can also be observed that the particle mesh method does not work well for low numbers of pixels between bodies. Furthermore one should not be deceived by the increasing error for larger distances, as the relative error is shown. A constant absolute error would show as an increasing relative error, as the magnitude of the forces decreases with distance. One can also see that the magnitude of the relative error is largely comparable to the Newtonian particle mesh method. This means that the method works quite well, as it was expected to be worse than the Newtonian particle mesh method and in reality it is only slightly worse. Lastly one can also see that the error is lower when iterating once than when iterating four times. The cause of this is unknown, but in the dynamic two body simulation it will be shown that iterating four times does work better.

Secondly, P.M. Visser derived that for certain initial values, the trajectories of the two bodies become circular orbits around the center of mass. The trajectories of the numerical solution of this particular two-body problem are shown in figure (4.4). In this simulation, the grid size was $128 \times 128 \times 128$ pixels, and the mass ratio of the two bodies was 1.5. It can be concluded that the orbits of the bodies are quite accurate, as the differences between the orbits based on Milgrom's formula and the particle mesh code are on the subpixel scale. This is also shown in figure (4.5), where the energies of this system as function of time are plotted. In these graphs, oscillations in the energy can be seen, even for the simulations using the analytical solution for the force. The cause of these oscillations is unknown, but it was determined that their frequency is $\sqrt{2}$ times the orbital frequency. As the orbital plane is perpendicular to the z-direction, there should be no force in the z-direction. Because of numerical errors and the self-interaction, however, there is a small but significant force in this direction, which causes the bodies to oscillate in the z-direction. The significance of this can be seen in the graphs of the angular momentum (4.6). These figures show that the method conserves energy quite well, even when the orbits spiral away completely from the initial orbit. Furthermore the figures show that iterating four times leads to better conservation of linear momentum than iterating only once, as the total linear momentum is what leads to the center of mass moving which is shown by the orbits spiralling away. The conservation of angular momentum seems to depend largely on the size of the time step.

Lastly, to show that the analytical formula for the force between 2 bodies in deep MOND also produces the correct trajectories for non-circular orbits, a simulation was done with the same initial conditions as the one with circular orbits, except that the initial velocities were halved. The results of this simulation are shown in figure (4.7). These results show that the analytical formula does indeed predict the same forces on the two bodies, except for the errors in the particle mesh method. The graphs for the energy and angular momentum do show, however, that the conservation of these quantities works less well than in the case of circular orbits.

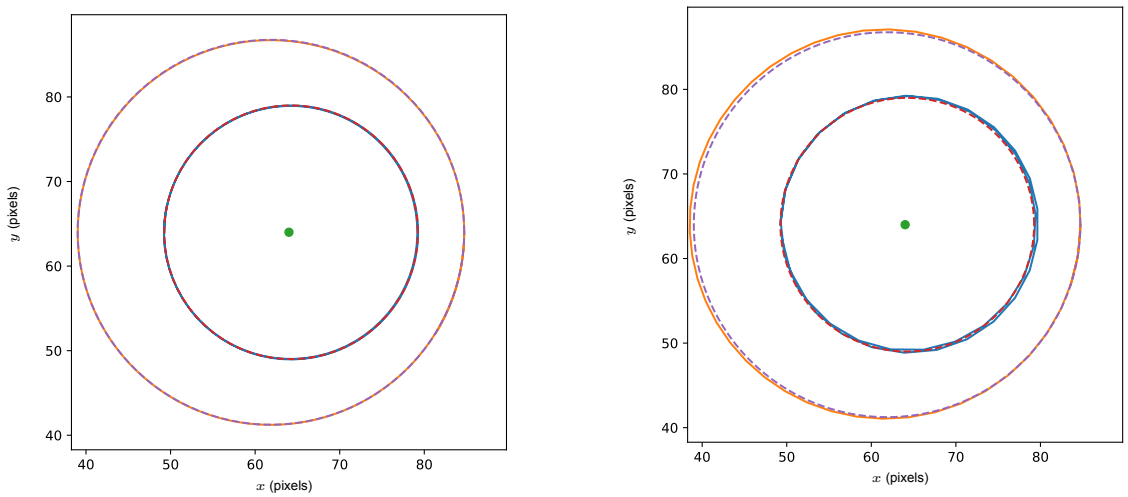


Figure 4.1: In these figures two orbits are shown of bodies in a Newtonian potential. The Kepler solution, neglecting the interaction between the two non central planets, is represented by the dotted line, whereas the solution found by the particle mesh code is shown as a solid line. The green point is the central mass. The figure on the left side shows a simulation with a time step 1/3 of the time step of the simulation shown on the right side.

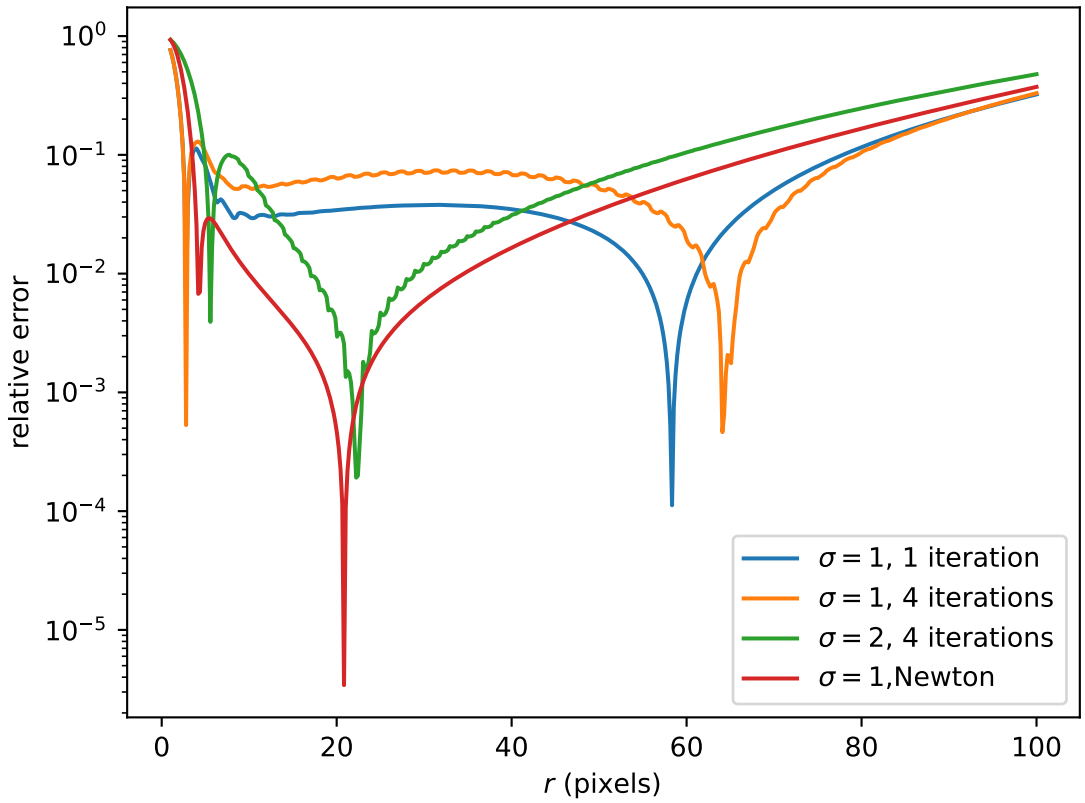
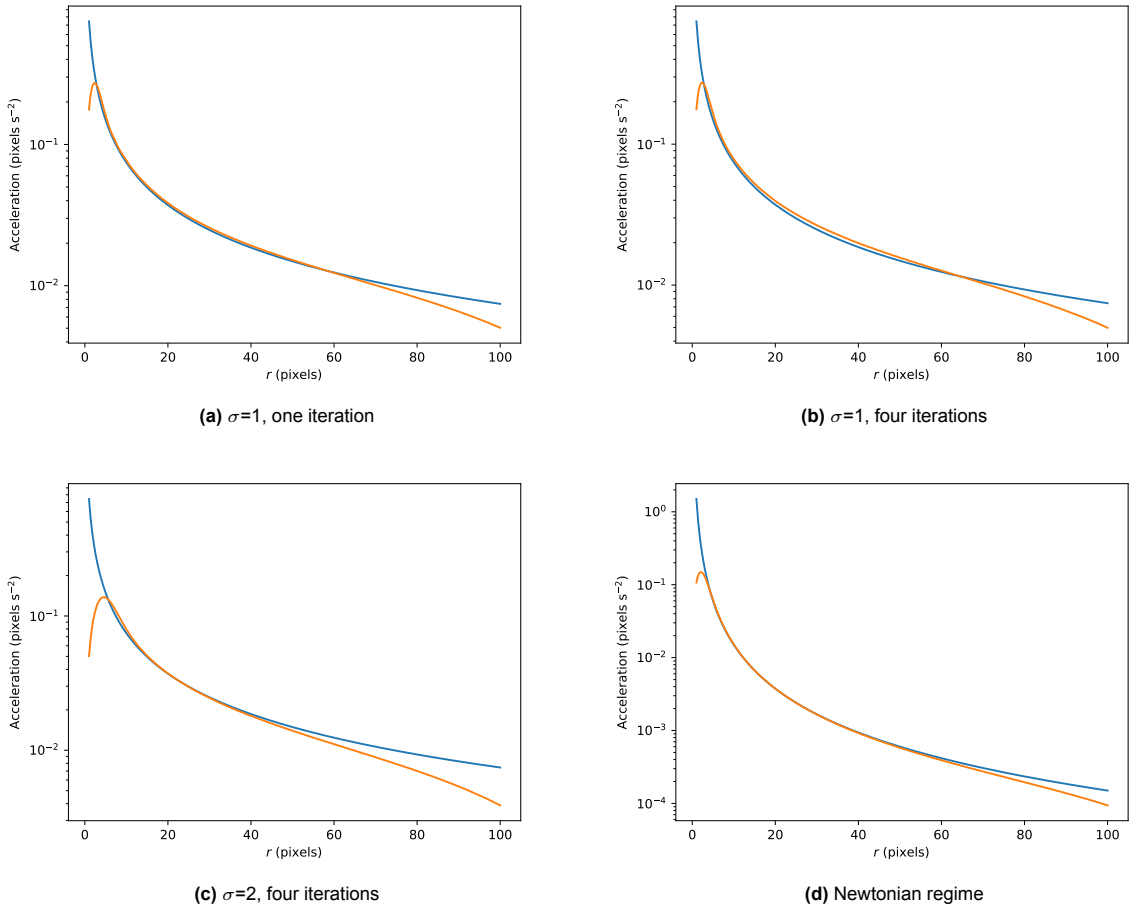


Figure 4.2: In this figure the relative errors of the accelerations on one of the bodies in a two body problem is shown. This was done three times in the deep MOND regime, and once in the Newtonian regime for comparison. The distance between two neighboring pixels was $\frac{1}{2} \cdot 10^{10}$ in this simulation, the ratio between the masses was 1.5 and the grid size was $256 \times 256 \times 256$ pixels. To obtain the errors for the deep MOND regime, the numerical solution obtained via the iterative procedure described in the method was compared to Milgrom's formula. For the Newtonian regime, the Newtonian particle mesh code was compared to Newton's formula for gravity. The errors in the deep MOND regime were obtained for two different levels of Gaussian density smoothing and two different number of iterations of the main loop. The dips in the error are at points where the numerical solution and Milgrom's formula give equal results.



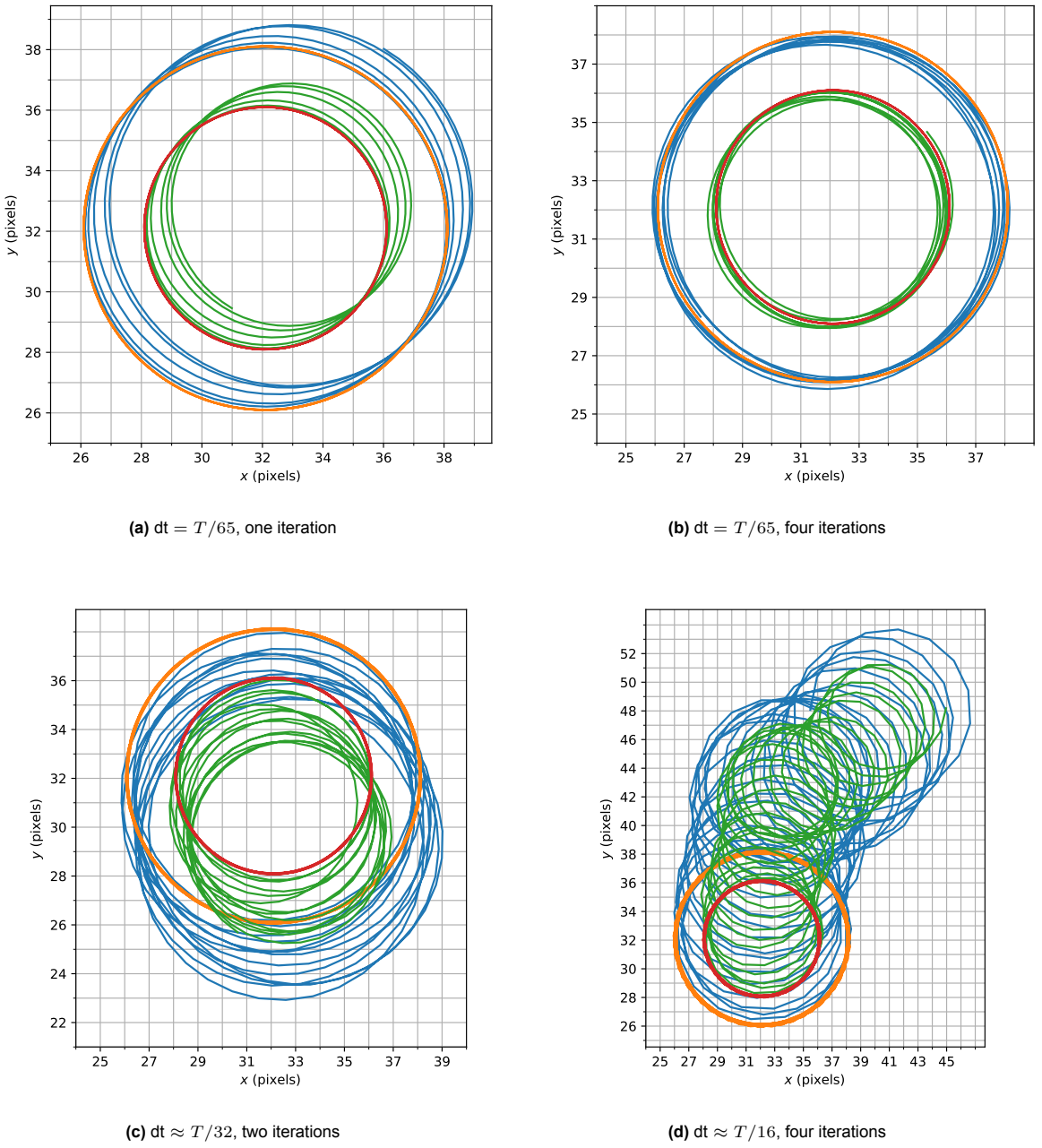
(a) σ=1, one iteration

(b) σ=1, four iterations

(c) σ=2, four iterations

(d) Newtonian regime

Figure 4.3: four accelerations on one of the bodies in the two-body problem. Only the component in the direction of the other body is shown. The first three figures show the accelerations for a deep MOND potential, whereas the last figure shows the Newtonian potential. The distance between two neighbouring pixels was $\frac{1}{2} \cdot 10^{10}$ in this simulation. The acceleration as calculated by Milgrom's formula (4.1), or Newton's formula (2.1) for the Newtonian regime, is shown in blue and the acceleration as calculated by the particle mesh code is shown in orange. For the simulations in the deep MOND regime, the simulations were executed for two different levels of Gaussian smoothing, namely the standard deviation of the Gaussian σ was changed between $\sigma = 1$ and $\sigma = 2$. The number of times that the main loop for finding the MONDian accelerations was executed was also changed between 1 iteration and four iterations.



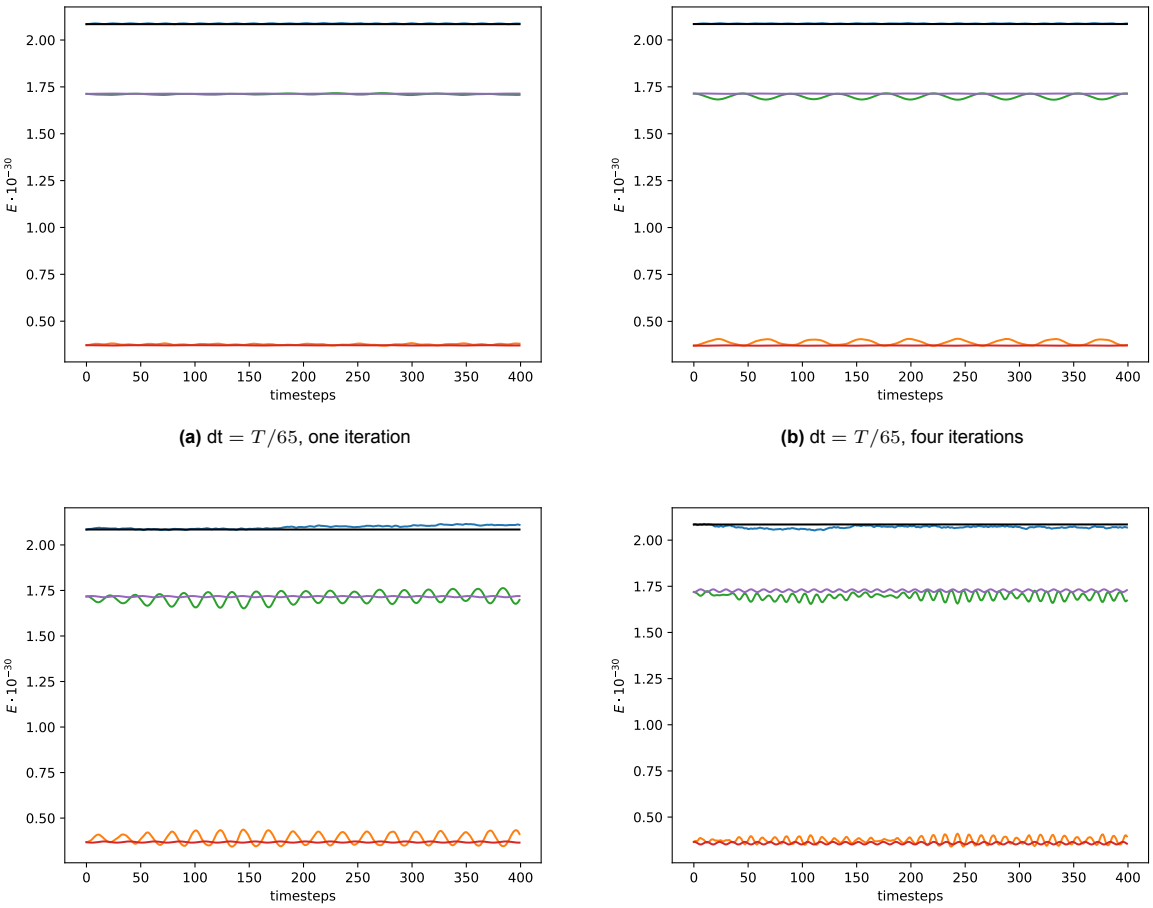
(a) dt $= T/65$, one iteration

(b) dt $= T/65$, four iterations

(c) dt $\approx T/32$, two iterations

(d) dt $\approx T/16$, four iterations

Figure 4.4: Four simulations of the two-body problem in the deep MOND regime. The simulations were done using Leapfrog integration, once using Milgrom's formula for the force between two bodies in the deep MOND regime, shown in orange and red, and once using the particle mesh method, shown in green and blue. The masses had a ratio of 1.5, the initial position and velocity vectors were chosen such that they were orthogonal at the start and with magnitudes such that circular orbits are found. The figures show simulations with different time steps and with different numbers of main loop iterations. The simulation with the smallest time step had a time step of dt=$T/65$, the simulation shown in the bottom left used twice this time step, and the simulation shown in the bottom right used four times this time step. These figures show that for small time steps and enough iterations, the particle mesh method is in agreement with Milgrom's predictions.



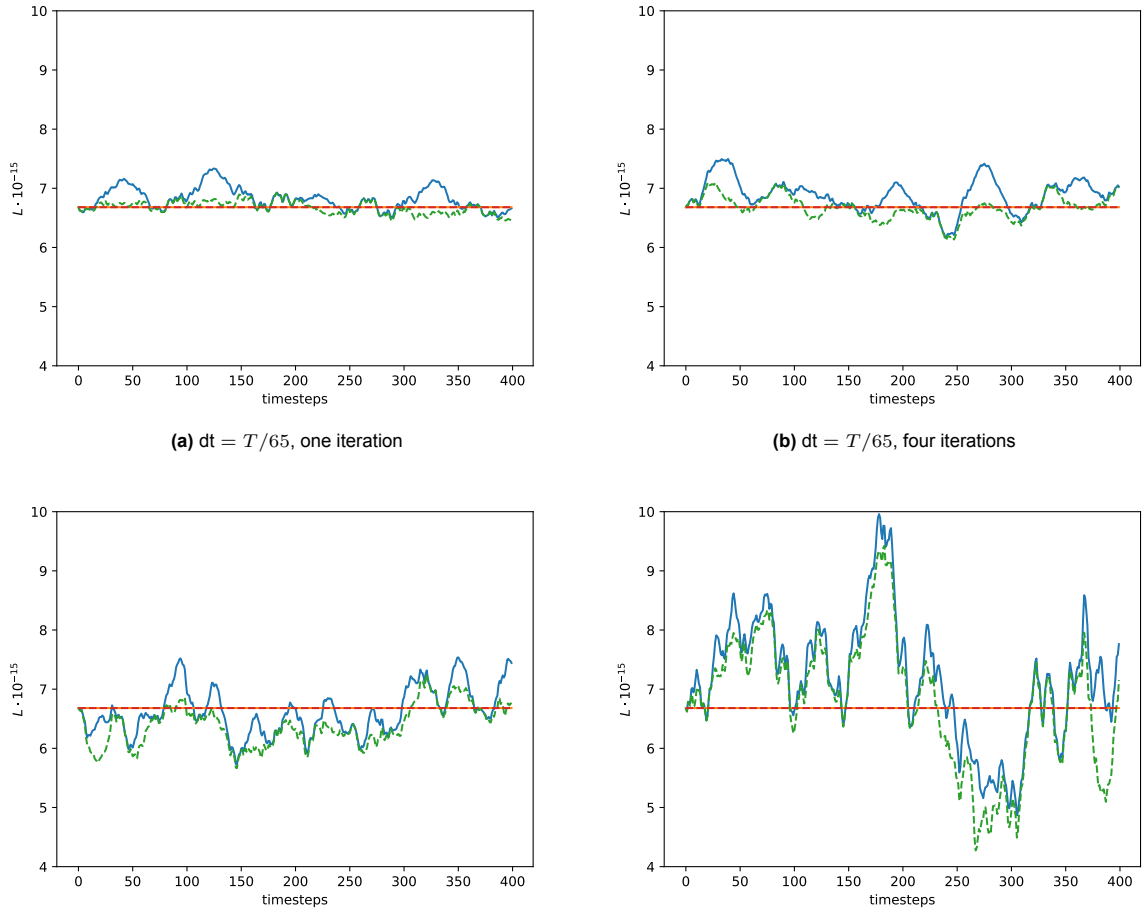
(a) dt $= T/65$, one iteration

(b) dt $= T/65$, four iterations

(c) dt $\approx T/32$, two iterations

(d) dt $\approx T/16$, four iterations

Figure 4.5: Four energy graphs of two-body simulations in deep MOND. The simulations were done using Leapfrog, once using Milgrom's formula for the force between two bodies in deep MOND 4.1 and once using the particle mesh method. The kinetic, potential and total energies are shown. For the simulations using Milgrom's formula, the total, kinetic and potential energies are shown in black, red and purple respectively. For the simulations using the particle mesh method, the total, kinetic and potential energies are shown in blue, orange and green respectively. For both of these the average was taken and is shown as a dashed line. The masses had a ratio of 1.5,the initial position and velocity vectors were chosen such that they were orthogonal at the start and with magnitudes such that circular orbits are found. The figures show simulations with different time steps and with different numbers of main loop iterations. The simulation with the smallest time step had a time step of dt=$T/65$, the simulation shown in the bottom left used twice this time step, and the simulation shown in the bottom right used four times this time step.



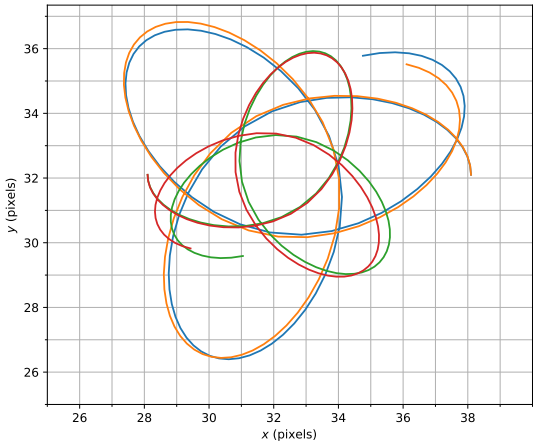
(a) dt $= T/65$, one iteration

(b) dt $= T/65$, four iterations

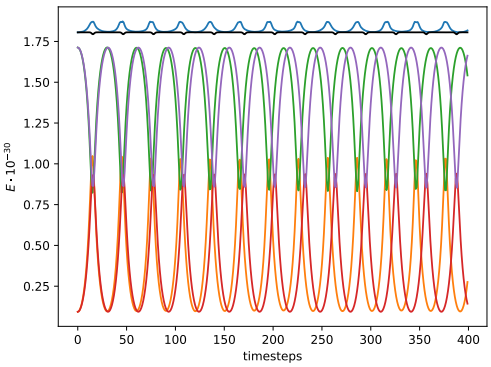
(c) dt $\approx T/32$, two iterations

(d) dt $\approx T/16$, four iterations

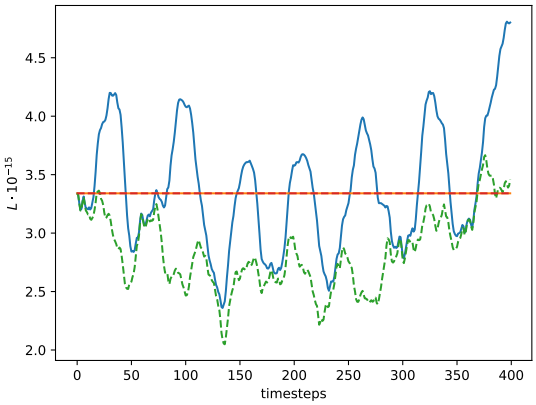
Figure 4.6: Four angular momentum graphs of two-body simulations in deep MOND. The z-component of the angular momentum is shown as a dashed line, whereas the norm of the angular momentum is shown as an continuous line. The simulations were all done using Leapfrog integration, once using the iterative particle mesh method, shown in blue and green, and once using Milgrom's formula (4.1), shown in red and orange. This graph shows that when using Milgrom's formula, the angular momentum is exactly conserved and points completely in the z-direction. When using the iterative particle mesh method though, due to errors in the found accelerations the angular momentum is not conserved any longer.



(a) Trajectories of bodies in two-body simulation in deep MOND. The simulation using Milgrom's formula is shown in orange and red, whereas the simulation using the iterative particle mesh method is shown in blue and green.



(b) Energy graph of two-body simulation in deep MOND. The kinetic, potential and total energies are shown. For the simulations using Milgrom's formula, the total, kinetic and potential energies are shown in black, red and purple respectively. For the simulations using the particle mesh method, the total, kinetic and potential energies are shown in blue, orange and green respectively.



(c) Angular momentum graph of two-body simulation in deep MOND. The z-component of the angular momentum is shown as a dashed line, whereas the norm of the angular momentum is shown as an continuous line. The simulation using Milgrom's formula is shown in orange and red, whereas the simulation using the iterative particle mesh method is shown in blue and green.

Figure 4.7: Position, energy and angular momentum graphs of a two-body simulation in deep MOND. For this simulation the initial conditions were chosen such that the bodies would not move in a circular orbit. The simulation was done once using the iterative particle mesh method and once using Milgrom's formula (4.1). A time step of dt$= 86400$ seconds was used, with four iterations per time step for the particle mesh method.

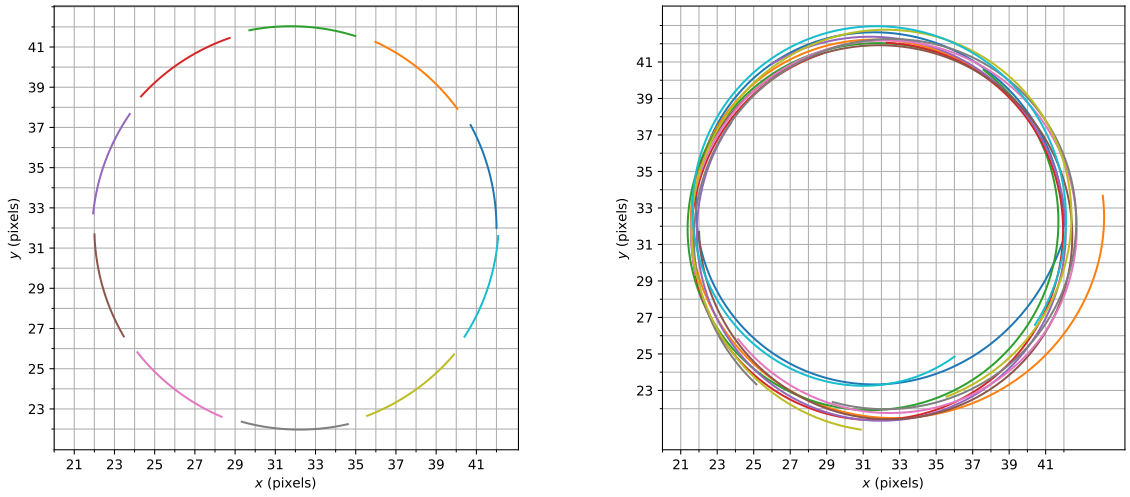
4.3. Ring system

A third system for which an analytical solution in deep MOND is known, is a ring system. This system consists of a central mass m_0 with a ring around it. This ring will be approximated in the code by N point particles at constant radius r from the central mass, with constant distance between the ring particles and with mass m. To have a good approximation of an actual ring, a large N should be used. In chapter (2), an equation was derived for the forces on the bodies in this ring system:

$$\boldsymbol{F} = -\frac{2\sqrt{Ga_0}}{3Nr}\left(M^{3/2} - m_0^{3/2} - Nm^{3/2}\right)\hat{r}. \tag{4.3}$$

In the previous equation $M = m_0 + Nm$. This force was then used to calculate the initial velocities needed for circular orbits, by equating it to the centripetal force.

In figure (4.8) the trajectories of some of the bodies of a simulation with $N = 100$ bodies are shown. In these simulations a time step of $dt = T/576 = 307 \cdot 10^6$ s was used, with T being the orbital period and the main loop was iterated over 4 times each time step. Furthermore in this simulation $G = 6.6674 \cdot 10^{-11}, a_0 = 1.2 \cdot 10^{-10}$ were used. The trajectories of the bodies show that after enough time steps the ring starts to shift away from the center. This is most likely due to the chaotic nature of the ring system, which means that small errors in the calculated accelerations will eventually lead to large errors in the positions and velocities of the bodies. The fact that initially the bodies in the ring do start out in a circular orbit does provide evidence for the correctness of equation (4.3).



(a) Trajectories after 50 time steps

(b) Trajectories after 576 timesteps

Figure 4.8: Position graphs of 10 bodies from a simulation of a ring system with $N = 100$ bodies in the ring. Only 10 out of 100 bodies are shown as otherwise the trajectories will overlap too much to show the positions of the individual bodies. The trajectories after 50 time steps and after 576 time steps are shown. The simulation was done using the particle mesh code together with Leapfrog. The graphs show that initially the bodies start in a circular orbit around the central mass, but eventually spiral away from the initial center.

4.4. Isothermal sphere

Another system for which an analytical solution of the deep MOND potential can be derived is a isothermal sphere. This isothermal sphere is a system of N particles, with N being in the thermodynamic limit, with equal masses m in hydrostatic equilibrium. P.M. Visser derived, using the hydrostatic equillibrium condition and the ideal gas law, that for this system the following explicit expressions:

$$\phi(r) = \frac{2\sqrt{GMa_0}}{3} \log\left(1 + \frac{r^{3/2}}{b^{3/2}}\right), \tag{4.4}$$

$$\boldsymbol{g}_{\mathsf{M}}(\boldsymbol{r}) = -\boldsymbol{r}\sqrt{\frac{GMa_0}{b^3 r}}\left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-1}, \tag{4.5}$$

$$M(r) = \frac{Mr^3}{b^3}\left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-2}, \tag{4.6}$$

$$\rho(r) = \frac{3M}{4\pi b^3}\left(1 + \frac{r^{3/2}}{b^{3/2}}\right)^{-3} \tag{4.7}$$

In these equations $M = Nm$ is the total mass and b is a free parameter signifying the size of the sphere. Furthermore, using the virial equation derived by Milgrom [15] it was found in chapter 2 that:

$$\frac{\overline{v^2}}{2} = \frac{\sqrt{GMa_0}}{3} \tag{4.8}$$

Here $\overline{v^2}$ is the average of the velocities of the particles squared. Now the kinetic and the potential energies are:

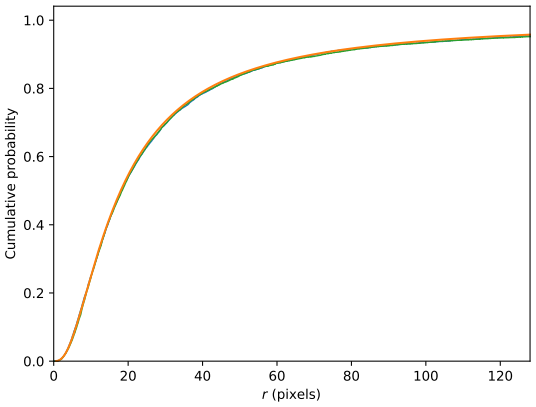
$$E_{\mathsf{kin}} = \frac{M\overline{v^2}}{2} = \frac{M\sqrt{GMa_0}}{3}, \quad E_{\mathsf{pot}} = 4\pi \int_0^\infty \phi(r)\rho(r)r^2 \mathsf{d}r = \frac{3M\overline{v^2}}{2} = M\sqrt{GMa_0}$$

To simulate this isothermal sphere one has to first create the initial state. The positions of the masses should be radially distributed via the cumulative distribution function (4.6), whereas the velocities should follow the Maxwell-Boltzmann distribution, with the average of the squared velocities $\overline{v^2} = \frac{2\sqrt{GMa_0}}{3}$.

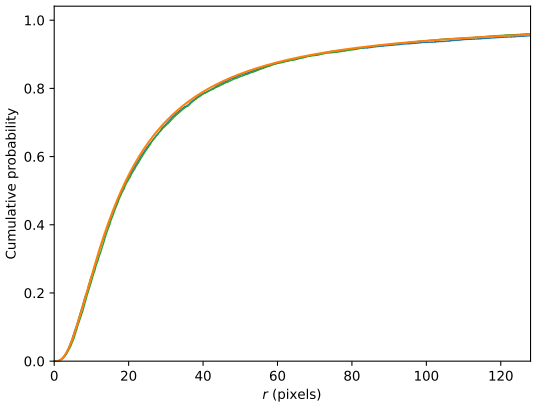
Using the particle mesh code, the time evolution of this system was tested. The simulation using the particle mesh code was compared with a simulation using the analytical solution for the force (4.5) together with Leapfrog integration. The simulation parameters were chosen as follows:

- Grid size: $256 \times 256 \times 256$ pixels, $b = 10$ pixels
- Simulation time: $T = 90$ days, time steps: 432
- Main loop iterations per timestep: one

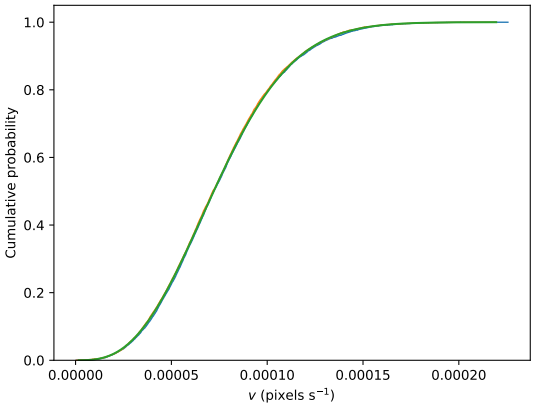
The energy graphs of this simulation are shown in figure (4.12), whereas the radial distributions of the mass and speed at the start and finish of the simulation are shown in figure (4.9). Both of these graphs show that the simulation using the particle mesh code agrees extremely well with the simulation using the analytical formula for the force. This can be explained using figure (4.11). In these figures the projected potential is shown, which is almost completely radially symmetric. In this case though, $\boldsymbol{H}_d \approx 0$, so the initial guess for $\boldsymbol{H}_d$ is already very good and the main loop only needs one iteration to converge to an accurate solution. The radial symmetry of the initial and final projected mass densities is shown in figure (4.10). One might ask how the potential can be almost completely symmetric, even though the symmetry of projected densities is quite rough. This can however, be explained by the usage of a Gaussian smoothing function to transform the mass density of the particles in space to a mass density on the grid, which will smooth the density a lot. In figure (4.13) the angular momentum is shown. One can easily see that this quantity is not conversed in the particle mesh code, which can be explained by the usage of a cuboid grid which breaks the spherical symmetry. As there is no spherical symmetry the angular momentum is not conserved.



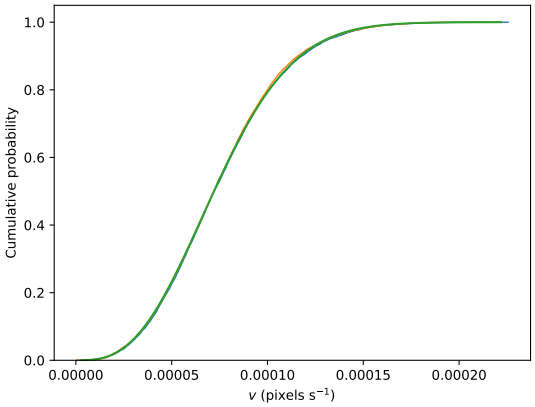
(a) Cumulative mass distribution for the simulation with the analytical force



(b) Cumulative mass distribution for the particle mesh simulation

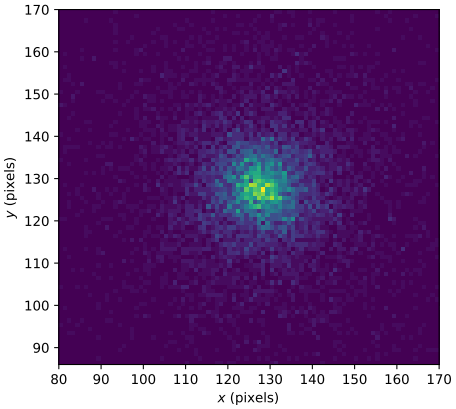


(c) Cumulative speed distribution for the simulation with the analytical force

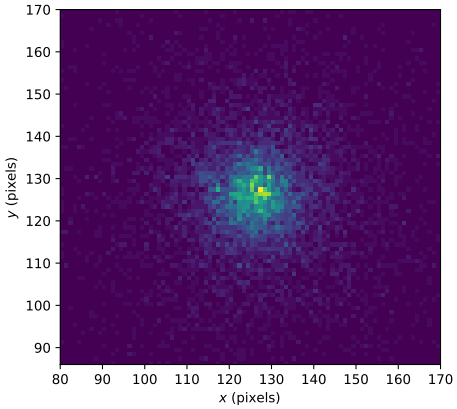


(d) Cumulative speed distribution for the particle mesh simulation

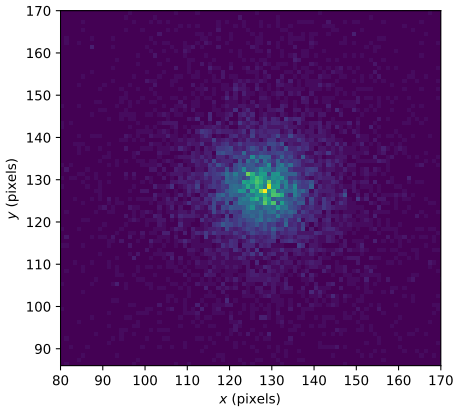
Figure 4.9: Distributions of mass and speed of the simulations of the isothermal sphere with $N = 10000$ particles. The simulation with the analytical force (4.5) are shown on the left, whereas the simulation using the particle mesh code is shown on the right. The initial distribution is shown in blue, whereas the distribution at the end of the simulation is shown in green. The distribution from which the initial positions or velocities are sampled is shown in orange. Note that due to the high number of particles all of the graphs almost completely overlap and therefore only one line is shown.



(a) Projected density at time step 0.

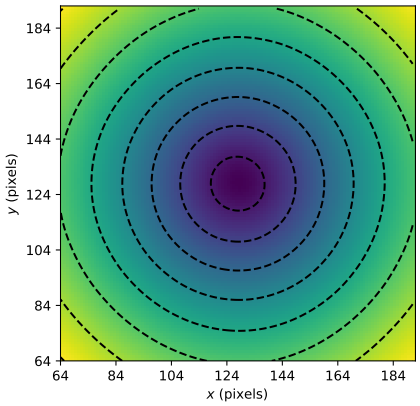


(b) Projected density at the end of the simulation using the particle mesh method.

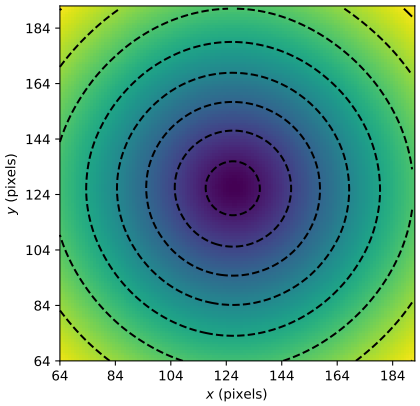


(c) Projected density at the end of the simulation using the analytical force.

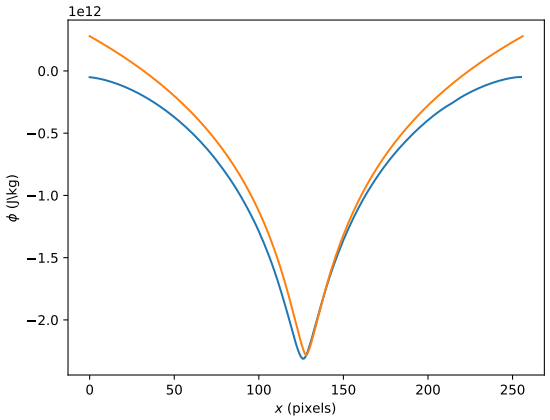
Figure 4.10: Projected densities of the isothermal sphere. The projected density at the start of the simulations is shown, together with the projected density at the end of the simulation using the particle mesh code and together with the projected density at the end of the simulation using the analytical formula for the force (4.5). The masses were distributed randomly via distribution function (4.7), with (128,128,128) being the origin. Note that the center of mass is slightly offset from the origin, due to the randomness of the initial distribution. At the end of the simulation the center of mass is even more offset, due to the total momentum at the start of the simulation being nonzero.



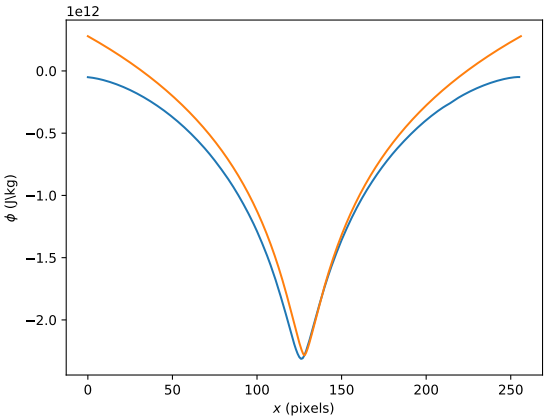
(a) Projected potential at time step 0. The black dashed lines are the contour lines.



(b) Projected potential at the last time step. The black dashed lines are the contour lines.



(c) Cross section of the potential at the time step 0. The potential as calculated by the particle mesh method is shown in blue, whereas the analytical formula for the potential of an isothermal sphere (4.4) is shown in orange.



(d) Cross section of the potential at the last time step. The potential as calculated by the particle mesh method is shown in blue, whereas the analytical formula for the potential of an isothermal sphere (4.4) is shown in orange.

Figure 4.11: Projected potential and cross-section of the potential of the isothermal sphere as calculated by the particle mesh method. Both are shown at time step 0 as well as at the end of the simulation. The isothermal sphere was simulated in deep MOND using an iterative particle mesh method and Leapfrog integration was used. The difference in offset between the cross-section of the potential and the analytical solution is caused by the center of mass not being exactly at the origin, which is at (128,128,128), whereas the analytical solution does assume that the center of mass is at the origin.

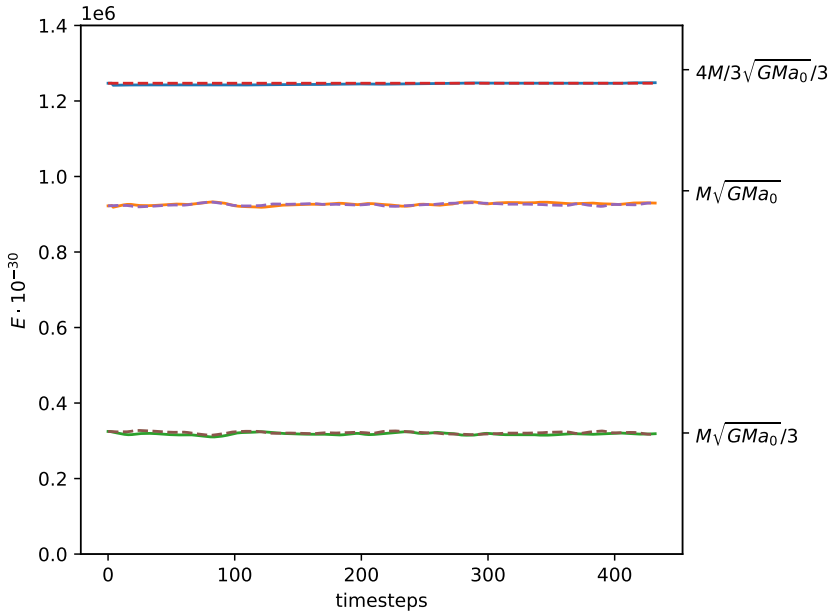


Figure 4.12: Energy graph of the simulation of the isothermal sphere with $N = 10000$ particles. The total, potential and kinetic energy of the simulation using the analytical formula for the force (4.5) are shown in red, purple and brown respectively, whereas the total, potential and kinetic energy of the simulation using the particle mesh code are shown using blue, orange and green respectively. Both simulations were done using Leapfrog integration. The graph shows that for both simulations the total energy is almost completely conserved, and that both simulations give largely the same results for the energy. Furthermore, it is shown that the kinetic energy matches well with the theoretical value for the kinetic energy, whereas the potential energy is slightly lower than the theoretical value.

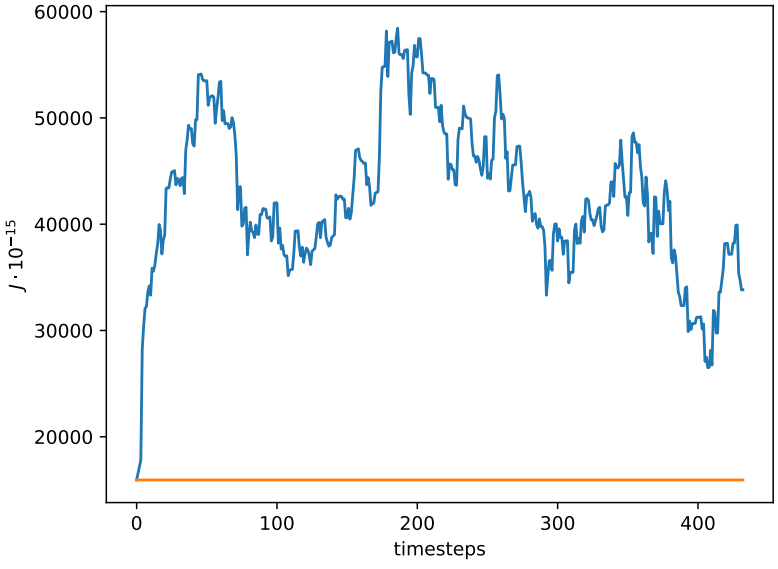


Figure 4.13: Angular momentum graph of the simulation of the isothermal sphere with $N = 10000$ particles. Only the magnitude of the angular momentum is shown. The angular momentum of the simulation using the analytical formula for the force (4.5) is shown in orange, whereas the angular momentum of the simulation using the particle mesh code is shown in blue. As expected, when using the analytical formula for the force, the angular momentum is exactly conserved. When using the particle mesh code however, the angular momentum is not conserved at all. The reason for this is that the simulation using the particle mesh code uses a cuboid grid, which breaks the spherical symmetry and which leads to angular momentum not being conserved.

4.5. Computation time

Aside from testing the accuracy of the model, testing the computation time is also quite important, as this of course largely impacts how usable the method is. This was done by testing how long it takes to compute the accelerations on the bodies in a ring system, using a grid of $128 \times 128 \times 128$ pixels and doing four main loop iterations, the results of which are shown in figure (4.14). Note that these computations were done on my own laptop and that by using a supercomputer the computation times can definitely be shortened. The scalings of the computation times, however, will stay the same on every device on which the simulations are done. From the graph it can be seen that, as expected, the time it takes to execute the FFT's is independent of the number of particles in the system. Furthermore, the time it takes to distribute the masses of the particles to the grid and to distribute the accelerations back to the particles scales linearly with the number of particles in the system. With these two scalings combined, it can be concluded that up until the computation time it takes to distribute the masses to the grid and the accelerations back to the particles is roughly equal to the computation time of the FFT's, the number of particles is largely irrelevant for the computation time, whereas afterwards the number of particles will largely determine the computation time.

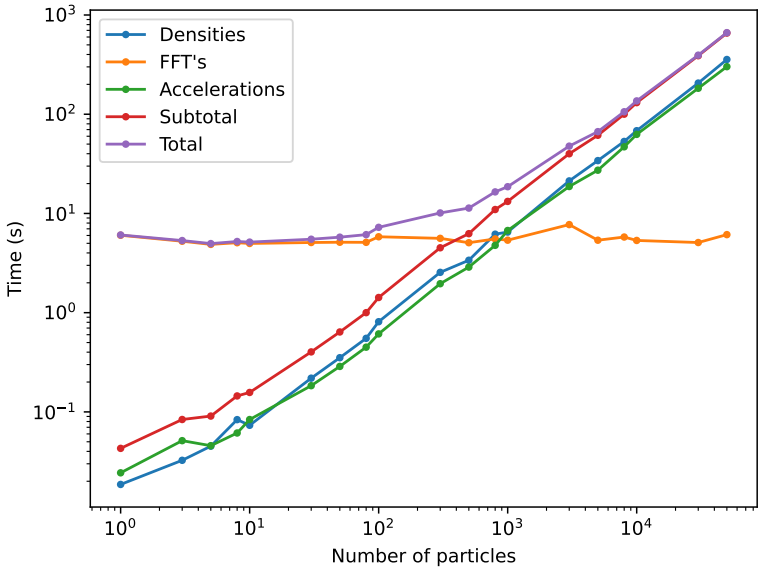


Figure 4.14: Computation time of MOND accelerations for systems consisting of a ring of point particles orbiting a central point mass. The particle mesh method can be divided into three stages, which each take significant computation time. First the density on the grid has to be found, after which the accelerations on the grid points can be found using lots of FFT's and lastly the accelerations on the bodies can be found from the accelerations on the grid points. Each of these stages is shown in this graph, with a subtotal, which is the sum of the computation times of finding the density and computing the accelerations on the particles, and a total, which is the total computation time.

5

Research recommendations

5.1. Speeding up the algorithm

In future researches one might need to speed up the algorithm to be able to effectively use it. Depending on the number of particles in the system one might want to focus on the computation time of the FFT's or on the computation time of translating the particles to densities on a grid and translating the acceleration field on a grid back to the particles. In the first case one has the following options:

- Try to reduce the number of iterations needed to find the MOND accelerations
- Implement the FFT more efficiently

Note that the current implementation of the FFT uses the FFTW library, which is one of the fastest free implementations of the FFT, so improving this might be difficult, but it might be worth looking into how this library can be used best for this application. In case one wants to focus on the second category, some options are:

- Using balls to smooth the masses of a particle over, instead of a box
- Parallelising this part of the code
- Implementing this part of the code in a faster programming language

The first point was actually already implemented in the code and was confirmed to improve computation times quite a bit, the impact on the accuracy of the code was, however, not yet tested. For the second point, this part of the code should be highly parallelisable, as the list of particles can be split up in equal parts, then the density on the grid of each of these parts can be computed, after which they can be added back together. Something similar can be done to find the accelerations on the particles from the grid. The third option is valid as the code is currently implemented in Python, which is not a huge problem for the most part of the code as the FFT's are executed using C and hence do not face the slowness of Python, and the other parts of the code do not take significant time. For this second category of computation though, using a faster programming language than Python is expected to speed up the code significantly.

5.2. Improving the accuracy

To improve the accuracy of the algorithm one would first need to find the causes of the inaccuracy of the algorithm. Two of these causes are already known, it is however not known how much inaccuracy they cause. First of all is the self-interaction. This self-interaction is caused by the need to represent a particle at a random point in space by masses on a finite grid. This random point in space will in general not be on a grid vertex, and therefore the mass of the particle is distributed over multiple grid vertices to imitate a mass between those vertices. This is done using a smoothing function. Due to the nonlinearity of the Poisson equation for MONDian gravity however, the gravity between the grid vertices over which the mass is smoothed will result in a nonzero acceleration on the particle. Using different smoothing function will result in different self-interactions, with Gaussian smoothing typically giving the least self-interaction. To combat this self-interaction one could look more into smoothing

functions and find one that works well specifically for MOND, one could try adding mass in such a way that the gravity of this extra mass will cancel out the self-interaction, or one could look further into the ways self-interactions are lowered in other N-body code algorithms such as the a tree algorithm. The second cause of inaccuracy is the approximation of discrete Fourier transform identities by continuous Fourier transform identities. All of the Fourier transform identities derived and used in this research hold for continuous Fourier transforms, and do generally not hold for discrete Fourier transforms, while only discrete Fourier transforms were used. The identities for the continuous Fourier transforms are good approximations, however one might be able to improve the accuracy of the algorithm by using the corresponding identities for discrete Fourier transforms.

5.3. Model extensions

To be able to simulate real systems in the galaxy, one might need to add parts to the code, which will handle physical phenomena outside of gravity. Examples of such phenomena are:

- Introduce comoving coordinates to model the expansion of the universe
- Add hydrodynamics to simulate gasses
- Add a method to handle close encounters

Using comoving coordinates instead of stationary coordinates will become quite important when one wants to model systems at the cosmological scale, where the effects of the expansion of the universe will become significant. Furthermore, in case one wants to for example simulate galaxy clusters, which typically contain more mass in their intergalactic gas than in the galaxies inside of them, one might want to hydrodynamics to simulate this gas.

5.4. Possible applications of algorithm

Instead of improving the algorithm, one might be interested in how this algorithm can be used to simulate parts of the universe and research the validity of MOND by comparing observational data with a simulation. For this kind of research a couple of things need to be taken into account. One needs to find a system of which observational data is available, the gravity in the system is in the MOND regime and the number of bodies in the system should be sufficiently low to have fast computation times. Note that when considering if the gravity in the system is weak enough to be in the MOND regime, one needs to take the external field effect into account, which means that one needs to take the forces that act on the system as a whole into account as well. An example of this would be that when one would want to simulate the solar system using MOND[4], one would need to take the gravitational force to the center of the milky way into account, around which the solar system orbits.

Examples of such systems are:

- The solar system[4]
- Globular clusters[2]
- Tidal dwarf galaxies[16]
- Galaxy clusters[22]
- Wide binary stars[19]

In the case one wants to simulate the solar system, one could look into the perihelion precession of the outer planets in the solar system, and compare simulations using both MONDian gravity and Newtonian gravity to observational data, and try to verify the results of L. Blanchet and J. Novak in[4]. For the globular clusters it would be interesting to determine velocity dispersions inside of these clusters, as they are expected to deviate highly from the velocity dispersions in the case of Newtonian dynamics[2].Tidal dwarf galaxies form an interesting alternative as these are not expected to contain dark matter. Hence one could simulate these and determine the rotation curves, which in the case that they are flat could give great evidence for MOND[16]. In the case of galaxy clusters one could investigate if using MOND will remove the need for dark matter to explain the stability of these clusters. One could also try to investigate the cooling of the intracluster gas, as it is unknown why this gas cools down less than expected. Lastly as mentioned in [19], wide binary stars form a good test for MOND as there is predicted to be little to no dark matter in the system, which means one can compare the situation with Newtonian gravity with the MOND situation, in which one also needs to take the gravitational pull

from the center of the galaxy into account as an external field. Note that some of these systems might already have been researched, as this model might be more accurate however, it could be interesting to revisit some of these topics using this particle mesh method.

6

Conclusion

In this thesis an iterative particle mesh N-body code was developed to simulate systems in MOND. Afterwards this code was tested in deep MOND and numerous results were found.

First of all, it was concluded that to convert the densities in real space to the mesh, the cloud in cell method which is typically used in Newtonian particle mesh codes, is not sufficient anymore as it introduces nonphysical oscillations in the acceleration field. Instead, it was found that Gaussian smoothing should be used.

Secondly, by simulating only one body, it was found that particles in this code interact with themselves, resulting in accelerations on the order of $10^{-3}\frac{\sqrt{MGa_0}}{\Delta x}$ pixels s^{-2} when Gaussian smoothing with standard deviation $\sigma = 1$ is used. This self-interaction was expected, as it is a consequence of the nonlinearity of the MOND Poisson equation.

Next, the two-body problem was simulated. The force between two bodies was calculated for different distances between them and it was found that the magnitude of the relative numerical error in the force as calculated by the particle mesh code between two bodies in deep MOND was comparable to the magnitude of the same error in Newtonian dynamics.

Furthermore, it was shown in all test cases that the method generally respects the conservation of energy, which can be attributed to the usage of Leapfrog integration, whereas angular momentum is generally not conserved very well, which is most likely caused by the usage of a cuboid grid which breaks the spherical symmetry.

Fifthly, it was shown that predictions of the analytical formula for the force between two bodies and for the potential of an isothermal sphere can be reproduced using the particle mesh code, which gives great support for the correctness of these formulae and the correctness of the particle mesh code.

It was also shown that one typically needs to iterate one to four times in the main loop, with once being sufficient for nearly spherically symmetric systems.

Another conclusion was that just like in Newtonian particle mesh code, close encounters are not handled well by this altered particle mesh code.

Lastly, it was shown that the computation time of the forces depends on the computation time of the FFT's, which only depend on the size of the grid, and on the computational time needed to distribute the point masses in real space to the mesh and to distribute the acceleration from the mesh to the point masses, which scales linearly with the number of particles in the system. These two computation times will be comparable for a grid of $128 \times 128 \times 128$ pixels and 500-5000 particles, depending on the computer on which the code is run.

Some research recommendations include speeding up the algorithm, for example by completely parallelising the code and or writing it in C, improving the accuracy, by trying to reduce self-interactions and extending the model, by introducing a method to handle close encounters. Furthermore, possible applications of this algorithm include simulating the solar system, wide binary stars, tidal dwarf galaxies and galaxy clusters, and comparing the simulations to observational data.

References

[1] P. A. R. Ade et al. “iPlanck/i2015 results”. In: *Astronomy &amp Astrophysics* 594 (Sept. 2016), A13. DOI: 10.1051/0004-6361/201525830.

[2] H. Baumgardt, E. K. Grebel, and P. Kroupa. “Using distant globular clusters as a test for gravitational theories”. In: *Monthly Notices of the Royal Astronomical Society: Letters* 359.1 (May 2005), pp. L1–L4. ISSN: 1745-3925. DOI: 10.1111/j.1745-3933.2005.00021.x. eprint: https://academic.oup.com/mnrasl/article-pdf/359/1/L1/6402197/359-1-L1.pdf.

[3] Gianfranco Bertone, Dan Hooper, and Joseph Silk. “Particle dark matter: evidence, candidates and constraints”. In: *Physics Reports* 405.5-6 (Jan. 2005), pp. 279–390. DOI: 10.1016/j.physrep.2004.08.031.

[4] Luc Blanchet and Jerome Novak. *Testing MOND in the Solar System*. 2011. arXiv: 1105.5815 [astro-ph.CO].

[5] James S. Bullock. *Notes on the Missing Satellites Problem*. 2010. DOI: 10.48550/ARXIV.1009.4505.

[6] G. N. Candlish, R. Smith, and M. Fellhauer. “RAyMOND: an N-body and hydrodynamics code for MOND”. In: *Monthly Notices of the Royal Astronomical Society* 446.1 (Nov. 2014), pp. 1060–1070. ISSN: 0035-8711. DOI: 10.1093/mnras/stu2158. eprint: https://academic.oup.com/mnras/article-pdf/446/1/1060/4159122/stu2158.pdf. URL: https://doi.org/10.1093/mnras/stu2158.

[7] D. Welsh G. Grimmet. *Probability, An introduction*. Oxford University Press, 2014.

[8] Langhua Hu and Guo-Wei Wei. “Nonlinear Poisson Equation for Heterogeneous Media”. In: *Biophysical Journal* 103.4 (2012), pp. 758–766. ISSN: 0006-3495. DOI: https://doi.org/10.1016/j.bpj.2012.07.006.

[9] P. Londrillo and C. Nipoti. “N-MODY: a code for collisionless N-body simulations in modified Newtonian dynamics.” In: *Memorie della Societa Astronomica Italiana Supplementi* 13 (Jan. 2009), p. 89. DOI: 10.48550/arXiv.0803.4456. arXiv: 0803.4456 [astro-ph].

[10] Stacy S. McGaugh. “The Baryonic Tully-Fisher Relation of Gas-rich Galaxies as a Test of ΛCDM and MOND”. In: *The Astronomical Journal* 143.2, 40 (Feb. 2012), p. 40. DOI: 10.1088/0004-6256/143/2/40. arXiv: 1107.2934 [astro-ph.CO].

[11] Stacy S. McGaugh, Michael K. Barker, and W. J. G. de Blok. “A Limit on the Cosmological Mass Density and Power Spectrum from the Rotation Curves of Low Surface Brightness Galaxies”. In: *The Astrophysical Journal* 584.2 (Feb. 2003), pp. 566–576. DOI: 10.1086/345806.

[12] M. Milgrom. “A modification of the Newtonian dynamics as a possible alternative to the hidden mass hypothesis.” In: *The Astrophysical Journal* 270 (July 1983), pp. 365–370. DOI: 10.1086/161130.

[13] M. Milgrom. “Modified dynamics predictions agree with observations of the Hiota kinematics in faint dwarf galaxies contrary to the conclusions of Lo, Sargent, and Young”. In: *The Astrophysical Journal* 429 (1994).

[14] Mordehai Milgrom. “Forces in nonlinear media”. In: *Journal of Physics A: Mathematical and General* 35.6 (Feb. 2002), pp. 1437–1453. DOI: 10.1088/0305-4470/35/6/308.

[15] Mordehai Milgrom. “General virial theorem for modified-gravity MOND”. In: *Phys. Rev. D* 89 (2 Jan. 2014), p. 024016. DOI: 10.1103/PhysRevD.89.024016.

[16] Mordehai Milgrom. “MOND and the Mass Discrepancies in Tidal Dwarf Galaxies”. In: 667.1 (Sept. 2007), pp. L45–L48. DOI: 10.1086/522049. arXiv: 0706.0875 [astro-ph].

[17] Mordehai Milgrom. “Nonlinear conformally invariant generalization of the Poisson equation to D>2 dimensions”. In: *Physical Review E* 56.1 (July 1997), pp. 1148–1159. DOI: 10.1103/physreve.56.1148.

[18] Massimo Persic, Paolo Salucci, and Fulvio Stel. “The universal rotation curve of spiral galaxies — I. The dark matter connection”. In: *Monthly Notices of the Royal Astronomical Society* 281.1 (July 1996), pp. 27–47. ISSN: 0035-8711. DOI: 10.1093/mnras/278.1.27. eprint: https://academic.oup.com/mnras/article-pdf/281/1/27/30383982/281-1-27.pdf. URL: https://doi.org/10.1093/mnras/278.1.27.

[19] Charalambos Pittordis and Will Sutherland. “Wide Binaries from GAIA EDR3: preference for GR over MOND?” In: *The Open Journal of Astrophysics* 6 (Feb. 2023). DOI: 10.21105/astro.2205.02846.

[20] J.W Eastwood R.W Hockney. *Computer Simulation Using Particles*. CRC Press, 1988.

[21] V. C. Rubin, Jr. Ford W. K., and N. Thonnard. “Extended rotation curves of high-luminosity spiral galaxies. IV. Systematic dynamical properties, Sa -> Sc.” In: 225 (Nov. 1978), pp. L107–L111. DOI: 10.1086/182804.

[22] R. H. Sanders. “Clusters of galaxies with modified Newtonian dynamics”. In: 342.3 (July 2003), pp. 901–908. DOI: 10.1046/j.1365-8711.2003.06596.x. arXiv: astro-ph/0212293 [astro-ph].

[23] S. Torres-Flores et al. “GHASP: an H kinematic survey of spiral and irregular galaxies - IX. The near-infrared, stellar and baryonic Tully-Fisher relations72 ”. In: *Monthly Notices of the Royal Astronomical Society* 416.3 (Aug. 2011), pp. 1936–1948. DOI: 10.1111/j.1365-2966.2011.19169.x.

[24] R. B. Tully and J. R. Fisher. “A new method of determining distances to galaxies.” In: *Astronomy and Astrophysics* 54 (Feb. 1977), pp. 661–673.

A

Appendix

A.1. Code

The code can be found in the following GitHub repository: `https://github.com/Joost987/MONDPMesh`

A.2. Gaussian weight functions

Theorem A.2.1 (W_{2n} is asymptotically equal to a Gaussian)**.**

$$W_{2n} \to \sqrt{\frac{6}{2\pi n}} e^{-\frac{1}{2}(\frac{x}{\sqrt{n/6}})^2} \qquad \text{as } n \to \infty$$

Proof. Let it be noted that $\bigwedge$ is a probability density function. Suppose $Y_1, Y_2, \ldots, Y_n$ are independent random variables drawn from $\bigwedge$. Then their sum, $\sum Y_n = \sum_{k=1}^{n} Y_k$, is a random variable drawn from [7]:

$$\bigwedge * \bigwedge * \ldots * \bigwedge = W_{2n}$$

However, by the central limit theorem, the distribution of $\sum Y_n$ converges to a normal distribution. In particular, as $\mathbb{E}[Y_k] = 0$ and Var$[Y_k] = 1/6$:

$$W_{2n} \to \sqrt{\frac{6}{2\pi n}} e^{-\frac{1}{2}\frac{x^2}{n/6}} \qquad \text{as } n \to \infty$$

In the above, $\to$ stands for convergence in distribution. Rewriting this using $\sigma^2 = n/6$,

$$W_{2n} \to \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \qquad \text{as } n \to \infty$$

Hence W_{2n} is asymptotically equal to $\sqrt{\frac{6}{2\pi n}} e^{-\frac{1}{2}\frac{x^2}{n/6}}$ □