Delft University of Technology

MSc Thesis

Double MSc: Applied Physics - Computer Engineering

Faculty of Applied Sciences

Faculty of Electrical Engineering, Mathematics and Computer Science

Galaxy Clusters in MOND: from Aether Theories to FEM in FEniCS

Author
Vieri Mattei (4750942)

Daily Supervisor
Dr Paul M Visser

Chair of the Committee
Prof Catherine Pappas

Committee Member - Computer Engineering
Dr Zaid Al-Ars

Committee Member - Applied Physics
Dr Stephan Eijt

10/12/2020

TUDelft

Delft
University of
Technology

Abstract

Modified Newtonian Dynamics (MOND) can account for a variety of phenomena on galactic scales without the need for dark matter, but it cannot fully explain the mass contained in galaxy clusters. We explore two possible solutions to this problem: relativistic extensions of MOND, and FEM simulations of the apparent matter distribution in clusters, utilising the non-linear AQUAL formulation to analyse non spherically symmetric systems.

We consider Covariant Emergent Gravity, and we show that the theory is inconsistent with the original formulation of Emergent Gravity. Moreover, we show that either the theory is incompatible with observations, or it presents grave theoretical difficulties. We then suggest that a covariant formulation of EG can be obtained through a Generalised Einstein Aether (GEA) theory, which is capable of retrieving the MOND PDE, and is, at the same time, consistent with observational constraints.

Regarding the FEM simulations of the apparent matter distribution in galaxy clusters, we construct a sample of 15 clusters from the catalogs of Reiprich and Abell for the baryonic mass distribution present in galaxy clusters. We choose FEM for its ability to treat the combination of continuous and discrete mass distributions without the need for smoothing. This is necessary, as in MOND we cannot apply the principle of superposition or the weak lensing formalism. We then utilise the FeniCS software package to study the properties of these clusters. We simulate each cluster with elements up to degree 3, thanks to a speedup by a factor of ~ 100 obtained by the use of local mesh refinement for the serial case. In addition, we run the code in parallel on a single-threaded 8-core CPU, achieving near optimal weak scaling for the regime of interest.

For the mass distribution in the galaxy clusters, we analyse the distribution of baryons, Phantom Dark Matter (PDM) and apparent mass both close to the core and around each galaxy. We find that the PDM tends to clump around the galaxies, regardless of the gas to galaxy mass ratio. Moreover, we show that both the apparent mass and the PDM can exhibit negative masses as predicted by Milgrom. Our observations on the density of PDM around the galaxies match recent observations of small scale weak lensing. In addition, our results for negative mass distributions provide an opportunity to test a prediction of MOND that can never be replicated in the dark-matter paradigm, and shed light on the properties of non-spherically symmetric mass distributions, that have, up to now, not been studied in the literature for the fully non-linear case.

Acknowledgements

The process of writing this thesis has been extremely confusing at times, often exciting and fun, and occasionally incredibly frustrating. After all of it, the only thing that I am sure about is that it would have not been the same had it not been for all the help I received since I enrolled at TU Delft over three years ago. I want to thank Arno Haket for helping me find the right courses for the bridging program, and plan the structure of my double MSc. Another special thanks goes to Kees Lemmens, which opened my eyes to the world of FEniCS. Without his advice I would still be trying to hopelessly solve the MOND PDE on Matlab. I want to thank all the members of my thesis committe: Dr Al-Ars, for accepting to be on the committee on such short notice, and for the advice on official matters which many times stresses me out more than exams; Prof Catherine Pappas, for accepting to being the supervisor for my internship almost two years ago, and the chair of the committee for my thesis; Dr Eijt, for pointing me in the direction of the project, and being extremely helpful with a lot of great feedback during the thesis; my daily supervisor Dr Visser, for always being available, open minded, and providing insight on wildly different topics in physics, and also for being supportive and passionate throughout the whole project, and helping me learn a really surprising amount of stuff in what feels like a really short period of time. Of course, I have to thank all my jeebers, whether they are in Delft, Italy, Scotland or scattered around the world, for always being there to have a laugh, talk, or just spend time together. Finally, I want to thank my parents and my sister for always being there for me. Of course none of this would be possible without their support, and I hope to reunite with them soon after this crazy year. I am sure my grandparents are also watching from somewhere and being glad I am finally almost done with studying. Or am I...

Contents

Part I

Applied Physics

Chapter 1

Gravity

This chapter provides a general background for the work presented in the following chapters. Three main frameworks are described: for the nonrelativistic case, Newtonian gravity and MOND. Emphasis is placed on their predictions for galactic rotation curves. For the relativistic case, General Relativity is described. For entropic gravity, the ideas of Emergent Gravity are introduced, focusing mainly on the approach by Verlinde. After presenting the Lagrangian formalism, the chapter will end by showing how the first three theories can be derived using a Lagrangian approach.

1.1. Introduction

In an article published in 1981, one of the most brilliant minds of modern physics made an optimistic prediction: by the end of the twentieth century, a **theory of everything** would be discovered. He speculated that theoretical physics would fulfill its ultimate goal of describing the universe as a whole. It would do so by means of a single framework capable of unifying all forces and providing all boundary conditions needed to coherently describe the evolution of the cosmos through time and space. The author was Stephen Hawking, and the best candidate for a theory of everything was at the time **supergravity** [1].

Almost forty years later, no such theory has been found. Nonetheless, the microscopic world is described with astonishing precision by **QFT** (Quantum Field Theory), unifying the principles of **Quantum Mechanics** with those of **Special Relativity**. Meanwhile, the gravitational dynamics of the planets are governed by the laws of **General Relativity**. However, the rules describing the two scales appear fundamentally incompatible, and **Quantum Gravity** continues to elude theoretical physicists.

Many unanswered questions remain. The **Standard Model** of particle physics, unifying all forces other than gravity, cannot precisely predict the mass of the neutrino, and any evidence corroborating expansions of the model such as **Supersymmetry** has yet to be detected. Furthermore, on the scales of galaxies, galaxy clusters and beyond, many observations fail to match the predictions of General Relativity:

1. Stars in the outskirts of galaxies move at speeds that far exceed those predicted by the total galactic mass inferred by observations [2];

2. Galaxies in galaxy clusters move too fast to remain gravitationally bound to each other [3];

3. The accelerated expansion rate of the universe does not match that stemming from the vacuum energy predicted by QFT [4].

To cope with the disagreement between theory and observation, two concepts have been introduced in the current **Standard Model of Cosmology** (**SMC**): **dark matter** and **dark energy**. As a result, the model is named **Λ-Cold Dark Matter** (**Λ-CDM**), where Λ indicates dark energy.

Although successful in describing a variety of phenomena, ranging from galactic rotation curves to large scale structure formation, the Λ-CDM model is by no means the only pathway towards explaining the gravitational anomalies pointed out. Its driving principle is the postulate that General Relativity is valid at all macroscopic scales, which means that the observations can only be explained by introducing additional source terms to the Einstein Equations, such as dark matter and dark energy. On the opposite end of the spectrum, one could hypothesise that the laws of gravitation that hold on the scale of the Solar System are the limit of a more general theory of gravity, which can account for all gravitational behaviour based solely on the observable mass present in the universe (the so called baryons, such as protons and neutrons).

Theories which follow the latter approach belong to the family of **Modified Gravity**, and will be the focus of this work. However, before going into more detail about these, an overview will now be given of the classical theories of gravity that will be discussed in the following chapters.

1.2. Newtonian Gravity

1.2.1. Basic Concepts

The same force that accelerates an apple down from its tree is responsible for the motion of the planets around a star. This was the point of view of Isaac Newton, who considered gravity as a universal attractive force. The classification of gravity as a force has been made obsolete by General Relativity. Nevertheless, Newton's theory was revolutionary, and is still used nowadays as an excellent approximation of gravitational interaction in systems where both the masses and relative velocities of the bodies involved are below a certain threshold.[1]

The magnitude of the well known attractive force between two massive objects is given by the inverse square law:

$$F = G\frac{m_1 m_2}{r^2}, \tag{1.2.1}$$

[1] The weak-gravity, low-velocity limit is defined for velocities $v \ll c$, and masses satisfying $M/r \ll M/r_s \implies r/r_s \gg 1$, where $r_s = 2GM/c^2$ is the Schwarzschild radius.

where F represents the magnitude of the force, m_1 and m_2 are the masses of the two bodies, r is their distance, and G is Newton's gravitational constant $G \approx 6.674 \cdot 10^{-11} \frac{m^3}{kgs^2}$. The force is attractive and directed along the vector connecting the positions of the two masses.

Another fundamental concept of Newtonian mechanics is Newton's second law, stating that a change in momentum is related to an equivalent force:

$$F = \frac{\mathrm{d}p}{\mathrm{d}t} = m_1 a, \tag{1.2.2}$$

where p and a are respectively the magnitudes of the momentum and acceleration. It is then easy to see that one can obtain the acceleration due to the gravitational force by equating (1.2.1) and (1.2.2):

$$\boxed{G\frac{m_1 m_2}{r^2} = m_1 a \implies a = G\frac{m_2}{r^2}.} \tag{1.2.3}$$

One can deduce an extremely important principle from this relation:

> The acceleration a test particle undergoes due to its gravitational interaction with a massive object of mass m does not depend on the test particle's mass or other physical properties. It depends solely on the mass of the massive object m, the separation distance r, and a constant G.

A simple consequence of this statement is that all objects on Earth (as well as any other planet or star, ignoring deviations from the approximation of a perfect sphere) will feel the same gravitational acceleration. Moreover, to arrive at (1.2.3), it was assumed that the mass relating force and acceleration is proportional to the mass responsible for gravitational interaction for some fixed constant C. This apparently innocuous notion would later become one of the keystones of General Relativity, known as the equivalence principle. In General Relativity the masses are exactly equal, whereas in Newtonian gravity one has:

> Inertial mass and gravitational mass are proportional, and related by a fixed constant: $m_a = C m_g$.

1.2.2. PDE Formulation: The Newton-Poisson equation

When formally expressing the laws of physics, a powerful tool is the **Partial Differential Equation** (**PDE**). For mass distributions more complicated than two point particles, one can derive the Newtonian laws of gravitation through the use of the Poisson equation for gravity:

$$\nabla^2 \phi\left(\vec{r}\right) = 4\pi G \rho\left(\vec{r}\right), \tag{1.2.4}$$

where $\nabla^2 = \nabla \cdot \nabla$ is the Laplacian operator, $\rho(\vec{r})$ an arbitrary mass distribution varying in space and $\phi(\vec{r})$ the gravitational potential. It is easy to see that shifting the potential by an arbitrary constant α leaves (1.2.4) unchanged, as:

$$\nabla\phi = \nabla(\phi + \alpha). \tag{1.2.5}$$

To reconcile the PDE formulation with the classical formula for the gravitational attraction between two point masses, one can solve (1.2.4) for a point source of mass m in the origin as follows:

$$\nabla^2\phi(\vec{r}) = 4\pi G m \delta(\vec{r}). \tag{1.2.6}$$

Making the physically sound assumption that the potential vanishes at physical infinity, one can take a large sphere as the domain Ω and impose the boundary condition:

$$\phi|_{\partial\Omega} = 0 \tag{1.2.7}$$

It can then be recognised that in (1.2.6) ϕ is none other than the Green's function for the infinite space Poisson equation. In three dimensions the solution is known, and yields (see, for example, [5]):

$$\boxed{\phi = -\frac{Gm}{r}.} \tag{1.2.8}$$

The following observations can now be made: since gravity is a conservative force, the force on a test particle can be obtained as the gradient of the potential energy. Given that the gravitational potential ϕ has dimension of energy per unit mass, its gradient gives the force per unit mass, viz. the gravitational acceleration:

$$\vec{a} = \nabla\phi. \tag{1.2.9}$$

For a spherically symmetric potential, one has $\phi = \phi(r)$, and the above reduces to:

$$\boxed{a(r) = \frac{\partial\phi}{\partial r} = \frac{Gm}{r^2},} \tag{1.2.10}$$

where the acceleration is radial. It can hence be seen that the Poisson formulation recovers the simple case of two-body attraction.

1.3. Rotation Curves

1.3.1. Solar System

On the scale of the Solar System, Newtonian gravity has been extremely successful. The planetary motions can be calculated with high accuracy, apart from some corrections due to General Relativity such as the perihelion precession of Mercury (see, for example, [6]). An example is given by the rotation curves of the planets around the Sun, which are in excellent agreement with predictions. Rotation curves describe the velocity of a body in a gravitational field as a function of its distance from the center of the mass distribution. Assuming approximately circular orbits,[2] one can give a prediction of the velocities based on the distribution of matter[3].

In the Solar system, 99.8 % of the total mass is contained in the Sun, the location of which can then to good approximation be taken as the center of mass. It follows that it is reasonable to model the Sun as a point source, and treat the planets as test particles orbiting around it.[4] On the other hand, the ratio between the solar radius r_S and its average distance from Pluto d_{SP} (the farthest planet) is $r_S/d_{SP} \approx 10^{-4}$.

As previously mentioned, all planets will undergo an acceleration determined solely by their distance from the Sun and the solar mass $M_\odot$. Assuming perfectly circular orbits, one can then equate the gravitational acceleration with the centripetal acceleration:

$$a = v^2/r \rightarrow v^2 = ar. \tag{1.3.11}$$

The velocity as a function of radial distance is then easily obtained as:

$$\boxed{v = \sqrt{\frac{GM_\odot}{r}}}, \tag{1.3.12}$$

with $M_\odot$ the mass of the Sun.

In the Solar System, the inverse square root behaviour is observationally confirmed to great accuracy [7]. Newtonian predictions match observations for the rotation curves of the planets.

1.3.2. Galaxies

For galaxies, one could build a simple model in which the mass density is uniform within a certain radius, and drops to zero beyond it, a useful approximation of the density of stars and gas dropping to a negligible amount outside of the central core. Although inexact, such a model should reproduce the qualitative features of the data. In the case of spherical symmetry, this would be expressed as:

$$\rho = \begin{cases} \frac{M}{V} & r < r_0 \\ 0 & r > r_0 \end{cases} \xrightarrow{V(r)=\frac{4}{3}\pi r^3} \begin{cases} \frac{3M}{4\pi r^3} & r < r_0 \\ 0 & r > r_0 \end{cases} \tag{1.3.13}$$

[2] The orbits of the planets in the solar system are elliptical. However, circular orbits are very good approximations as because the orbital eccentricities are small. The planet with the largest eccentricity is Mercury with a value $e_M \approx 0.2$, whereas the Earth has $e_E \approx 0.0167$. A perfectly circular orbit has an eccentricity $e = 0$.

[3] The distribution of matter can be observed directly, or inferred indirectly. In the solar system, one has the former, but on galactic scales and beyond, the latter is the norm.

[4] The test particle treatment implies ignoring the gravitational force the planets exert on each other.

Solving the Poisson equation (1.2.4) for the given mass distribution produces the following accelerations:

$$g = \begin{cases} \frac{4}{3}G\pi r\rho & r < r_0 \\ \frac{GM}{r^2} & r > r_0, \end{cases} . \tag{1.3.14}$$

The rotational velocities immediately follow:

$$v^2 = \begin{cases} \frac{4}{3}G\pi r^2\rho & r < r_0 \\ \frac{GM}{r} & r > r_0 \end{cases} \implies v = \begin{cases} \sqrt{\frac{4}{3}G\pi\rho}r & r < r_0 \\ \sqrt{\frac{GM}{r}} & r > r_0, \end{cases} \tag{1.3.15}$$

so that two regimes can be identified: inside the mass distribution bodies have velocities that linearly increase with the distance from the origin as $v_{in} \propto r$, and outside of it the velocities fall with increasing distance as $v_{out} \propto \sqrt{\frac{1}{r}}$.

> While the linear rotation profile is in fact observed in the innermost regions of galaxies, the inverse square root decay does not match observations. Instead, the rotation speeds are seen to flatten out.

The first to measure this phenomenon with sufficient accuracy, thus giving reliable evidence of the tension with the Newtonian prediction, was Vera Rubin in 1970 [8]. Since then, increasingly precise measurements have confirmed the preliminary results. An example is given in fig. 1.3.1, taken from [9] and showing both predicted and measured rotation curves for the galaxy M33, with a clear disagreement between the two. In order for the observations to be explained, one of two routes can be taken:

1. The laws of gravitation formulated by Newton should apply to all scales, as also indicated by the slow-moving, low-gravity regime of GR. The discrepancy between the prediction and measurement of the rotational velocities is a consequence of neglecting non-radiating matter. This means that the universe is permeated by a non-baryonic form of matter which interacts exclusively with baryonic matter through its gravitational pull: this is the hypothesised **dark matter**.

2. The laws of gravitation are different on the scales of galaxies and galaxy clusters. The work described in all of the following sections is based on this assumption. Its implications will begin to be explored in the next section.

These two options can be summarised as follows:

> A correct theory of gravity must either modify the source term or the interaction. The former case gives rise to dark matter, the latter to modified gravity.

For the measurements to be explained in the $r > r_0$ domain, one could hence postulate a correction to either the mass distribution or to the gravitational acceleration, or to both.

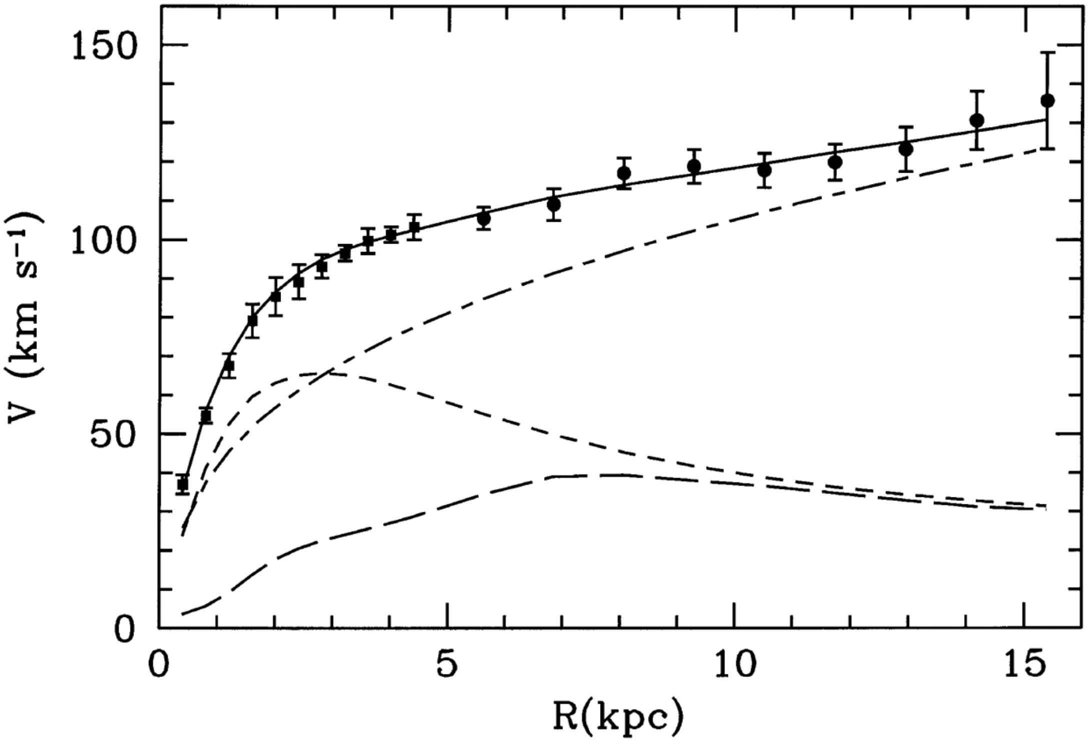

Figure 1.3.1: The rotation curve of galaxy M33 from [9]. The markers represent the measured data, with the continuous line giving the best fit. The short dashed and long dashed lines denote the contributions by star and gas mass distributions respectively, whereas the dot dashed line denotes the additional contribution by dark matter density to match the observation.

In the first case, one would need to assume that for $r > r_0$ the mass distribution is not zero, but follows a special profile:

$$v = \sqrt{ar} \propto v_{out} \implies a = \frac{4G\rho\pi r^3}{3r^2} \propto \frac{1}{r} \implies \rho \propto \frac{1}{r^2}. \tag{1.3.16}$$

Here, v_{out} is a constant velocity. What was just given is an extremely simplistic solution, although it captures the essence of the approach, namely, to find a mass density able to recreate flat rotation curves outside of the visible baryonic matter core of galaxies. More realistic distributions are, for example, those described in [10], such as:

$$\rho = \frac{\rho_0}{\frac{r}{R_s}\left(1 + \frac{r}{R_s}\right)^2}, \tag{1.3.17}$$

with ρ_0 the density of the visible baryonic matter and R_s a parameter depending on the structure of the dark matter "halo".

1.4. Modified Newtonian Dynamics

1.4.1. Origin and Regimes of Validity

In the previous section, a description was given of the failure of Newtonian gravity to describe galaxy rotation curves without resorting to exotic, undetected forms of matter. It is perhaps best to introduce **Modified Newtonian Dynamics** (**MOND**) as a possible solution to this issue, before describing the theory in more detail.

If one accepts that Newton's laws might not be valid on the scale of galaxies, one can obtain flat rotation curves by an acceleration of the form:

$$\boxed{v = \sqrt{ar} \propto v_{out} \rightarrow a = |\nabla \phi| \propto \frac{1}{r}.} \tag{1.4.18}$$

For a spherically symmetric potential, this implies:

$$\boxed{\partial_r \phi \propto \frac{1}{r} \implies \phi \propto \ln(r)\,.} \tag{1.4.19}$$

This simple observation was in fact the basis of MOND. The theory was created to reproduce observations on galactic scales, especially rotation curves, utilising only the visible matter distribution, in stark contrast with dark matter theories.

First outlined by Milgrom in 1983 [11] and by Milgrom and Bekenstein in 1984 [12], MOND was initially expressed as one of two possible forms:

- A modification to Newton's second law: $F \rightarrow m\mu\left(a/a_0\right)a$;
- A modification to Newtonian gravity: $a \rightarrow a\mu\left(a/a_0\right)$.

In both cases, $\mu(x)$ is a free function of the theory, and a_0 is a new constant of nature with dimension of acceleration, Milgrom's constant. Although it can be noted that both formulations would result in the same effect on gravity, the former implies a different behaviour of all forces of nature. To avoid ambiguity, it should be pointed out that the latter form is used throughout this work.

> The free function, or to be more precise the interpolation function, $\mu\left(a/a_0\right)$ defines an acceleration scale, and the theory predicts a different dynamics from Newtonian gravity in the regime of low accelerations.

To see why this is the case, the PDE defining the theory will now be analysed.

1.4.2. MOND PDE and Interpolation Functions

The equation governing MOND is the non-linear Poisson equation:

$$\boxed{\nabla \cdot \left[\mu\left(\frac{|\nabla\phi|}{a_0}\right)\nabla\phi\right] = 4\pi G\rho,} \tag{1.4.20}$$

where ϕ is the modified gravitational potential.

The RHS of the MOND equation coincides with that of the Newtonian Poisson equation. This implies that no additional sources are introduced in the model.

The interpolation function $\mu(x)$, crucial to ensure the predictions of the theory match observations, has the asymptotic behaviour:

$$\lim_{x\to\infty}\mu(x) = 1, \quad \lim_{x\to 0}\mu(x) = x, \tag{1.4.21}$$

which is satisfied by various forms, proposed in [12] and [13]. The most important are:

1. Standard: $\mu(x) = \frac{x}{\sqrt{1+x^2}}$
2. Simple: $\mu(x) = \frac{x}{1+x}$
3. Exponential: $\mu(x) = 1 - e^{-x}$
4. Piecewise: $\begin{cases} x & x < 1 \\ 1 & x > 1 \end{cases}.$

Each interpolation function has a different behaviour between the two asymptotes, as shown in fig.1.4.1. It is hence clear that in the two asymptotes, the LHS of eq. (1.4.20) takes on the forms:

$$\frac{|\nabla\phi|}{a_0} \gg 1 \implies \nabla^2\phi = 4\pi G\rho, \tag{1.4.22}$$

$$\frac{|\nabla\phi|}{a_0} \ll 1 \implies \nabla \cdot \left[\frac{|\nabla\phi|}{a_0}\nabla\phi\right] = 4\pi G\rho. \tag{1.4.23}$$

Equivalently, one can define two asymptotic regimes:

For $|\nabla\phi| \gg a_0$ MOND reproduces Newtonian gravity. For $|\nabla\phi| \ll a_0$ one is in the **Deep MOND** regime. Milgrom's constant has the value $a_0 = 1.2 \times 10^{-10} m/s^2$

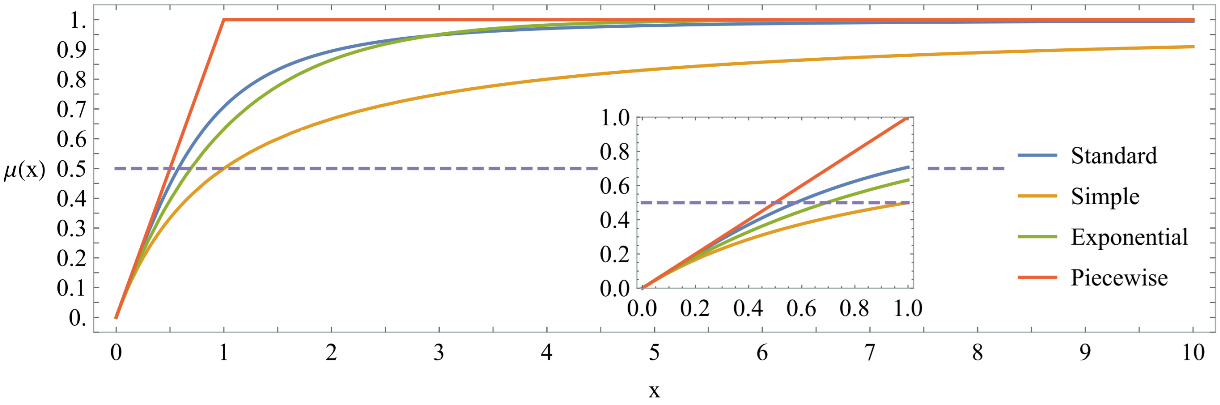

Figure 1.4.1: Possible interpolation functions $\mu(x)$. The x axis represents the argument of the function, namely the ratio between the acceleration of a given test particle and the acceleration scale a_0. It can be seen that for $x \approx 6$ all interpolations are close to unity, implying the Newtonian regime $\mu(x) \approx 1$. Conversely, for $x \lessapprox 0.2$ all interpolation are to good approximation linear, $\mu(x) \approx x$

1.4.3. The Deep MOND Regime

Stars in the outskirts of galaxies often have accelerations in the Deep MOND regime. It is hence possible to explain their rotation speeds through the MOND equation. The PDE for this regime reads:

$$\nabla \cdot \left[\frac{|\nabla \phi|}{a_0} \nabla \phi \right] = 4\pi G \rho. \tag{1.4.24}$$

In order to show that this gives a logarithmic potential as in eq. (1.4.19), which is required to obtain flat rotation curves, one can take the simple case of a test mass outside of a spherical mass distribution. Treating the mass distribution as a concentrated point source at the origin, one then obtains:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(\frac{r^2}{a_0}\left(\frac{\partial \phi}{\partial r}\right)^2\right) = 4\pi m G \delta\left(\vec{r}\right), \tag{1.4.25}$$

where m is the total mass of the system. A boundary condition can be derived from the requirement that the acceleration vanishes at infinity:

$$\left.\frac{\partial \phi}{\partial r}\right|_{r\to\infty} = 0. \tag{1.4.26}$$

The above can be solved by repeated integration over a large spherical volume at the boundary of which the acceleration is taken to vanish. For the LHS one obtains:

$$\int \frac{1}{r^2}\frac{\partial}{\partial r}\left(\frac{r^2}{a_0}\left(\frac{\partial \phi}{\partial r}\right)^2\right) 4\pi r^2 \mathrm{d}r = 4\pi \frac{r^2}{a_0}\left(\frac{\partial \phi}{\partial r}\right)^2 + C, \tag{1.4.27}$$

with C an integration constant. For the RHS the integration property of the Dirac Delta function is used, namely:

$$\int_V \delta\left(\vec{r}\right) \mathrm{d}V = 1, \tag{1.4.28}$$

which, after rearranging, yields the complete expression:

$$\left(\frac{\partial \phi}{\partial r}\right)^2 = \frac{1}{r^2}\left(mGa_0\right), \tag{1.4.29}$$

where the integration constant was set to $C = 0$. This result shows that the acceleration has the form:

$$a = \sqrt{Gma_0}\frac{1}{r}, \tag{1.4.30}$$

with a corresponding potential:

$$\phi = \sqrt{Gma_0}\ln r. \tag{1.4.31}$$

This simple analysis shows that the theory can provide a solution that qualitatively produces flat rotation curves far from the center of mass in a galaxy.

The value of a_0 was chosen to match observations made after analysing a collection of rotation curve data.

1.4.4. The Baryonic Tully-Fisher Relation

The success of MOND in inferring rotation curves from visible mass alone is remarkable. Many other predictions are made by MOND, and the most relevant will now be briefly discussed to highlight its empirical predictive power based solely on visible mass. In 1977, it was observed that the **luminosity**[5] of a galaxy is correlated to the asymptotic rotational speed[6] of the stars that reside in it [14] . This empirical law was named the Tully-Fisher relation after the two astronomers who discovered it.

Furthermore, it was later shown by McGaugh that the relation did not hold for **Low Surface Brightness** (**LSB**) galaxies. In this type of galaxy, gas makes up a non-negligible fraction of the total mass, which can hence no longer be inferred by stellar luminosity alone. It was instead shown in [15] [16] that, by accounting for the gas mass, one could obtain a fit for both High and Low Surface Brightness Galaxies.

The baryonic Tully Fisher relates the total baryonic mass of a galaxy to its asymptotic rotational velocity as $M_b \propto V_r^4$.

The ability of the baryonic Tully-Fisher relation to universally fit galaxy data, regardless of the ratio of stellar mass to total mass is highlighted in 1.4.2, taken from [15]. In order to see the connection between this relation and the MOND potential it must be noted

[5] Luminosity is a measure of the electromagnetic radiation reaching an observer from a source.

[6] Asymptotically, rotation curves are flat predicted by MOND.

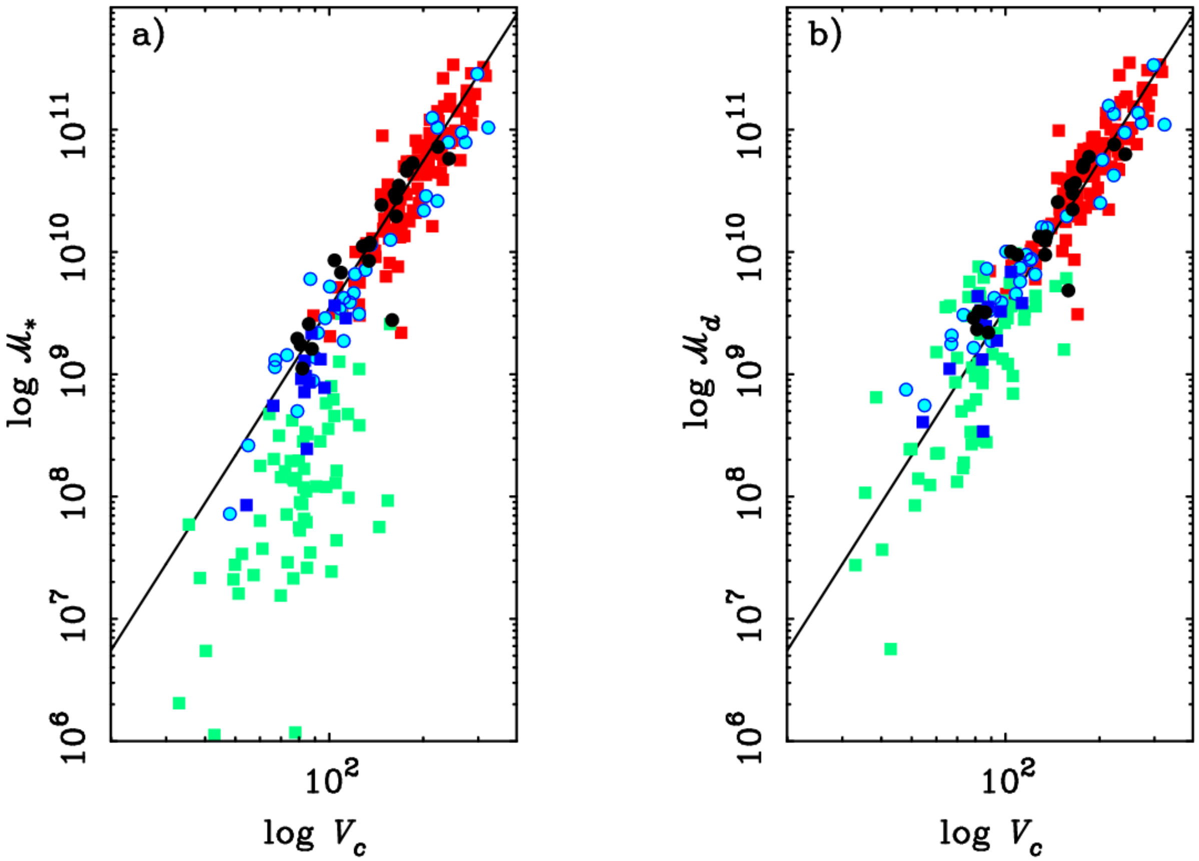

Figure 1.4.2: The figure on the left gives a representation of the Tully-Fisher relation after converting luminosity to stellar mass. It is clear that for galaxies with lower stellar masses the relationship no longer holds. On the other hand, when the gas mass is included to obtain the total baryonic mass, one can see that the fit is good regardless of the ratio between stellar mass and gas mass. The markers represent the different techniques used to obtain the data and are not relevant to the current discussion. Figure taken from [15].

that: for stars away from the galactic center, one can assume that the gravitational pull can be approximated by that of a point mass of mass M_b, the total mass of the baryons including stars and gas. The solution from the spherically symmetric case, found in eq. (1.4.30) of the previous section, can then be used. When the velocity for circular orbits predicted by this acceleration is calculated, one obtains:

$$V_r = \sqrt{ar} = \left(\sqrt{GM_b a_0}\frac{1}{r}r\right)^{\frac{1}{2}} = (GM_b a_0)^{\frac{1}{4}} \,. \tag{1.4.32}$$

The baryonic Tully-Fisher relation can be recovered directly by additionally fixing the value of the constant of proportionality as:

$$\boxed{M_b = \frac{V_r^4}{Ga_0}.} \tag{1.4.33}$$

Therefore, the baryonic Tully-Fisher relation is extremely important as it confirms the general concept inferred by rotation curve measurements:

In MOND, gravitational phenomena are entirely determined by baryonic matter.

1.5. General Relativity

Certain aspects of gravity cannot be described by either Newton or MOND. This is due to the fact that these theories are nonrelativistic: space and time are treated independently. Phenomena such as the precession of the orbit of Mercury and the bending of light by a massive body cannot be coherently explained in these frameworks.[7] Similarly, cosmology relies heavily on the principles of relativity.

It can be shown (see for example [6]) that Newtonian gravity is retrieved from General Relativity in the case of slow moving objects and weak-fields (bodies with low mass densities, with $M/r \ll M/r_s$, with r_s the Schwarzschild radius). As a result, this is an excellent approximation in the Solar System, but does not match the surroundings of a black hole or the dynamics of fast moving cosmic rays.

1.5.1. The Metric Tensor

General Relativity (GR) represents a natural extension of Special Relativity (SR). The latter unifies space and time into a 4D spacetime, which is described by the **metric tensor** $\eta_{\mu\nu}$. In both theories, the distance between two points is given by the **spacetime interval**. For SR, this is given by[8]:

$$\mathrm{d}s^2 = \eta_{\mu\nu}\mathrm{d}x^\mu\mathrm{d}x^\nu = -\mathrm{d}t^2 + \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2. \tag{1.5.34}$$

The above formula describes the flat **Minkowski** spacetime[9]. It is valid if no gravitational effects are present. On the other hand, GR allows for a more general form of the metric tensor:

[7] Surprisingly, it was shown in 1804 by von Soldner that in the limit of a vanishing mass, Newtonian gravity would predict a bending due to the presence of a massive body. This turned out to reproduce the general relativistic result up to a factor of 2. For a review and English translation, see [17].

[8] This is in fact the square of the spacetime interval, comparable to r^2 in Euclidean space. Unlike in Euclidean space, the notion of negative path length is well defined, and zero distance can exist for points which are not coinciding.

[9] It should be noted that, for the units to agree, the metric should be: $\mathrm{d}s^2 = -c^2\mathrm{d}t^2 + \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2$. However, from here on, $c = 1$ is assumed so that space and time have the same dimension.

General Relativity relates the curvature of the spacetime to the effects associated with gravity.

1.5.2. The Geodesic Equation

What is described as a force in Newtonian gravity is explained in GR as the motion of particles on straight lines in the spacetime. These straight lines are called **geodesics**[10], and their form depends on the metric tensor, which describes the background spacetime. One can conider geodesics as the shortest path between two points, and make an analogy with surfaces: for a flat spacetime (analogous to a 2D plane) the geodesics will be equivalent to the notion of a straight line, and all points in the path will be coplanar. However, for a curved spacetime (analogous to the surface of a sphere, or a deformed membrane), the shortest path between two points on a surface will no longer be straight. Nonetheless, it will be shorter than any other path connecting the starting and ending point. The statement about particles travelling along geodesics is the equivalent to Newton's first law of motion: unless acted upon by a force, a body will move at a constant velocity. In General Relativistic terms:

Bodies in free fall travel along a geodesic.

It was previously shown in (1.2.3) that, according to Newton, the acceleration a particle undergoes due to gravity does not depend on its mass. This followed from the equivalence of inertial and gravitational mass. This equivalence is just as fundamental in GR, and is referred to as the **weak equivalence principle**.

To show that unaccelerated particles travel along geodesics, one introduces the **covariant directional derivative**, which quantifies how much a vector (more generally, a tensor of any rank) fails to stay parallel to itself along a given path.[11] A path which keeps a tensor parallel to itself at all points is therefore the curved space equivalent of keeping that tensor constant in flat space. Calling the path $x^\mu(\lambda)$, $\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}$ is the tangent vector to the path itself, and the formal requirement is the following:

$$\frac{D}{\mathrm{d}\lambda}\left(\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}\right) = 0. \tag{1.5.35}$$

This is the **geodesic equation**. Here, $\frac{D}{\mathrm{d}\lambda}$ is the covariant directional derivative, which

[10] Particles only travel along a geodesic when no force is acting on them. Hereby, gravity itself cannot be classified as a force.

[11] With the 2D analogy, in a flat space such as a plane all elements of a vector are unchanged when the vector's origin is changed. This is not true for curved spaces, e.g. the surface of a sphere: the same tangent vector at two separate points generally does not point in the same direction. When displacing a vector, the final direction depends on the path taken.

vanishes at all points on the path. In synthesis, (1.5.35) states that the tangent vector is constant along the path.

A constant tangent can be geometrically interpreted as a constant derivative of the curve representing the path: this can only be the case for a linear function. The geodesic is hence the curved space equivalent of a straight line.

Eq. (1.5.35) can be rewritten as:

$$\boxed{\frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\lambda^2} + \Gamma^\mu_{\rho\sigma}\frac{\mathrm{d}x^\rho}{\mathrm{d}\lambda}\frac{\mathrm{d}x^\sigma}{\mathrm{d}\lambda} = 0.} \tag{1.5.36}$$

The new symbols $\Gamma^\mu_{\rho\sigma}$ were introduced in the process. They are called the **Christoffel symbols** (also called Christoffels) and quantify the curvature of the spacetime. They do so by describing how the derivative of a quantity with respect to each coordinate is affected by the remaining coordinates. For a space with no curvature, the symbols vanish at all points[12] and one simply retrieves the notion that any path along which the second derivative is zero corresponds to a straight line. Non vanishing Christoffels generalise this notion to curved space(time)s, so geodesics are no longer restricted to straight lines[13]. The Christoffels can be defined through the geodesic equation itself, as given in eq. 1.5.36. In addition, they can also be defined by using the partial derivatives of the metric tensor $g_{\mu\nu}$, as:

$$\boxed{\Gamma^\sigma_{\mu\nu} = \frac{1}{2}g^{\sigma\rho}\left(\partial_\mu g_{\nu\rho} + \partial_\nu g_{\rho\mu} - \partial_\rho g_{\mu\nu}\right).} \tag{1.5.37}$$

1.5.3. The Covariant Derivative

For any differential calculation, one must ensure that the concept of keeping a quantity constant is well defined. The use of a partial derivative assumes that the components of a tensor are defined equally at all points in spacetime. In a curved background, it has been shown above that this assumption no longer holds: one must account for the component of the variation that stems from parallel transporting the quantity along a geodesic, irrespective of any change in the quantity itself[14]. One then introduces a **covariant derivative** as follows:

[12] To be precise, for a flat space one can always choose coordinates where all Christoffels vanish. Curvilinear, such as spherical, coordinates in flat space also generate non-zero Christoffels, but these can be eliminated by choosing a different coordinate system, such as Cartesian.

[13] This makes intuitive sense along a sphere, on which the shortest path between two points is an arc rather than a straight line.

[14] As previously explained, the components of a vector parallel transported on the surface of a sphere will mix with each other along the path, although the vector itself represents the same quantity. The effect is clearly coordinate dependent, as changing the coordinate system will restore the original components.

> Differentiation in curved spacetime must ignore any variation that is due exclusively to the chosen coordinate system. The covariant derivative describes coordinate-independent variations.

It is then a natural conclusion to define the covariant derivative as the partial derivative plus a correction accounting for the curvature of spacetime so that the expression is valid for any coordinate system. The correction is the Christoffel symbol, and the covariant derivative acting on a 4-vector u^α and respective co-vector u_α is defined as[15]:

$$\boxed{\nabla_\mu u^\alpha = \partial_\mu u^\alpha + \Gamma^\alpha_{\mu\nu} u^\nu, \quad \nabla_\mu u_\alpha = \partial_\mu u_\alpha - \Gamma^\nu_{\mu\alpha} u_\nu.} \tag{1.5.38}$$

What quantity can then give an unambiguous description of the the curvature of a spacetime? To answer this question, one can again refer to the surface of a sphere. One can take a vector and two independent paths along which it can be moved. The final direction of the vector will depend on the order in which the two paths are taken. The difference in the final direction of the vector will give a measure of the curvature of the sphere's surface.

Generalising this concept, the aim is to quantify the variation due solely to the curvature, and not the influence of an arbitrary coordinate system. It is natural to use the covariant derivative. Moreover, the difference of the order of the paths taken on the final direction of the vector is given by the commutator of the two covariant derivatives along each path. This commutator is known as **Riemann Tensor** $R^\rho{}_{\sigma\mu\nu}$:

> The Riemann Tensor quantifies the curvature of a spacetime as the coordinate independent variation in the value of a tensor, when this tensor is parallel transported along two infinitesimal paths in a different order.

Mathematically[16]:

$$[\nabla_\mu, \nabla_\nu]\, V^\rho = R^\rho{}_{\sigma\mu\nu} V^\sigma. \tag{1.5.39}$$

Two important quantities can be derived from the Riemann Tensor: the Ricci tensor $R_{\mu\nu}$ and the Ricci scalar R. They are defined as:

$$\boxed{R_{\mu\nu} = R^\lambda{}_{\mu\lambda\nu}, \quad R = R^\mu_\mu} \tag{1.5.40}$$

[15] The generalisation to a tensor of rank n is obtained by repeating the process for each of the n indices.

[16] In reality, another term should be present on the RHS accounting for the torsion of the metric, that is the rotation of a body travelling along a geodesic. However, in GR all components of the torsion tensor are zero and the quantity will be neglected for the present discussion.

1.5.4. The Einstein Equations

Curvature has now been defined. The last step is to explain where it originates from. To do so, one has to introduce the **Einstein Equations**[17]:

$$\boxed{G_{\mu\nu} = 8\pi G T_{\mu\nu}.} \tag{1.5.41}$$

The LHS is the divergenceless[18] component of the Ricci tensor, the **Einstein tensor**. It quantifies the curvature of the spacetime and is defined as:

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R. \tag{1.5.42}$$

The RHS is the **Energy-Momentum tensor**, generalising the concept of energy (such as mass or radiation) for a given field.[19] In words, (1.5.42) states the following:

> Spacetime is curved by the presence of energy.

In the absence of energy, that is to say in vacuum, the following holds for any solution of the Einstein equations:

$$G_{\mu\nu} = 0 \implies R_{\mu\nu} = \frac{1}{2} g_{\mu\nu} R. \tag{1.5.43}$$

To show that this implies that both $R_{\mu\nu} = 0$ and $R = 0$, one can use the following relation for the Kronecker delta δ^ρ_ν:

$$g_{\mu\nu} g^{\mu\rho} = \delta^\rho_\nu. \tag{1.5.44}$$

This then implies that:

$$g_{\mu\nu} g^{\mu\nu} = \delta^\mu_\mu = 4. \tag{1.5.45}$$

Thus, multiplying both sides of eq. 1.5.43 yields:

$$R_{\mu\nu} = \frac{1}{2} g_{\mu\nu} R \implies R_{\mu\nu} g^{\mu\nu} = \frac{1}{2} g_{\mu\nu} g^{\mu\nu} R \implies R = 2R. \tag{1.5.46}$$

Finally, since the above can only be true if $R = 0$, and given that $g_{\mu\nu} \neq 0$, the following is implied:[20]

$$R_{\mu\nu} = R = 0. \tag{1.5.47}$$

However, the Ricci tensor and Ricci scalar only account for the trace of the Riemann tensor, which describes the change in the volume of a body in the spacetime. The traceless component of the Riemann tensor is called the **Weyl Tensor** $C_{\rho\sigma\mu\nu}$, and it accounts for

[17] G is Newton's constant, not the trace of the Einstein tensor.

[18] A divergenceless tensor is needed as the RHS is divergenceless by construction.

[19] The components of the energy momentum tensor describe energy density and momentum density, as well as the corresponding fluxes.

[20] In the expression $g_{\mu\nu} \neq 0$, 0 is the second rank zero tensor, not a scalar.

the deformation of a body that does not cause a change in its volume. For a spacetime with n dimensions, it is defined as:

$$C_{\rho\sigma\mu\nu} = R_{\rho\sigma\mu\nu} - \frac{2}{n-2}\left(g_{\rho[\mu}R_{\nu]\sigma} - g_{\sigma[\mu}R_{\nu]\rho}\right) + \frac{2}{(n-1)(n-2)}g_{\rho[\mu}g_{\nu]\sigma}R \tag{1.5.48}$$

The Weyl tensor gives the inherent curvature of the spacetime that does not originate from the presence of energy. For example, since it can be non-zero in free space, it describes the propagation of gravitational waves in vacuum.

1.6. Emergent Gravity

As mentioned in section 1.1, no one has so far managed to provide a framework capable of quantising General Relativity while avoiding the appearance of non-renormalisable infinities. On the other hand, in the second half of the twentieth century, the idea that gravity might be an emergent phenomenon, rather than a force mediated by a quantised particle, started to be taken seriously. This section gives a brief account of these developments, leading up to Verlinde's theory of Emergent Gravity (EG) and its consequent explanation of the phenomenology attributed to dark matter.

1.6.1. Origin and Formulations

Although extremely successful on an empirical scale, MOND lacks a foundation based on underlying principles motivating its predictions. The introduction of a low acceleration regime is postulated rather than derived, and the magnitude of Milgrom's constant a_0 is set in order to satisfy rotation curve data. Nonetheless, Jacob Bekenstein, one of the two founders of the theory, was a pioneer in the exploration of the boundary between General Relativity and Quantum Mechanics with his study of black hole thermodynamics. In [18], Bekenstein postulated an analog to the classical thermodynamic equation of state,

$$\mathrm{d}E = T\mathrm{d}S - P\mathrm{d}V, \tag{1.6.49}$$

with E, T, P and V being energy, entropy, pressure and volume respectively, purely in terms of the parameters describing a black hole:

$$\mathrm{d}M = \overbrace{\Theta\mathrm{d}\alpha}^{T\mathrm{d}S} + \overbrace{\vec{\Omega}\cdot\mathrm{d}\vec{L} + \Phi\mathrm{d}Q}^{-P\mathrm{d}V}. \tag{1.6.50}$$

Here $\vec{\Omega}$ is the rotational angular frequency, Φ the electric potential, M the mass, and Θ the analog of the temperature T. All mentioned quantities are scaled by a factor α, the rationalised area. Jacobson further developed the connection between entropy and spacetime, deriving the Einstein relations from the thermodynamic equation of state (1.6.49). However, Jacobson used Killing horizons rather than black hole horizons. He utilised Rindler coordinates to draw a parallel between the thermal spectrum detected by an accelerating Rindler observer, and the energy flux flowing across the local Rindler horizons in the form of heat, with $\delta Q = T\delta S$ [19].

1.6.2. Verlinde's approach

The main point that Verlinde describes in his first paper about EG [20] is the following:

> The concept of gravity is tightly correlated to that of information, and hence entropy. Gravity is an entropic force.

Quantities related to matter, first and foremost its position, contain information. A displacement is hence associated with a change in entropy, which determines a reaction force. An analogy is given as follows: consider a polymer in a heat bath. One can stretch the polymer by pulling it out of the bath, causing it to be straightened out and reducing its entropy. In the absence of an external force, the polymer will end up in a state maximising its entropy, corresponding to being curled up in a random fashion. This entropic reaction will cause a displacement of the previously straightened polymer leg, hence determining a force:

> An entropic force stems from the tendency of a system to return to a state of maximum entropy, and is not mediated by a specific field or particle.

This process has a different nature from the force carried by a field, and its quantised bosonic excitation: entropic forces do not require a quantisation such as that carried out for all the fundamental forces described by the SM. This is a potential way out of the theoretical difficulties associated with quantising gravity:

> Gravity might not have a quantised force carrier.

Another important association is made regarding the amount of information contained in a volume of space, through the holographic principle. This states that the information contained in a (hyper)volume can be determined by the properties of the (hyper)surface that bounds it. Developing the idea further in [21], Verlinde identifies an elastic phase of gravity. In this framework, the gravitational potential is described by means of a displacement field of the form:

$$\boxed{u_i = \frac{\phi}{\tilde{a}_0} n_i}. \tag{1.6.51}$$

The gravitational potential is indicated by ϕ, whereas n_i is a unit vector in $3D$ and $\tilde{a}_0$ is later equated to a scaled version of Milgrom's constant, $\tilde{a}_0 = 6a_0$. The strain tensor for the elastic displacement is also defined as:

$$\epsilon_{ij} = \frac{1}{2}\left(\partial_i u_j + \partial_j u_i\right). \tag{1.6.52}$$

This quantity is connected to the gravitational acceleration through the relation:

$$\epsilon_{ij} n_j = -\frac{g_i}{\tilde{a}_0}. \tag{1.6.53}$$

The main result of this approach is a relationship between the apparent dark matter distribution[21] and the gravitational potential depending exclusively on the amount of baryonic matter[22]:

$$\boxed{\left(\frac{8\pi G}{a_0}\Sigma_D\right)^2 = \left(\frac{d-2}{d-1}\right)\nabla_i\left(\frac{\phi_B}{a_0}n_i\right).} \tag{1.6.54}$$

In the above, Σ_D is the surface mass density for the apparent dark matter, and d is the dimension of the spacetime. The most important observation for a generalisation of the theory regards the displacement vector:

> Effects associated with dark matter can be explained by a gravitational potential carried by a vector function.

The relevance of this aspect will be made clearer when discussing modifications of GR. The consequence of (1.6.54) is an acceleration g_D due to apparent dark matter which takes the form:

$$\boxed{g_D = \sqrt{a_0 g_N} = \sqrt{Gma_0}\frac{1}{r},} \tag{1.6.55}$$

with g_N the Newtonian gravitational acceleration. This corresponds fully to the prediction for a point source in the deep MOND limit, as shown in (1.4.30). An important difference must, however, be pointed out:

> In EG the total acceleration depends on the sum of the baryonic and apparent DM components: $g_{tot} = g_B + g_D$, where the latter only appears after a threshold acceleration. In MOND, only one acceleration is present, and it depends on the chosen function $\mathcal{F}(u)$ appearing in the Lagrangian.

1.7. Lagrangian Formalism

The equations of motion for a classical particle determine its evolution based on two variables: its position x and its speed $\dot{x}$[23]. These fundamental quantities can be obtained through three equivalent formalisms: **Newtonian**, **Lagrangian** and **Hamiltonian**. As

[21] The assumption that gravity is fully determined by baryonic matter places Verlinde's theory in the family of modified gravity theories alongside MOND

[22] It is shown in chapter 2 that in its current form, this relation does not make the prediction that Verlinde implies. A correction is suggested alongside this observation. It should also be noted that a_0 can here be used in place of $\tilde{a}_0$ as the factors cancel on either side.

[23] A similar analysis can be applied to fields.

the present work focuses on the Lagrangian approach, a brief overview of this method will now be given.

1.7.1. Lagrangian and Lagrangian Density

The Lagrangian is a scalar function [24]. When working with fields rather than particles, as will now be the case in this work, one needs to account for the behaviour of the field in all points in space. To do so, the **Lagrangian** is defined in terms of a **Lagrangian Density** L_D:

$$L = \int L_D\left(\phi,\ \partial_\mu \phi\right) \mathrm{d}^3 x, \tag{1.7.56}$$

with ϕ the field and $\partial_\mu \phi$ its partial derivative[25].

1.7.2. Notation and Unit System

For the sake of clarity, standard notation will be defined here. Latin indices run from 1 to 3 and describe three spatial dimensions[26]. Greek indices denote spacetime indices and run from 0 to 3:

$$A^i = \left(A^1, A^2, A^3\right), \quad B^\mu = \left(A^0, A^1, A^2, A^3\right) \tag{1.7.57}$$

Three equivalent notations are used for partial derivatives:

$$\frac{\partial \phi}{\partial x^\mu} = \partial_\mu \phi = \phi_{,\mu} \, . \tag{1.7.58}$$

Regarding dimensions, natural units are used. In these units, speed is set to be dimensionless ($c = 1$), energy to have inverse time dimension ($\hbar = 1$), and the other necessary physical units are inferred using the formula for the Compton wavelength. This relates relativistic (the speed of light c) and quantum mechanical (the energy of a quantum of the EM field $\hbar$) scales through the equivalence in energy between a photon of wavelength λ and a particle of mass m:

$$\lambda = \frac{\hbar}{mc}. \tag{1.7.59}$$

This implies that:

$$[\text{time}] = [\text{distance}] = [\text{mass}]^{-1}, \quad [G] = [mass]^{-2} \tag{1.7.60}$$

When necessary, physical quantities will be analysed according to their mass dimension.

[24] More precisely a Lorentz scalar, invariant under Lorentz transformations. However, a class of theories breaking Lorentz invariance will be analysed in later sections, so it is best to leave the description general.

[25] In curved spacetimes, one has to replace the partial by a covariant derivative, or include the Jacobian in the Lagrangian density. The two methods are equivalent, and the former is used throughout this work.

[26] The coordinates are not restricted to Cartesian but can equally denote e.g. cylindrical or spherical coordinates.

1.7.3. Equations of Motion

To obtain the equations of motion from a Lagrangian, one can use a **variational** approach. The goal is to find a path through time that minimises the Lagrangian. This is done through a functional S called the **Action**, defined as:

$$S = \int L \, \mathrm{d}t = \int L_D \, \mathrm{d}^4 x. \tag{1.7.61}$$

Setting the variation of the action with respect to the field to zero yields the **Euler Lagrange** (EL) Equations Of Motion (EOM)[27]:

$$\boxed{\frac{\delta S}{\delta \phi} = \frac{\partial L_D}{\partial \phi} - \partial_\mu \left(\frac{\partial L_D}{\partial \left(\partial_\mu \phi \right)} \right) = 0}, \tag{1.7.62}$$

where $\frac{\delta S}{\delta \phi}$ is the functional derivative. As can be seen, the quantity entering the EOM is L_D rather than L, so the former will be referred to as the Lagrangian L throughout the rest of this work, as is customary in the literature.

1.7.4. Example EOM

Given that most of the work presented in the following chapters is based on obtaining the EOM from a Lagrangian, a few examples will now be given for the three theories discussed above. These show the power of the formalism in defining a self consistent theory predicting the behaviour of all fields involved with just a few steps.

Newtonian Gravity.— As shown in section 1.2, Newtonian gravity is fully defined by the Newton-Poisson equation. However, it has not yet been shown how to derive this expression. One method of doing so is to use the following nonrelativistic Lagrangian L_N for the Newtonian potential field ϕ_N:

$$L_N = - \left(\rho \phi_N + \frac{\left(\nabla \phi_N \right)^2}{8 \pi G}. \right) \tag{1.7.63}$$

The EL equations for ϕ_N immediately reproduce the Newton-Poisson equation.

MOND.— As shown in [12], the full modified Poisson equation for the MOND field ϕ can be retrieved from the Lagrangian:

$$\boxed{L_M = - \left(\rho \phi + \frac{a_0^2}{8 \pi G} \mathcal{F} \left[\frac{\left(\nabla \phi \right)^2}{a_0^2} \right] \right).} \tag{1.7.64}$$

[27] Again, covariant derivatives replace partials in curved spacetime.

Here $\mathcal{F}\left(x^2\right)$ is a free function. It is convenient to make the substitution $u = x^2$. It can then be seen that (1.7.63) and (1.7.64) coincide for $\mathcal{F}(u) = u$. The reason for this can be deduced upon derivation of the EL equations, where one can see that:

$$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}u} = \mu(x), \quad \mathcal{F}(u) = u \implies \mu(x) = 1. \tag{1.7.65}$$

On the other hand, the deep MOND regime is obtained for $\mu(x) = x = u^{1/2}$:

$$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}u} = u^{1/2} \implies \mathcal{F}(u) = \frac{2}{3}u^{3/2} + A, \tag{1.7.66}$$

where A is a constant of integration. It is important to note the following:

> The function $\mathcal{F}(u)$ reproduces deep MOND behaviour for $\mathcal{F}(u) \propto u^{3/2}$, and Newtonian gravity for $\mathcal{F}(u) \propto u$.

For any relativistic extension of the theory, it would be necessary to build a Lagrangian containing a similar term. This aspect will be discussed in more detail in the next chapter.

General Relativity.— Unlike the previous two cases, GR cannot be described by a single scalar field. Moreover, its EOM do not define a potential, but a spacetime metric. As shown in section 1.5, two main equations are needed to describe the dynamics of the spacetime: the geodesic equations and the Einstein equations. However, these equations originate from separate Lagrangian functions. The Lagrangian L_G generating the geodesic equations can be intuitively explained:

> As the geodesics define the shortest path between two spacetime points, the equations defining them originate from minimising the spacetime interval.

This is expressed as[28]:

$$L_G = \mathrm{d}s^2 = g_{\mu\nu}\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}\frac{\mathrm{d}x^\nu}{\mathrm{d}\lambda} \tag{1.7.67}$$

The EL equations take on a slightly different form, as the dynamic field being described is in fact the coordinate vector, and the differentiation is carried out with respect to proper time τ. The full EOM read[29]:

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}\tau}\left(\frac{\partial L}{\partial \dot{x^\mu}}\right) - \frac{\partial L}{\partial x^\mu} = \frac{\mathrm{d}^2 x^\mu}{\mathrm{d}\tau} + \Gamma^\mu_{\nu\rho}\frac{\mathrm{d}x^\nu}{\mathrm{d}\tau}\frac{\mathrm{d}x^\rho}{\mathrm{d}\tau} = 0.} \tag{1.7.68}$$

[28] To be precise, this expression gives the square of the distance between two points in spacetime. However, for timelike paths (those of massive particles), extremising the square is equal to extremising the distance. Geodesics for massless particles are also described by the resulting EOM.

[29] The following important observation must be made: The LHS has its free index lowered, whereas the RHS has it raised. However, as the LHS equals zero, raising its index will not change the result. Although formally incorrect, the relationship holds.

It can be seen from the above that the Christoffels can be calculated directly from the EOM.

The other Lagrangian concerns the curvature itself. It emerges that the Einstein equations are in fact the EOM for the metric tensor, which is the field defining GR. The only[30] independent scalar describing curvature is the Ricci scalar. It follows that it is the only candidate for the Einstein-Hilbert curvature Lagrangian L_{EH}:

$$L_{EH} = R. \tag{1.7.69}$$

1.7.5. The Necessity of a Relativistic Lagrangian

The outstanding experimental success of GR in predicting gravitational phenomena makes the following clear:

Any complete theory of gravity must match all the confirmed predictions of GR.

It follows that a fully fledged gravitational theory must have a relativistic nature. Consequently, it must be possible to make precise predictions on the relationship between the spacetime curvature and the presence of energy. It has been shown that these principles in GR directly follow from the two Lagrangians given above. Moreover, any observation regarding cosmology cannot be formally treated if not in a relativistic setting. It is hence of paramount importance to formulate the laws of proposed gravitational theories through a relativistic, covariant Lagrangian. In the next chapter, this approach will be described for a theory originating from the principles of EG which is capable of reproducing MOND phenomenology while being consistent with the GR framework.

1.8. Galaxy Clusters and Gravitational Lensing

So far, MOND has been treated as a single, well defined theory which is derived from a Lagrangian. However, the PDE originating from the Lagrangian in eq. 1.7.64 is not the only possible formulation of MOND. There are, in fact, three main formulations which are of interest:

1. The fully non-linear theory, named **AQUAdratic Lagrangian** (**AQUAL**), which is defined by the modified Poisson equation with interpolation function $\mu(x)$. This is the formulation which we have discussed so far. Since it is derived from a Lagrangian, AQUAL satisfies conservation laws such as the conservation of energy and momentum.

2. The **algebraic** (**also called pristine**) formulation, which does not require the solution of the non-linear PDE. Instead, the MOND acceleration g_A is algebraically

[30] There are others, but they all contain third order or higher derivatives of the metric.

related to the Newtonian acceleration via:

$$\vec{g_N} = \mu\left(\left|\vec{g_A}\right|/a_0\right)\vec{g_A}. \tag{1.8.70}$$

This formulation is not derived from a Lagrangian and, as shown in [11], it violates conservation of momentum.

3. The quasi-linear formulation, QUMOND, which provides a middle ground between the two formulations described above. Although this formulation is derived from a Lagrangian, and hence satisfies the required conservation laws, the QUMOND potential ϕ_Q is not the solution of a non-linear PDE. Instead, it is the solution to the Poisson equation with a modified mass density $\hat{\rho}$, which can be obtained from the pristine acceleration $\vec{g}_A$:

$$\nabla^2\phi_Q = \hat{\rho} = -\frac{1}{4\pi G}\nabla\cdot\vec{g}_A \tag{1.8.71}$$

It can be shown that the three formulations are equivalent in the case of a spherically symmetric mass distribution [22]. More generally, both the pristine formulation and QUMOND are simplified versions of AQUAL, and they are more tractable both analytically and numerically. However, the mass distribution in galaxy clusters is inherently non-spherical, as it contains, in addition to the smooth **Intra Cluster Medium** (**ICM**), up to thousands of galaxies. In order to determine the mass distribution inside a cluster, it is hence necessary to make use of AQUAL. This is especially true if we study gravitational lensing, as the correct mass distribution cannot be inferred by the use of QUMOND or pristine MOND for non-spherically symmetric distributions. However, most of the work that has been carried out in the analysis of weak lensing for galaxy clusters in MOND has utilised the pristine formulation, as in [23] and [24]. On the other hand, most of the work on N-body simulations in MOND has been carried out through the use of QUMOND [25]. In chapter 5, we will analyse the apparent mass distribution inferred from gravitational lensing measurements which assume the gravitational potential is Newtonian, and we will also examine the relationships between the dark and baryonic matter components in a non-spherically symmetric case for a variety of clusters with different ICM and galaxy distributions.

Chapter 2

Covariant Emergent Gravity: Current Status and Issues

It has been made clear in the previous chapter that the study of a theory of modified gravity should be conducted through the Lagrangian formalism. It would be desirable to recover both the predictions of GR in the relativistic setting and the phenomenology successfully explained by MOND in the limit of low acceleration. A good candidate to succeed in this feat is Verlinde's EG: in [20] the framework is shown to be able to reproduce Newtonian gravity, and in [21] the MOND phenomenology is recovered using key principles of GR[1]. However, the theory lacks a Lagrangian. It cannot, therefore, be analysed through the same formalism as GR and MOND. The MOND paradigm does not have a universally accepted relativistic formulation either[2].

In [26], Hossenfelder makes an attempt at deriving a Lagrangian for Verlinde's EG, capable of reproducing MOND phenomenology. Verlinde's displacement field is termed the imposter vector field and indicated by a 4-vector u^α. This theory is named **Covariant Emergent Gravity** (CEG).

2.1. The CEG Lagrangian

The main claim made in [26] is that EG, as formulated in [21], is internally inconsistent, meaning that different equations for the same variable, e.g. the main field of interest u^α, give expressions that are not equal to each other in the general case. This is due to the fact that heuristic arguments are used throughout [20] and [21]. Furthermore, the theory is not derived from a Lagrangian approach, which would allow for the definition of symmetries, conserved quantities, and equations of motion for each field present.

[1] String Theory is also frequently cited, but it is known that GR naturally emerges from most string theoretic formulations.

[2] Many relativistic expansions of MOND exist, amongst which BIMOND and TeVeS, but they present theoretical problems as well as clashing with observations. This point will discussed in the next chapter in more detail.

2.1.1. Quantities of Interest

In order to provide a theoretically sound background for Verlinde's theory to formally make contact with observations, the quantity of interest is the elastic displacement field, which is directly related to the gravitational potential ϕ in [21] by:

$$u_i = \frac{\phi n_i}{a_0}. \tag{2.1.1}$$

The starting point chosen by Hossenfelder to build a Lagrangian is the relationship between the surface mass density for the apparent DM and the gravitational potential due to baryonic matter ϕ:

$$\left(\frac{8\pi G}{a_0}\Sigma_D\right)^2 = \left(\frac{d-2}{d-1}\right)\nabla_i\left(\frac{\phi}{a_0}n_i\right) \tag{2.1.2}$$

The apparent DM surface mass density is defined in [21] as:

$$\Sigma_D = \frac{a_0}{8\pi G}\tilde{\epsilon}. \tag{2.1.3}$$

In (2.1.2), d represents the dimension of the spacetime, which in the case of interest is $d = 4$, and $\tilde{\epsilon}$ is the largest principal strain of the displacement field[3]. Plugging the definition (2.1.3) into (2.1.2), one arrives at:

$$\left(\frac{8\pi G}{a_0}\frac{a_0}{8\pi G}\tilde{\epsilon}\right)^2 = \frac{2}{3}\nabla_i\left(\frac{\phi}{a_0}n_i\right). \tag{2.1.4}$$

This gives an equation for $\tilde{\epsilon}^2$:

$$\boxed{\tilde{\epsilon}^2 = \frac{2}{3}\nabla_i\left(\frac{\phi}{a_0}n_i\right) = \frac{2}{3}\nabla_i u_i} \tag{2.1.5}$$

By taking a further covariant derivative on each side, the required formula is obtained:

$$\nabla_j\left(\tilde{\epsilon}^2\right) = \frac{2}{3}\nabla_j\nabla_i\left(u_i\right). \tag{2.1.6}$$

The strain is a symmetric tensor and is defined equivalently in [26] and [21] (up to an extra factor of $\frac{1}{2}$ in [21]) through[4]:

$$\boxed{\epsilon_{\mu\nu} = \nabla_\mu u_\nu + \nabla_\nu u_\mu.} \tag{2.1.7}$$

[3] $\tilde{\epsilon}$ is defined with a tilde to avoid confusion due to the different definitions given in [26] and [21] and used throughout this chapter.

[4] It should be noted that Verlinde repeatedly uses the notation for the covariant derivative without ever introducing Christoffel symbols. Therefore, he uses partial derivatives, rather than covariant ones.

Conversely, the quantity ϵ is defined through different expressions in [21] and [26](hence the tilde on $\tilde{\epsilon}$ when referring to the quantity from [21]). At this point it should be noted that from here on, indices will be used in the conventional way for tensor calculus as defined in section 1.7.2, to avoid confusion when comparing the expressions from [21] and [26].

Returning to $\tilde{\epsilon}$, Verlinde defines this quantity through the relation[5]:

$$\epsilon'_{ij} n_j = \tilde{\epsilon} n_i, \tag{2.1.8}$$

that is to say, the eigenvalue of the deviatoric stress tensor corresponding to the normalised eigenvector n_i of the displacement field u_i [21]. The deviatoric stress tensor, which is the traceless component of ϵ_{ij}, is defined as:

$$\epsilon'_{ij} = \epsilon_{ij} - \frac{1}{d-1} \epsilon^k_k \delta_{ij}. \tag{2.1.9}$$

It is the component of the strain responsible for deformation, but not for (hyper)volume change. On the other hand, ϵ is defined in [26] as the scalar contraction of the strain tensor:

$$\boxed{\epsilon = \epsilon^\mu_\mu.} \tag{2.1.10}$$

Nonetheless, for the following calculations one has no choice but to follow the definition given by Hossenfelder of $\tilde{\epsilon^2} \equiv \chi$, which allows for the most general definition of the kinetic energy of the relevant field u^α:

$$\boxed{\tilde{\epsilon}^2 = \chi = \left(\nabla_\mu u^\nu\right)^2.} \tag{2.1.11}$$

However, $\left(\nabla_\mu u^\nu\right)^2$ is not a properly defined tensor, but it can be made formally correct by defining it as a linear combination of all the possible contractions between the indices μ and ν of the derivatives, or equivalently of the strain tensor defined in eq. (2.1.7):[6]

$$\chi = \zeta \left(\epsilon^\mu_\mu\right)^2 + \iota \left(\epsilon_{\mu\nu}\epsilon^{\mu\nu}\right) + \xi \left(\epsilon^{\mu\nu\sigma\rho}\epsilon_{\mu\nu}\epsilon_{\sigma\rho}\right). \tag{2.1.12}$$

The fully antisymmetric Levi-Cività symbol $\epsilon^{\mu\nu\sigma\rho}$ should not be confused with the strain tensor $\epsilon_{\mu\nu}$, and is defined as (see for example [27]):

$$\epsilon^{\mu\nu\sigma\rho} = \begin{cases} -1 & \text{even permutations of } \mu\nu\sigma\rho \\ 1 & \text{odd permutations of } \mu\nu\sigma\rho \\ 0 & \text{otherwise} \end{cases}$$

[5] Again, Verlinde uses all lowered Latin indices.

[6] Hossenfelder translates all 3D quantities quantities used by Verlinde into 4D quantities with the same properties and normalisation. It is hence to be expected that the gravitational acceleration is fully defined by the symmetric 4D strain tensor $\epsilon_{\mu\nu}$. Problems with this approach are highlighted at the end of the chapter.

As $\epsilon^{\mu\nu\sigma\rho}$ is fully antisymmetric and $\epsilon_{\mu\nu}$ fully symmetric, their contraction vanishes, giving the general expression:

$$\chi = \zeta \left(\epsilon^{\mu}_{\mu}\right)^2 + \iota \left(\epsilon_{\mu\nu}\epsilon^{\mu\nu}\right) \tag{2.1.13}$$

It can be shown that this approach is in fact equivalent to the one adopted in [26], where the kinetic term $\chi = \tilde{\epsilon}^2$ is defined as the sum of all possible contractions giving a quadratic derivative, with arbitrary coefficients α, β and γ:

$$\chi = \alpha \overbrace{(\nabla_\nu u^\nu)(\nabla_\mu u^\mu)}^{A} + \beta \overbrace{(\nabla_\mu u_\nu)(\nabla^\mu u^\nu)}^{B} + \gamma \overbrace{(\nabla_\mu u_\nu)(\nabla^\nu u^\mu)}^{C}. \tag{2.1.14}$$

Although the choice of the constants is not unique, Hossenfelder decides to use:

$$\alpha = \frac{4}{3}, \quad \beta = \gamma = -\frac{1}{2}. \tag{2.1.15}$$

It can be noted that

$$\begin{aligned}\epsilon_{\mu\nu}\epsilon^{\mu\nu} = (\nabla_\mu u_\nu + \nabla_\nu u_\mu)(\nabla^\mu u^\nu + \nabla^\nu u^\mu) = \\ (\nabla_\mu u_\nu)(\nabla^\mu u^\nu) + (\nabla_\mu u_\nu)(\nabla^\nu u^\mu) + (\nabla_\nu u_\mu)(\nabla^\mu u^\nu) + (\nabla_\nu u_\mu)(\nabla^\nu u^\mu).\end{aligned} \tag{2.1.16}$$

One can then switch the dummy indices $\mu \leftrightarrow \nu$ in the first and second terms, obtaining

$$\begin{aligned}(\nabla_\nu u_\mu)(\nabla^\nu u^\mu) + (\nabla_\nu u_\mu)(\nabla^\mu u^\nu) + (\nabla_\nu u_\mu)(\nabla^\mu u^\nu) + (\nabla_\nu u_\mu)(\nabla^\nu u^\mu) = \\ 2(\overbrace{(\nabla_\nu u_\mu)(\nabla^\nu u^\mu)}^{B} + \overbrace{(\nabla_\nu u_\mu)(\nabla^\mu u^\nu)}^{C}).\end{aligned} \tag{2.1.17}$$

Similarly, from the definition of $\epsilon^{\mu}_{\mu} = \epsilon$ (which comes from (2.1.10), and hence lacks the tilde) one obtains:

$$(\epsilon^{\mu}_{\mu})^2 = (g^{\mu\nu}\epsilon_{\mu\nu})^2 = (g^{\mu\nu}(\nabla_\mu u_\nu + \nabla_\nu u_\mu))^2 = ((\nabla_\mu u^\mu + \nabla_\mu u^\mu))^2 = 4\,\overbrace{(\nabla_\mu u^\mu)^2}^{A} \tag{2.1.18}$$

These calculations allow us to rewrite the kinetic term as

$$\boxed{\chi = -\frac{1}{4}\epsilon_{\mu\nu}\epsilon^{\mu\nu} + \frac{1}{3}\left(\epsilon^{\mu}_{\mu}\right)^2,} \tag{2.1.19}$$

which is now entirely in terms of the physically relevant quantity of strain rather than bare derivatives. It is therefore clear that this is totally equivalent to using (2.1.12) with coefficients:

$$\zeta = \frac{1}{3}, \quad \iota = -\frac{1}{4}. \tag{2.1.20}$$

It is useful to point out that, in order to achieve a Lagrangian describing a physical theory which depends on the quantity of strain, as required for the original work in [21], the parameters are indeed restricted to $\beta = \gamma$. There is no reason to introduce the antisymmetric component $\xi \left(\epsilon^{\mu\nu\sigma\rho}\epsilon_{\mu\nu}\epsilon_{\sigma\rho}\right)$, as was done by Lim and Wang in [28] and [29], since this will always vanish in the Lagrangian. This reflects an important property that the chosen kinetic term imposes on the theory[7]:

> The framework of CEG can be fully described by the symmetric strain tensor $\epsilon_{\mu\nu}$. Anti-symmetric combinations of the field derivatives can be ignored.

As explained in [26], the choice of the coefficients is different from what one would obtain from a field describing gauge bosons, the prime example of which is the EM field. In the latter case, the Lagrangian for the free field is given as:

$$L_{EM} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} \tag{2.1.21}$$

Here it is important to note the main difference between the postulated imposter field and the EM field. For the former the relevant physical quantity is given by a fully symmetric tensor $\epsilon_{\mu\nu}$, whereas, for the latter, the quantity of interest is fully antisymmetric, and is defined as:

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu , \tag{2.1.22}$$

where the 4-vector A_μ is the EM 4-potential (for the generalisation to curved spacetime partial derivatives are replaced with covariant ones). One then finds from (2.1.21):

$$L_{EM} = -\frac{1}{4}\left(\partial_\mu A_\nu - \partial_\nu A_\mu\right)\left(\partial^\mu A^\nu - \partial^\nu A^\mu\right) = \tag{2.1.23}$$

$$-\frac{1}{4}\left[\left(\partial_\mu A_\nu\right)\left(\partial^\mu A^\nu\right) - \left(\partial_\mu A_\nu\right)\left(\partial^\nu A^\mu\right) - \left(\partial_\nu A_\mu\right)\left(\partial^\mu A^\nu\right) - \left(\partial_\nu A_\mu\right)\left(\partial^\nu A^\mu\right)\right]. \tag{2.1.24}$$

This can be simplified by switching the indices $\mu \leftrightarrow \nu$ in the first two terms, giving:

$$L_{EM} = -\frac{1}{2}\left[\left(\partial_\mu A_\nu\right)\left(\partial^\mu A^\nu\right) - \left(\partial_\mu A_\nu\right)\left(\partial^\nu A^\mu\right)\right]. \tag{2.1.25}$$

Hence, by direct comparison with (2.1.14), it can be seen that for a gauge boson (the photon in this example) the correct choice of parameters would be proportional to:

$$\alpha = 0, \quad \beta = -\gamma = -\frac{1}{2}. \tag{2.1.26}$$

One could have arrived directly at the conclusion that the imposter field would not behave as a gauge boson by observing that the vanishing of the last term of (2.1.12) for the symmetric field $\epsilon_{\mu\nu}$ would not have occurred for the antisymmetric field tensor $F_{\mu\nu}$. Therefore, both the approach taken by Hossenfelder and the one proposed in this work lead in the same direction as per the choice of the coefficients in the kinetic term χ. For now the numerical value of the coefficients has been chosen to match Hossenfelder's, in order to retrieve the MOND equation, as will be shown throughout the rest of the chapter.

[7] This is a rather heavy constraint, and it will be shown in the next chapter how it makes the theory inconsistent with observations.

2.1.2. The Kinetic Term

Classically, the Lagrangian for a field contains three components, each describing a different energy contribution: **kinetic energy**, **potential energy**, and **gradient energy**[8]. The approach taken by Hossenfelder for the kinetic term will now be explained.

To introduce a kinetic term in the Lagrangian, one looks at the Euler Lagrange equations in a curved background, which amounts to making use of the appropriate metric tensor, and changing partial derivative to covariant ones:

$$\frac{\partial L}{\partial u^\mu} = \partial_\nu \left(\frac{\partial L}{\partial \left(\partial_\nu u^\mu \right)} \right) \rightarrow \frac{\partial L}{\partial u^\mu} = \nabla_\nu \left(\frac{\partial L}{\partial \left(\nabla_\nu u^\mu \right)} \right) . \tag{2.1.27}$$

By direct comparison with (2.1.6), with the chosen definition of $\tilde{\epsilon}^2 = \chi$ from (2.1.11), the RHS of (2.1.27) is recognised as:

$$\nabla_\nu \left(\frac{\partial L}{\partial \left(\nabla_\nu u^\mu \right)} \right) \propto \nabla_\nu \left(\nabla_\rho u^\sigma \right)^2 . \tag{2.1.28}$$

This corresponds to:

$$\frac{\partial L}{\partial \left(\nabla_\nu u^\mu \right)} \propto \delta^\nu_\mu \left(\nabla_\rho u^\sigma \right)^2 , \tag{2.1.29}$$

where the Kronecker delta function was introduced, defined as:

$$\delta^\nu_\mu = \begin{cases} 1 & \mu = \nu \\ 0 & \mu \neq \nu . \end{cases}$$

It can then be noted that for (2.1.29) to hold, the Lagrangian for the free field has to have the form:

$$\boxed{L_\theta \propto \left(\nabla_\mu u^\nu \right)^3 = \chi^{3/2}.} \tag{2.1.30}$$

The following should be highlighted:

The Lagrangian for CEG utilises a term containing a power of 3/2 for the squared derivative of the field. This matches the structure of the nonrelativistic MOND Lagrangian described in section 1.7.4 exactly. Consequently, the retrieval of MOND-like behaviour in the nonrelativistic regime should not come as a surprise.

[8] For a particle, we have only the kinetic and potential components.

2.2. The Effective Metric

For the imposter field to have an impact on the gravitational interactions in a relativistic setting, one can directly define its relationship with the metric tensor[9]. In [21], Verlinde gives the following relation:

$$h_{ij} = \delta_{ij} - \frac{a_0}{c^2}\left(u_i n_j + u_j n_i\right). \tag{2.2.31}$$

It should be noted that this relation only defines the spatial component of the metric. The RHS of (2.2.31) is fully symmetric under $i \to j$ since $\delta_{ij} = \delta_{ji}$, and u_i only differs from n_i in magnitude:

$$u_i = \frac{\phi}{a_0} n_i. \tag{2.2.32}$$

This gives the equality:

$$u_i n_j = \frac{\phi}{a_0} n_i n_j = u_j n_i = \frac{\phi}{a_0} n_j n_i. \tag{2.2.33}$$

Starting from (2.2.31) one should arrive at the expression:

$$h_{ij} = \delta_{ij} - 2\frac{u_i n_j}{L}. \tag{2.2.34}$$

On the other hand, Hossenfelder defines the effective spatial metric $\tilde{h}_{ij}$ as:

$$\boxed{\tilde{h}_{ij} = g_{ij} - \frac{u_i n_j}{L}.} \tag{2.2.35}$$

The relationship $1/L = a_0/c^2$ was used and a general spatial metric g_{ij} was chosen instead of a Euclidean one.[10] While there is no issue with the substitution $\delta_{ij} \to k_{ij}$, the following must be observed:

There is a factor of 2 that is erroneously not accounted for in the effective metric $\tilde{h}_{ij}$ used by Hossenfelder. The result is that the perturbation caused by the imposter field on the spatial metric is halved.

However, in the next section, eq. (2.2.35) is used to reproduce the calculations in [26], after which observations will be made on the results. To arrive at the equations of motion for u^α, the following important assumption is made: since the imposter field is responsible

[9] In reality, this is not the only approach to introduce additional fields into the framework of GR. Moreover, theories with multiple metric tensors have also been experimentally disproved. A more thorough discussion is given in the next chapter.

[10] The Euclidean metric corresponds to an nxn identity matrix δ_{ij}, corresponding to an n-dimensional flat space.

for the modification of gravity felt by massive particles, the coupling of u^α to the energy-momentum tensor of matter should be governed by the effective metric $\tilde{h}_{\mu\nu}$ from (2.2.35), rather than by $h_{\mu\nu}$. This leads directly to the interaction term.[11]:

$$\boxed{L_{int} = -\frac{u^\mu u^\nu}{Lu} T_{\mu\nu}.} \tag{2.2.36}$$

The above is a direct generalisation of the spatial metric to include the 0^{th} component, and lacks a factor of 2 as in the case of eq. 2.2.35. This approach cannot be justified, as the metric shown in (2.2.35) is itself the spatial component of a full spacetime metric given in [21], and does not match the final form given by Hossenfelder:

The effective metric chosen for interaction with matter does not match the spacetime metric described in Verlinde's EG.

$T_{\mu\nu}$ is the energy-momentum tensor of normal matter, defined in [26] as:

$$T_{\mu\nu} = L_m g_{\mu\nu} - 2\frac{\delta L_m}{\delta g^{\mu\nu}}. \tag{2.2.37}$$

L_m is the matter Lagrangian, the form of which is unimportant for now. On the contrary, it should be mentioned that:

The definition of $T_{\mu\nu}$ in [26] contains a mistake, as the functional derivative is used on the Lagrangian, which is a function rather than a functional.

The correct relation can be given through the matter action $S_m = \int L_m \, \mathrm{d}t$, with L_m the Lagrangian for the matter field [6]:

$$\boxed{T_{\mu\nu} = -2\frac{1}{\sqrt{-g}}\frac{\delta S_m}{\delta g^{\mu\nu}}.} \tag{2.2.38}$$

If one wants to use the Lagrangian rather than the action, $T_{\mu\nu}$ can be found using Noether's theorem as (see, for example, [30]):

$$T_{\mu\nu} = g_{\mu\nu} L_m - \frac{\partial L_m}{\partial\left(\partial_\mu A^\alpha\right)}\partial_\nu A^\alpha. \tag{2.2.39}$$

In the above expression, A^μ is the field appearing in the Lagrangian L_m.[12] It is important to point out that the Noether formulation is conserved, but not necessarily symmetric,

[11] It can be shown that the addition of this term is equivalent to replacing the spacetime metric with an effective metric in the action for the source term in the Einstein Hilbert action.

[12] The same approach could be utilised for a scalar field.

unlike (2.2.38). As observed in [28], the addition of L_{int} directly modifies the geodesics of massive particles, which are now obtained from the new action[13]:

$$S_m = \frac{m}{2}\int \mathrm{d}\tau g_{\mu\nu}\dot{x}^{\mu}\dot{x}^{\nu} \rightarrow S_m + S_{int} = \frac{m}{2}\int \mathrm{d}\tau \left(g_{\mu\nu} - \frac{u_{\mu}u_{\nu}}{Lu}\right)\dot{x}^{\mu}\dot{x}^{\nu}, \tag{2.2.40}$$

where $\dot{x}$ denotes $\frac{dx}{d\tau}$, and τ is the proper time. The statement that two non-conformally[14] related metrics are present in the theory directly implies that massive particles move along geodesics that are different from those along which purely gravitational phenomena propagate, e.g. gravitational waves. A simple but powerful statement for conformally related metrics is the following[15]:

The null geodesics of a given metric $g_{\mu\nu}$ are unchanged by a conformal transformation producing the metric $f(\phi)\, g_{\mu\nu}$, with f an arbitrary function of a chosen field ϕ.

To show this, one can simply look at the condition for a zero spacetime interval in both metrics, corresponding to a lightlike connection between two points:

$$\boxed{\mathrm{d}s^2 = g_{\mu\nu}\frac{\mathrm{d}x^{\mu}}{\mathrm{d}\lambda}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\lambda} = 0 \implies f(\phi)\, g_{\mu\nu}\frac{\mathrm{d}x^{\mu}}{\mathrm{d}\lambda}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\lambda} = 0 \;\; \forall \;\; f, \phi.} \tag{2.2.41}$$

Conversely, a non-zero spacetime interval will scale by a factor of $f(\phi)$, implying that the timelike (and spacelike) geodesics are affected by the transformation, and hence massive particles travel along geodesics of the effective metric, and not the background metric. By analysing the form of the effective metric introduced, one can conclude the following:

In CEG, the background metric $g_{\mu\nu}$ and effective metric $\tilde{g}_{\mu\nu}$ are not conformally related.

This will have important repercussions for the experimental verification of the theory. In particular, since photons contribute to the total energy momentum tensor $T_{\mu\nu}$, it is not immediately clear which metric should describes the geodesics along which they travel in CEG. In chapter 3, we explore the two possible options, namely, that their geodesics are defined by the background or effective metric. We show that the former case leads to problems in the definition of event horizons, while the latter is incompatible with recent observations of the concurrent detection of gravitational and EM waves from the same source.

[13] When analysing timelike paths one can set $\lambda \rightarrow \tau$, where τ is the proper time and $\dot{x^{\mu}}$ indicates a derivative of x^{μ} w.r.t. τ.

[14] Essentially, two metrics are conformally related to each other if angles are preserved when one is mapped onto the other.

[15] The statement uses a scalar field, but the reasoning can be expanded to a function of any tensor field, provided that the two metrics are proportional to each other.

2.3. Imposter Field EOM

As the Lagrangian terms for the background metric and normal matter do not include the imposter field explicitly, it is now possible to calculate the equations of motion for u^α following (2.1.27):

$$\frac{\partial L}{\partial u^\sigma} = \nabla_\mu \left(\frac{\partial L}{\partial \left(\nabla_\mu u^\sigma\right)} \right). \tag{2.3.42}$$

The total Lagrangian for the vector field is composed of the two components L_{int} and L_θ, which describe the interaction of the field with matter and the dynamics of the field itself respectively:

$$\boxed{L = L_{int} + L_\theta = -\frac{u^\mu u^\nu}{Lu} T_{\mu\nu} + \frac{{m_p}^2}{L^2}\chi^{3/2} - \frac{\lambda^2 {m_p}^2}{L^4} u_\mu u^\mu}\,. \tag{2.3.43}$$

The LHS of (2.3.42) yields:

$$\frac{\partial}{\partial u^\sigma}\left(\overbrace{-\frac{u^\mu u^\nu}{Lu} T_{\mu\nu}}^{A} \overbrace{-\frac{\lambda^2 {m_p}^2}{L^4} u_\mu u^\mu}^{B} \right)$$

$$\frac{\partial A}{\partial u^\sigma} = \left(\frac{-\delta^\mu_\sigma u^\nu - \delta^\nu_\sigma u^\mu}{Lu} + \frac{u^\mu u^\nu u^\sigma}{Lu^3} \right) T_{\mu\nu} = \frac{1}{L}\left(-n^\nu T_{\sigma\nu} - n^\mu T_{\mu\sigma} + n^\mu n^\nu n_\sigma T_{\mu\nu}\right).$$

The assumption was made that $T_{\mu\nu}$ is fully symmetric (which always holds if it is defined through (2.2.37)) and $n^\mu = u^\mu/u$ was used. Thus, one obtains:

$$\begin{aligned} \frac{\partial A}{\partial u^\sigma} &= \frac{n^\mu}{L}\left(-2\delta^\nu_\sigma + n^\nu n_\sigma\right) T_{\mu\nu} \\ \frac{\partial (A+B)}{\partial u^\sigma} &= \frac{n^\mu}{L}\left(-2\delta^\nu_\sigma + n^\nu n_\sigma\right) T_{\mu\nu} - \frac{2\lambda^2 {m_p}^2}{L^4} u_\sigma . \end{aligned}$$

For the RHS of (2.3.42), the only term contributing is χ, which gives:

$$\frac{\partial \chi^{3/2}}{\partial\left(\nabla_\mu u^\sigma\right)} = \frac{3}{2}\chi^{1/2} \frac{\partial \chi}{\partial\left(\nabla_\mu u^\sigma\right)} = \frac{3}{2}\chi^{1/2} \frac{\partial}{\partial {u^\sigma}_{;\mu}} \left(\overbrace{-\frac{1}{4}\epsilon_{\nu\rho}\epsilon^{\nu\rho}}^{C} \overbrace{+\frac{1}{3}\epsilon^2}^{D} \right). \tag{2.3.44}$$

To avoid clutter, the standard notation was introduced:

$$\partial_\mu u^\sigma = {u^\sigma}_{,\mu} \tag{2.3.45}$$

$$\nabla_\mu u^\sigma = {u^\sigma}_{;\mu}\,. \tag{2.3.46}$$

Carrying out the differentiation:

$$\frac{\partial C}{\partial u^{\sigma}{}_{;\mu}} = -\frac{1}{4}\left[\left(\frac{\partial \epsilon_{\nu\rho}}{\partial u^{\sigma}{}_{;\mu}}\epsilon^{\nu\rho}\right) + \left(\epsilon_{\nu\rho}\frac{\partial \epsilon^{\nu\rho}}{\partial u^{\sigma}{}_{;\mu}}\right)\right] = \tag{2.3.47}$$

$$-\frac{1}{4}\left[\left(\delta^{\mu}_{\nu}\delta_{\rho\sigma} + \delta^{\mu}_{\rho}\delta_{\sigma\nu}\right)\epsilon^{\nu\rho} + \epsilon_{\nu\rho}\left(\delta^{\rho}_{\sigma}\delta^{\nu\mu} + \delta^{\mu\rho}\delta^{\nu}_{\sigma}\right)\right] = -\epsilon^{\mu}_{\sigma} \tag{2.3.48}$$

$$\frac{\partial D}{\partial u^{\sigma}{}_{;\mu}} = \frac{1}{3}\frac{\partial \epsilon^{2}}{\partial u^{\sigma}{}_{;\mu}} = \frac{8}{3}\nabla_{\alpha}u^{\alpha}\delta^{\mu}_{\sigma} = \frac{4}{3}\epsilon\delta^{\mu}_{\sigma}. \tag{2.3.49}$$

The following differential identity was used:

$$\frac{\partial u^{\mu}}{\partial u^{\nu}} = \delta^{\mu}_{\nu}. \tag{2.3.50}$$

In addition, the definition of ϵ was utilised:

$$\epsilon^{\mu}_{\mu} = 2\nabla_{\mu}u^{\mu}. \tag{2.3.51}$$

Plugging these results into (2.3.42), the four equations of motion are obtained as:

$$\frac{3m_p^2}{2L^2}\nabla_{\mu}\left[\chi^{1/2}\left(-\epsilon^{\mu}_{\sigma} + \frac{4}{3}\epsilon\delta^{\mu}_{\sigma}\right)\right] = \frac{n^{\mu}}{L}\left(-2\delta^{\nu}_{\sigma} + n^{\nu}n_{\sigma}\right)T_{\mu\nu} - \frac{2\lambda^2 {m_p}^2}{L^4}u_{\sigma}. \tag{2.3.52}$$

Multiplying both sides by $-L$, one finally arrives at a form comparable to that given in [26]:

$$\boxed{-\frac{3m_p^2}{2L}\nabla_{\mu}\left[\chi^{1/2}\left(-\epsilon^{\mu}_{\sigma} + \frac{4}{3}\epsilon\delta^{\mu}_{\sigma}\right)\right] = n^{\mu}\left(2\delta^{\nu}_{\sigma} - n^{\nu}n_{\sigma}\right)T_{\mu\nu} + \frac{2\lambda^2 {m_p}^2}{L^3}u_{\sigma}}. \tag{2.3.53}$$

However, it must be noted that there is a difference in sign for the second term on the RHS between the result obtained here and that of [26]. This is encouraging, as it is noted in [31] that a mistake is present in [26], in the calculation of the energy-momentum tensor in a de Sitter background, and the way to avoid it is, in fact, to assume that the imposter Lagrangian is incorrect, and should be rewritten as:

$$L_{\theta} = \frac{{m_p}^2}{L^2}\chi^{3/2} + \frac{\lambda^2 {m_p}^2}{L^4}\left(u_{\mu}u^{\mu}\right)^2. \tag{2.3.54}$$

Neglecting the newly introduced power of 2 for the inner product $u_{\alpha}u^{\alpha}$, the modification would hence lead to the equations of motion initially proposed in [26], namely, the second term on the RHS would become:

$$\frac{2\lambda^2 {m_p}^2}{L^3}u_{\sigma} \to -\frac{2\lambda^2 {m_p}^2}{L^3}u_{\sigma}. \tag{2.3.55}$$

Through the above substitution, the EOM shown in eq. (9) of [26] are recovered.

The observation put forward by [31] to modify the Lagrangian has been motivated through the derivation presented above. The latter has been carried out independently.

However, the correction has no relevance for the connection to the MOND equation since the mass of the imposter field λ is set to zero, as explained in the next section.

2.4. Nonrelativistic Limit

2.4.1. Assumptions

Following [26], five assumptions are made to obtain a non-relativistic limit to make contact with MOND:

1. The mass of the imposter field u^α can be entirely neglected, so that $\lambda = 0$. In [26], the reason is given a posteriori, after the non-relativistic gravitational potential has been obtained. It states that the approximation used holds for $L^2 \gg r^2$, with r the distance between the gravitationally attracted bodies. This certainly holds for galaxies and clusters, the scale at which the approximation is aimed.

2. The reference frame is chosen to be close to the rest frame of u^α, so that the spatial component of the unit 4-vector in the direction of u^α is vanishing:

$$\boxed{n^\alpha = \frac{u^\alpha}{u} \approx \left(-1, \vec{0}\right)} \tag{2.4.56}$$

It is important to note the vagueness of the definition of this quantity in [26], which is not given in tensorial form but written as **n**, with no indication of whether it is the vector or covector form of n^α. Since the metric tensor used is of the form $-+++$, there is, in fact, a factor of -1 which is not specified. This will lead, as shown in this section, to an ambiguity in the calculation. It should be noted that by using the definition given in [26], one obtains:

$$\boxed{\phi = \frac{\sqrt{-u_\alpha u^\alpha}}{L} = \frac{u}{L}}. \tag{2.4.57}$$

Alongside (2.4.56), this implies:

$$\boxed{\frac{u^0}{u} = \frac{u^0}{\phi L} = -1 \to u^0 = -\phi L} \tag{2.4.58}$$

3. The solution is both spherically symmetric and time independent.

4. The energy momentum tensor is assumed to correspond to a pressureless fluid, characterised by $T_{i\mu} = 0$ and $T_{00} = \rho$.[16]

5. The background is assumed to be that of flat spacetime, so covariant derivatives are replaced by partial derivatives.

Although these assumptions seem reasonable, the last one is, in fact, not strictly allowed or correct. Replacing covariant derivatives by partial derivatives when working on flat spacetime is only allowed if working with Cartesian coordinates, as the Christoffels then vanish. However, generally:

> The replacement of covariant derivatives by partial derivatives can only occur in Cartesian coordinates for flat spacetime. This procedure is not allowed if working in spherical coordinates, as in this case. Consequently, the calculations contain a serious mistake.

2.4.2. RHS: The Source Term

First, the RHS of (2.3.53) is worked out, as it contains no derivative terms. By applying $T_{i\mu} = 0$ and taking the $\lambda \to 0$ limit, one obtains:

$$n^0 \left(2\delta^0_\sigma - n^0 n_\sigma\right) T_{00} = \tag{2.4.59}$$

$$\boxed{-\left(2\delta^0_\sigma + n_\sigma\right)\rho = \begin{cases} -(2+1)\rho = -3\rho & \sigma = 0 \\ 0 & \sigma \neq 0 \end{cases}} \tag{2.4.60}$$

where the assumption was made that definition (2.4.56) had the correct index placement. It can now be seen that the answer differs by a factor of 3 from the one obtained in [26]. Instead, if one assumes that the definition (2.4.56) is for the covector n_α, the RHS simplifies instead to 3ρ, which would result in a negative mass density. It is thus clear that the vector form is the correct one. Nonetheless:

> The mass density obtained in [26] is wrong by a factor of 3. This will directly impact the inferred value of a_0 when deriving the PDE for the MOND potential ϕ.

[16] Pressureless fluids are commonly used in astrophysics and cosmology to represent models of particles that interact exclusively through gravity. They can give a very good approximation of the structure of galaxies and clusters.

2.4.3. LHS: The MOND Term

Before proceeding with the simplification of the LHS involving differentiation, it is necessary to return to the claim that for flat spacetime one can simply convert covariant derivatives to partial derivatives. It is true that Minkowski spacetime is, in fact, flat, and its description in Cartesian coordinates does not produce any non-zero Christoffel symbols. However, in order to describe a spherically symmetric system, considering only differentiation with respect to the radial coordinate, spherical coordinates must be used. Although the spacetime described is the same, there are now non-vanishing Christoffel symbols, which are listed here for convenience (as can be found, for example, in [32]):

$$\Gamma^r_{\theta\theta} = -r, \quad \Gamma^r_{\phi\phi} = -r\sin^2(\theta), \quad \Gamma^\theta_{r\theta} = \frac{1}{r} \tag{2.4.61}$$

$$\Gamma^\theta_{\phi\phi} = -\sin(\theta)\cos(\theta), \quad \Gamma^\phi_{r\phi} = \frac{1}{r}, \quad \Gamma^\phi_{\theta\phi} = \cot(\theta). \tag{2.4.62}$$

When looking at the LHS of (2.3.53), ones needs to consider whether any of the Christoffel symbols from (2.4.61) are non vanishing. If this is the case, the covariant derivative does not match the partial derivative. Analysing, term by term, the LHS of (2.3.53) one gets:

$$\nabla_\alpha u^\beta = \partial_\alpha u^\beta + \Gamma^\beta_{\alpha\lambda} u^\lambda = \partial_\alpha u^\beta + \Gamma^\beta_{\alpha 0} u^0 = \partial_\alpha u^\beta. \tag{2.4.63}$$

The definition $n^i = 0$ was used, and it was noted from (2.4.61) that there are no non vanishing Christoffel symbols for the 0^{th} index t. One finds similar results for the covariant divergence:

$$\nabla_\mu u^\mu = \partial_\mu u^\mu + \Gamma^\mu_{\mu\lambda} u^\lambda = \partial_\mu u^\mu + \Gamma^\mu_{\mu 0} u^0 = \partial_\mu u^\mu. \tag{2.4.64}$$

The same can be said for the covariant derivative of the covector:

$$\nabla_\alpha u_\beta = \partial_\alpha u_\beta - \Gamma^\lambda_{\alpha\beta} u_\lambda = \partial_\alpha u_\beta. \tag{2.4.65}$$

Moreover, by definition the covariant derivative reduces to a partial derivative when applied to a scalar function[17], so one can directly conclude:

$$\nabla_\mu \epsilon = \partial_\mu \epsilon, \quad \nabla_\mu \chi = \partial_\mu \chi. \tag{2.4.66}$$

The assumption made by Hossenfelder about utilising partial derivatives has proved valid up until now. However, this is merely a coincidence:

> The factor that allows the use of partial derivatives up to this point is the chosen reference frame, in which the spatial components of u^a vanish, as do any relevant Christoffels.

[17] This notion is intuitive, as scalars do not possess a direction, and transporting them between two points in a curved spacetime will have no impact on their value.

If this assumption is not maintained, or the spherical symmetry is dropped, one generally has to use covariant derivatives. By utilising what was found in (2.4.63)-(2.4.66), one can explicitly evaluate the kinetic term, starting from the definition of χ:

$$\chi = \frac{4}{3}(\nabla_\nu u^\nu)(\nabla_\mu u^\mu) - \frac{1}{2}(\nabla_\mu u_\nu)(\nabla^\mu u^\nu) - \frac{1}{2}(\nabla_\mu u_\nu)(\nabla^\nu u^\mu) = \tag{2.4.67}$$

$$\frac{4}{3}(\partial_\nu u^\nu)(\partial_\mu u^\mu) - \frac{1}{2}(\partial_\mu u_\nu)(\partial^\mu u^\nu) - \frac{1}{2}(\partial_\mu u_\nu)(\partial^\nu u^\mu). \tag{2.4.68}$$

Noting that the only combination giving a non-zero result is $\partial_r u^0$, which cannot be achieved in the first and third term, the above simplifies to:

$$\boxed{\chi = -\frac{1}{2}(\partial_\mu u_\nu)(\partial^\mu u^\nu) = -\frac{1}{2}(\partial_r u_0)(\partial^r u^0) = \frac{1}{2}\left(\partial_r u^0\right)^2.} \tag{2.4.69}$$

To obtain $\partial^r = \partial_r$ and $u^0 = -u_0$ in the last expression, the metric for Minkowski spacetime in spherical coordinates, which has the following spacetime interval, was used (it is important to note the difference between the angular variable φ and the potential ϕ):

$$\mathrm{d}s^2 = -\mathrm{d}t^2 + \mathrm{d}r^2 + r^2\mathrm{d}\theta + r^2\sin^2(\theta)\,\mathrm{d}\varphi. \tag{2.4.70}$$

It is now opportune to pause and analyse the form of the kinetic term in this regime. It is easy to see that for a static, spherically symmetric scalar potential $\phi(r)$ the following always holds:

$$\nabla_\mu(\phi) = (0, \partial_r\phi, 0, 0) \implies (\nabla_\mu\phi)^2 = (\nabla\phi)^2 = (\partial_r\phi)^2 \tag{2.4.71}$$

After re-writing $u_0 = -\phi/a_0$, it can then be seen that χ corresponds to the argument of the function $\mathcal{F}(u)$ in the nonrelativistic Lagrangian for MOND given in (1.7.64), up to a factor of 1/2.

The power 3/2 for χ in the CEG Lagrangian is in direct analogy with the function $\mathcal{F}(u)$ for the deep MOND limit. The current EOM will inevitably generate a potential proportional to that of deep MOND.

Proceeding with the calculation, one gets:

$$\nabla_\mu\chi^{1/2} = \partial_\mu\chi^{1/2} = \frac{1}{2}\chi^{-1/2}\partial_\mu\chi = \frac{1}{4}\chi^{-1/2}\partial_\mu\left(\partial_r u^0\right)^2. \tag{2.4.72}$$

As explained for (2.4.68), $\epsilon = 2\partial_\mu u^\mu = 0$, yielding an RHS of the form:

$$-\frac{3m_p^2}{2L}\nabla_\mu\left[\sqrt{\chi}\left(-\epsilon^\mu_\sigma\right)\right] = \frac{3m_p^2}{2L}\left(\sqrt{\chi}\nabla_\mu\epsilon^\mu_\sigma + \epsilon^\mu_\sigma\partial_\mu\sqrt{\chi}\right). \tag{2.4.73}$$

The reason why the covariant derivative was kept on the scalar term will soon become apparent. The equation given in [26] is for the $\sigma = 0$ term, which is the only nonzero component of (2.4.59), and is calculated as:

$$\frac{3m_p^2}{2L}\left(\overbrace{\sqrt{\chi}\nabla_\mu\epsilon_0^\mu}^{A}+\overbrace{\epsilon_0^\mu\partial_\mu\sqrt{\chi}}^{B}\right) \tag{2.4.74}$$

$$A=\sqrt{\chi}\nabla_\mu\left(\nabla_0 u^\mu+\nabla^\mu u_0\right)=\sqrt{\chi}\left(\nabla_\mu\overbrace{\nabla_0 u^\mu}^{=0}+\nabla_\mu\nabla^\mu u_0\right). \tag{2.4.75}$$

It is useful to introduce the expression for the Laplace-Beltrami operator,[18] defined as:

$$\nabla_\mu\nabla^\mu f=\frac{1}{\sqrt{|g|}}\frac{\partial}{\partial x^\mu}\left(g^{\mu\nu}\sqrt{|g|}\frac{\partial f}{\partial x^\nu}\right). \tag{2.4.76}$$

In the above expression, $|g|$ is the magnitude of the determinant of the metric tensor, which for Minkowski space in spherical coordinates is given by:

$$|g|=\left|\prod_i g_{ii}\right|=\left|r^4\sin^2(\theta)\right|. \tag{2.4.77}$$

We then arrive at:

$$A=\sqrt{\chi}\frac{1}{r^2\sin(\theta)}\frac{\partial}{\partial x^\mu}\left(g^{\mu\nu}r^2\sin(\theta)\frac{\partial u_0}{\partial x^\nu}\right)= \tag{2.4.78}$$

$$-\sqrt{\chi}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u^0}{\partial r}\right). \tag{2.4.79}$$

The only non vanishing derivative is the radial one, so $\sin(\theta)$ cancels out. Furthermore, $g^{rr}=1$ was used, while the minus sign was picked up by raising $u_0=-u^0$. At this point it is important to note that, using partial rather than covariant derivatives in (2.4.74), the result would be considerably different, since the Laplace-Beltrami operator could not be utilised:

$$A=-\sqrt{\chi}\frac{\partial}{\partial r}\left(\frac{\partial u^0}{\partial r}\right)=-\sqrt{\chi}\frac{\partial^2 u^0}{\partial r^2}. \tag{2.4.80}$$

> As expected, the use of partial in place of covariant derivatives results in a different, incorrect, set of EOMs.

The second term from (2.4.74) is expanded as:

$$B=\epsilon_0^\mu\partial_\mu\sqrt{\chi}=\frac{1}{2}\epsilon_0^\mu\chi^{-1/2}\partial_\mu\chi=\frac{1}{4}\frac{\sqrt{2}}{|\partial_r u^0|}\nabla^r u_0\partial_r\left(\partial_r u^0\right)^2= \tag{2.4.81}$$

$$-\frac{1}{4}\frac{\sqrt{2}}{|\partial_r u^0|}\partial_r u^0\partial_r\left(\partial_r u^0\right)^2. \tag{2.4.82}$$

[18] The Laplace-Beltrami operator is the equivalent of the Laplacian for curved space or curvilinear coordinates. In fact, one can derive the Laplacian for flat space in curvilinear coordinates through this operator.

The following was used:

$$\nabla^r u_0 = \partial^r u_0 = -\partial_r u^0. \tag{2.4.83}$$

It is now possible to evaluate the full expression:

$$A + B = \sqrt{\chi}\nabla_\mu \epsilon_0^\mu + \epsilon_0^\mu \partial_\mu \sqrt{\chi} = \tag{2.4.84}$$

$$\sqrt{\frac{1}{2}}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u^0}{\partial r}\right) + \frac{1}{4}\frac{\partial u^0}{\partial r}\frac{\partial}{\partial r}\left(\frac{\partial u^0}{\partial r}\right)^2 \frac{\sqrt{2}}{\partial_r u^0} = \tag{2.4.85}$$

$$\sqrt{\frac{1}{2}}\left|\frac{\partial u^0}{\partial r}\right|\frac{\partial^2 u^0}{\partial r^2} + \sqrt{2}\frac{1}{r}\left|\frac{\partial u^0}{\partial r}\right|\frac{\partial u^0}{\partial r} + \sqrt{\frac{1}{2}}\frac{\left(\partial_r u^0\right)^2}{\left|\partial_r u^0\right|}\frac{\partial^2 u^0}{\partial r^2} = \tag{2.4.86}$$

$$\left|\frac{\partial u^0}{\partial r}\right|\left(\sqrt{\frac{1}{2}}\frac{\partial^2 u^0}{\partial r^2} + \sqrt{2}\frac{1}{r}\frac{\partial}{\partial r}u^0 + \sqrt{\frac{1}{2}}\frac{\partial^2 u^0}{\partial r^2}\right) = \tag{2.4.87}$$

$$\sqrt{2}\left|\frac{\partial u^0}{\partial r}\right|\left(\frac{\partial^2 u^0}{\partial r^2} + \frac{1}{r}\frac{\partial u^0}{\partial r}\right). \tag{2.4.88}$$

The equivalent notations ∂_r and $\frac{\partial}{\partial r}$ were each used where more convenient to avoid clutter. To make contact with MOND, one has to rewrite the following expression:

$$\sqrt{\frac{1}{2}}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u^0}{\partial r}\left|\frac{\partial u^0}{\partial r}\right|\right) = \tag{2.4.89}$$

$$\sqrt{\frac{1}{2}}\left(\frac{2}{r}\frac{\partial u^0}{\partial r}\left|\frac{\partial u^0}{\partial r}\right| + \frac{\partial^2 u^0}{\partial r^2}\left|\frac{\partial u^0}{\partial r}\right| + \frac{\partial u^0}{\partial r}\frac{\partial}{\partial r}\left|\frac{\partial u^0}{\partial r}\right|\right). \tag{2.4.90}$$

From the derivative of the absolute value:

$$\frac{d\left|x\right|}{dx} = \frac{x}{\left|x\right|} = \frac{\left|x\right|}{x}, \tag{2.4.91}$$

one retrieves the same expression as eq. 2.4.88:

$$\boxed{\sqrt{2}\left|\frac{\partial u^0}{\partial r}\right|\left(\frac{\partial^2 u^0}{\partial r^2} + \frac{1}{r}\frac{\partial u^0}{\partial r}\right).} \tag{2.4.92}$$

In order to obtain the exact form expressed by MOND, the Laplace-Beltrami operator is recognised in (2.4.89):

$$\sqrt{\frac{1}{2}}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial u^0}{\partial r}\left|\frac{\partial u^0}{\partial r}\right|\right) = \sqrt{\frac{1}{2}}\overbrace{\nabla_\mu}^{\nabla\cdot}(\overbrace{\left|\nabla^\nu u^0\right|}^{\left|\nabla u^0\right|}\overbrace{\nabla^\mu u^0}^{\nabla u^0}). \tag{2.4.93}$$

This leads directly[19] to the MOND expression for u^0:

$$\frac{1}{\sqrt{2}}\nabla\cdot\left(\left|\nabla u^0\right|\nabla u^0\right) = \frac{1}{\sqrt{2}a_0}\nabla\cdot\left[\left(\frac{\left|\nabla\phi\right|}{a_0}\right)\nabla\phi\right]. \tag{2.4.94}$$

[19] The extra factor of $1/a_0$ is cancelled in the final EOM by the prefactor of the LHS.

When equating this to the RHS of the EOM, and reinstating the constant prefactors one finally obtains:

$$\boxed{-\frac{3a_0}{2}\frac{1}{\sqrt{2}a_0}\nabla\cdot\left[\left(\frac{|\nabla\phi|}{a_0}\right)\nabla\phi\right]=-3G\rho\implies\nabla\cdot\left[\left(\frac{|\nabla\phi|}{a_0}\right)\nabla\phi\right]=2\sqrt{2}G\rho}\tag{2.4.95}$$

The above certainly resembles the MOND PDE. Although the prefactors are clearly incorrect, the coefficients in the Lagrangian could be changed to obtain the correct scale.

If the case of working exclusively with partial derivatives, the same result would not have been achieved. Instead, by using (2.4.80), one would have arrived at an LHS of the form:[20]

$$\sqrt{2}\left|\partial_r u^0\right|\frac{\partial^2 u^0}{\partial r^2}=\sqrt{\frac{1}{2}}\frac{\partial}{\partial r}\left(\left|\frac{\partial u^0}{\partial r}\right|\frac{\partial u^0}{\partial r}\right)=\frac{1}{\sqrt{2}}\partial_r\left(\partial_r\phi\right)^2.\tag{2.4.96}$$

This would determine the EOM:

$$\partial_r\left(\frac{\partial_r\phi}{a_0}\right)^2=2\sqrt{2}G\rho.\tag{2.4.97}$$

In this case the correspondence with MOND from (2.4.93) could not be retrieved. By consistently using partial derivatives, Hossenfelder in fact arrives at the final expression[21]:

$$\frac{3m_p^2L}{2}\partial_r\left(\partial_r\phi\right)^2=\rho\implies\partial_r\left(\partial_r\phi\right)^2=\frac{2}{3}Ga_0\rho.\tag{2.4.98}$$

The error in the prefactor for the mass distribution is to be expected given the mentioned calculation mistake in the RHS. The real problem is that the potential stemming from the above PDE is not logarithmic:

> The consistent use of partial derivatives leads to EOM that do not yield a logarithmic potential, as is, instead, claimed by Hossenfelder.

2.4.4. The Non Spherically Symmetric Case

Having made this observation, another question arises after this procedure: is it possible to further relax the assumptions made to obtain the nonrelativistic limit, in order to make contact with the MOND equation for a general system which does not possess spherical symmetry? After all, one could argue that many of the steps made, such as only considering the radial derivative of u^0, could feasibly be generalised to the three Cartesian

[20] The absolute value in the derivative of the field can be ignored when realising that the value of the gravitational acceleration never changes sign. In the convention used, the acceleration is always positive.

[21] It should also be observed that the Planck mass m_p has the wrong definition in [26], given through $G=m_p^2$ rather than the correct $G=m_p^{-2}$

coordinates. First, we need to check whether the scalar quantity of interest, χ, has the same structure when considering all spatial derivatives in Cartesian coordinates. This amounts to evaluating each of its terms, in a similar way to what was done in (2.4.67), but for the general case:

$$\chi = \frac{4}{3}\overbrace{(\nabla_\nu u^\nu)(\nabla_\mu u^\mu)}^{=0} - \frac{1}{2}(\nabla_\mu u_\nu)(\nabla^\mu u^\nu) - \frac{1}{2}\overbrace{(\nabla_\mu u_\nu)(\nabla^\nu u^\mu)}^{=0}. \tag{2.4.99}$$

For the first term it was observed that:

$$\nabla_i u^i = \nabla_0 u^0 = 0. \tag{2.4.100}$$

Similarly, for the third term:

$$\overbrace{(\nabla_0 u_i)}^{=0}\left(\nabla^i u^0\right) = (\nabla_i u_0)\overbrace{\left(\nabla^0 u^i\right)}^{=0} = 0. \tag{2.4.101}$$

Without assuming any specific reference frame, but still requiring that spacetime is flat so that all Christoffel symbols containing t are vanishing[22], the remaining term yields:

$$\chi = -\frac{1}{2}(\nabla_\mu u_\nu)(\nabla^\mu u^\nu) = -\frac{1}{2}(\nabla_i u_0)(\nabla^i u^0) = \frac{1}{2}(\nabla_i u^0)(\nabla^i u^0). \tag{2.4.102}$$

The above is justified by noting that for any flat spacetime metric one has $u_0 = -u^0$. Proceeding further, as the static case is still implied, the time derivative vanishes and one has:

$$\nabla_\mu \chi = \frac{1}{2}\nabla_\mu\left[(\nabla_i u^0)(\nabla^i u^0)\right] \to \frac{1}{2}\nabla_j\left[(\nabla_i u^0)(\nabla^i u^0)\right]. \tag{2.4.103}$$

The other term that needs to be evaluated is:

$$\nabla_\mu \epsilon^\mu_\sigma \xrightarrow{\sigma=0} \nabla_\mu\left[\overbrace{\nabla_0 u^\mu}^{=0} + \nabla^\mu u_0\right] = \tag{2.4.104}$$

$$\nabla_\mu \nabla^\mu u_0 = \nabla_i \nabla^i u_0 = -\nabla_i \nabla^i u^0. \tag{2.4.105}$$

The final expression is the spatial Laplace-Beltrami operator or, in other words, the Laplacian in arbitrary 3D coordinates. It is now possible to re-evaluate the full expression in general flat spacetime by making use of (2.4.104) and (2.4.103), neglecting the constants for the moment, as:

$$(\sqrt{\chi}\nabla_\mu \epsilon^\mu_\sigma + \epsilon^\mu_\sigma \nabla_\mu \sqrt{\chi}) = \tag{2.4.106}$$

$$-\sqrt{\frac{1}{2}\left(\nabla_i u^0\right)\left(\nabla^i u^0\right)}\left(-\nabla_i \nabla^i u^0\right) - \left(-\nabla^j u^0\right)\frac{1}{2}\frac{1}{\sqrt{\frac{1}{2}\left(\nabla_i u^0\right)\left(\nabla^i u^0\right)}}\frac{1}{2}\overbrace{\nabla_j\left[\left(\nabla_i u^0\right)\left(\nabla^i u^0\right)\right]}^{C}. \tag{2.4.107}$$

[22] Regardless of the coordinates chosen, spacetime metrics will always have vanishing Christoffel containing the time component. This is due to the fact that for observers in the same inertial frame, the time coordinate is the same regardless of spacetime location.

So far, no assumption about a coordinate system has been made. However, to reproduce the MOND equation, we need to identify the vector calculus operators for gradient and divergence:

$$\nabla^i \to \nabla, \quad \nabla_i u^i = \nabla \cdot u^i. \tag{2.4.108}$$

This requires raising the index on expressions containing ∇_i. However, doing so implies the use of the metric tensor. Therefore, to obtain an explicit expression we have to choose a coordinate system. Without loss of generality, from now on we will assume that the coordinates used are Cartesian, where the spatial metric is the 3x3 identity matrix, g_{ij} =diag$(1,1,1)$. So in this case one simply has $\nabla_i = \nabla^i = \nabla$, the gradient operator. From (2.4.107), we then obtain:

$$C = \nabla_j \left[\left(\nabla_i u^0\right) \left(\nabla^i u^0\right) \right] = \nabla_i u^0 \nabla_j \nabla^i u^0 + \nabla^i u^0 \nabla_j \nabla_i u^0 = 2 \nabla_i u^0 \nabla_j \nabla^i u^0. \tag{2.4.109}$$

Plugging the expression obtained above back into (2.4.107) yields:

$$\boxed{\sqrt{\frac{1}{2}} \left(\left|\nabla^i u^0\right| \nabla_j \nabla^j u^0 + \frac{\nabla^j u^0 \nabla_i u^0 \nabla_j \nabla^i u^0}{\left|\nabla^k u^0\right|} \right)}\,. \tag{2.4.110}$$

Now, this does not look like the general MOND equation, but it is. This can be proven by starting from the desired result and working backwards:

$$\nabla \cdot \left(\left|\nabla u^0\right| \nabla u^0 \right) = \nabla_i \left(\left|\nabla_j u^0\right| \nabla^i u^0 \right) = \tag{2.4.111}$$

$$\left|\nabla_j u^0\right| \nabla_i \nabla^i u^0 + \nabla^i u^0 \nabla_i \left|\nabla_j u^0\right| = \tag{2.4.112}$$

$$\left|\nabla_j u^0\right| \nabla_i \nabla^i u^0 + \frac{\nabla^i u^0 \nabla_i \nabla_j u^0 \nabla^j u^0}{\left|\nabla_k u^0\right|} = \tag{2.4.113}$$

$$\left|\nabla_j u^0\right| \nabla_i \nabla^i u^0 + \frac{\nabla^j u^0 \nabla_i u^0 \nabla_j \nabla^i u^0}{\left|\nabla^k u^0\right|}. \tag{2.4.114}$$

In the third line the derivative of the absolute value was used:

$$\nabla_i \left|\nabla_j u^0\right| = \nabla_i \left|\nabla_j u^0 \nabla^j u^0\right|^{1/2} = \tag{2.4.115}$$

$$\frac{\nabla_i \nabla_j u^0 \nabla^j u^0 + \nabla_i \nabla^j u^0 \nabla_j u^0}{2 \left(\nabla_k u^0 \nabla^k u^0\right)^{1/2}} = \frac{\nabla_i \nabla^j u^0 \nabla_j u^0}{\left|\nabla^k u^0\right|}. \tag{2.4.116}$$

In the last line we used the fact that ∇_i and ∇_j commute in Cartesian coordinates.

It has been proven that, without assuming any type of symmetry, the MOND equation can be fully recovered in Cartesian coordinates in a flat spacetime. In this general case, the kinetic term is $\chi = -1/2 \left(\nabla_\mu u_\nu\right) \left(\nabla^\mu \nabla^\nu\right)$.

2.5. An Outlook on the Nonrelativistic Limit of CEG

Having reproduced the approach by Hossenfelder for the nonrelativistic limit of CEG, it is clear that there are multiple errors in the procedure, some negligible and others problematic. The inaccuracies are the following:

1. The RHS is missing a factor of 3 in front of the mass distribution. Although not disastrous, the potential resulting from a full PDE using the wrong mass distribution would imply an erroneous value of a_0 to match observations;

2. The LHS is missing a factor of $\sqrt{2}$, possibly due to the evaluation of $\sqrt{\chi}$;

3. The given definition of the energy-momentum tensor is incorrect.

Furthermore, all but one of the assumptions made to obtain the nonrelativistic limit seem flawed:

1. The normalisation of the imposter field is given as the square of the potential, which is identified in the nonrelativistic limit as a logarithm. It is claimed that the mass term vanishes for $r \ll L$, which includes all non cosmological scales. The problem is then evident: the mass term will diverge for small r, as $\lim_{r\to 0} (\ln(r))\,2 = \infty$.

   > The mass term diverges in the nonrelativistic limit for small r, and neglecting it in the derivation cannot be justified.

2. It is assumed that one can choose a frame in which the imposter field is at rest. As the theory claims to expand EG, the spatial component of u^μ must follow Verlinde's prescription, $u^i \propto \phi/a_0$. On a Minkowski background, over which the derivation is carried out, this is inconsistent with a timelike field given the normalisation chosen by Hossenfelder. This will be shown to be true in the next chapter, but the idea is the following:

   > Either the chosen normalisation for u^α is incorrect, or the field is lightlike. Both possibilities clash with the choice of a rest frame for u^α, in which the spatial components vanish.

3. It is assumed that the solution is spherically symmetric. This should, in fact, be the case if one wants to match the deep MOND limit and obtain a logarithmic potential.

   > We have shown that the MOND equation can be retrieved without assuming the potential only has radial dependence.

4. The use of partial derivatives instead of covariants is completely incorrect when working with curvilinear coordinates, such as the spherical coordinates needed to derive a spherically symmetric solution. Using partial derivatives results in a PDE whose solution is not logarithmic:

   > The use of partial derivatives is not justified and leads to an incorrect form for the potential.

Moreover, no mention is made of an acceleration scale below which one should enter the deep MOND regime. Consequently, the only possibility the theory allows for is the coexistence of Newtonian and deep MOND potential in all regimes, with one of the two terms dominating the sum depending on the distance from the source. This is in clear contrast with the main tenet of MOND, which predicts that the transition should follow from a scale of acceleration, rather than distance. The following can then be said:

> The analysis carried out for the nonrelativistic case of CEG does not correctly represent Verlinde's idea, nor does it convincingly recover MOND behaviour.

The only consistent way in which a_0 enters CEG is through the normalisation of u^μ. However, we show in chapter 3 that this normalisation leads to inconsistencies, and is once again not compatible with Verlinde's EG.

Chapter 3

Addressing the Inconsistencies in Covariant Emergent Gravity

As shown in the previous chapter, several inconsistencies and conceptual mistakes have been found in the formulation of CEG and the resulting EOM. Various problems were pointed out while introducing the theory. However, certain considerations pertaining to choices made when defining the theory are best described separately, now that the results stemming from the approach taken by Hossenfelder have been shown.

The aim of this chapter is to describe a number of alternative approaches and ideas that seem more consistent with the original idea of EG, and which also eliminate certain contradictions present in the CEG framework.

3.1. A Natural Set of Coefficients for χ

The arguments used by Hossenfelder to determine the coefficients of the kinetic term χ cannot be coherently generalised. The requirement of conservation of the energy-momentum tensor in a De Sitter background gives rise to coefficients which are either argued against, or require a modification of the given Lagrangian, as shown in [28] [29]. It hence appears that either the definition of the Lagrangian is flawed, or a more systematic approach is needed in determining the coefficients for χ. Therefore, it is useful to analyse closely the equation from [21] through which Hossenfelder builds the Lagrangian in the first place. This is eq. 7.37 from [21]:[1]

$$\left(\frac{8\pi G}{a_0}\Sigma_D\right)^2 = -\left(\frac{d-2}{d-1}\right)\nabla_i\left(\frac{\phi_B}{a_0}n_i\right). \tag{3.1.1}$$

As shown at the start of the previous chapter, this can be simplified by using eq. 7.28 of [21],

$$\Sigma_D = \frac{a_0}{8\pi G}\tilde{\epsilon}. \tag{3.1.2}$$

[1] It should be noted that for ease of comparison, the notation used for the following equations is identical to that of [21], with all indices being lowered and latin indices representing 3-component vectors.

This gives:

$$\tilde{\epsilon}^2 = -\left(\frac{d-2}{d-1}\right)\nabla_i\left(\frac{\phi_B}{a_0}n_i\right). \tag{3.1.3}$$

It is useful give once again the definition of $\tilde{\epsilon}$, the largest principal strain, corresponding to the eigenvalue of the deviatoric strain tensor in the direction of the (normalised) displacement field:

$$\epsilon'_{ij}n_j = \tilde{\epsilon}n_i. \tag{3.1.4}$$

Now, while in [26] the quantity $\chi = \tilde{\epsilon}^2$ is treated as an arbitrary combination of terms quadratic in the derivative of the field, two important inequalities are overlooked, namely eqs. 7.29 and 7.30 of [21], which read respectively:[2]

$$\int_B \tilde{\epsilon}^2 \mathrm{d}V \leq V_M(B) \tag{3.1.5}$$

$$\tilde{\epsilon}^2 \leq \left(\frac{d-2}{d-1}\right){\epsilon'_{ij}}^2. \tag{3.1.6}$$

In the above, B defines a ball of radius r, and the equality in 7.29 retrieves the Tully Fisher relation when used alongside 7.30. As explained in chapter 1, MOND also predicts the same relation, and hence one could use this equality to fix the coefficients of the Lagrangian. In addition, the equality of 7.29 is used to arrive at 7.37, the equation from which the Lagrangian in [26] is initially obtained. It would hence appear a reasonable step to also assume equality for 7.30. By doing so, and plugging the relation into 7.37, one obtains:

$$-{\epsilon'_{ij}}^2\left(\frac{d-2}{d-1}\right) = \left(\frac{d-2}{d-1}\right)\nabla_i\left(\frac{\phi_B n_i}{a_0}\right). \tag{3.1.7}$$

After the dimension-dependent factors cancel out, one gets:

$${\epsilon'_{ij}}^2 = \nabla_i\left(u_i\right) \implies {\epsilon'_{ij}}^2 = \chi. \tag{3.1.8}$$

If one were consistent with Hossenfelder's generalisation of all 3D quantities to tensors in 4D spacetime with the same structure (as was done for the strain tensor and the displacement vector), plugging the definition of the deviator stress tensor ϵ'_{ij} in the above we would then have:

$$\boxed{\chi = \left(\epsilon_{\mu\nu} - \frac{1}{4}\epsilon^\mu_\mu g_{\mu\nu}\right)^2 = \epsilon_{\mu\nu}\epsilon^{\mu\nu} - \frac{7}{4}\left(\epsilon^\mu_\mu\right)^2 \neq -\frac{1}{4}\epsilon_{\mu\nu}\epsilon^{\mu\nu} + \frac{1}{3}\left(\epsilon^\mu_\mu\right)^2.} \tag{3.1.9}$$

The RHS of the inequality is the form χ takes with the parameters chosen by Hossenfelder. The following can be observed:

[2] The equality for eq. 3.1.5 holds for the maximum value in the largest principle strain $\tilde{\epsilon}$.

When generalising all quantities of EG from 3D to 4D spacetime, the coefficients for χ derived from [21] do not match those utilised by Hossenfelder.

Ultimately, this does not prove that the set of coefficients stemming from the equality with the deviator stress tensor is the correct one. This would however be the most natural choice if one argues that the displacement vector u^i has the same magnitude when generalised to the 4-vector u^α. However, the next section shows that the magnitude chosen for u^α leads to a number of contradictions.

3.2. Imposter field normalisation

In [26], the motivation for the timelike nature of the imposter field u^μ is given with respect to its magnitude. Hossenfelder claims that, although no indication is given in [21] about the displacement field having a mass, setting u^μ to be massless would clash with its normalisation. Hence, a mass term is written down for the imposter Lagrangian L_θ, without a physical reason, and with no equivalent description in Verlinde's original paper.

Moreover, the only point of contact that CEG makes with observationally confirmed data is in its E.O.M.s in the nonrelativistic limit, where MOND is retrieved. However, in this limit the mass term is neglected. It is argued that this is possible when working on the scale of galaxies and clusters, the size of which is negligible compared to that of the entire visible universe. It has, however, been shown in the previous chapter that the mass term diverges on non cosmological scales, indicating that this assumption does not hold.

Removing the mass term completely from the Lagrangian would have no impact on the quantifiable predictions of the theory. It should also be mentioned that both in the derivation presented here and that of [31], the mass term is found to have either the wrong sign, or the wrong power. Therefore, it is inconsistent with either the relativistic equations of motion presented directly in [26], or with the claim of conservation of the energy-momentum tensor in a De-Sitter background,[3] or with both.

What, then, can really be said about the nature of the imposter field, and whether it should, in fact, be a timelike or lightlike vector field, or not?[4] In order to answer the question, it is best to look directly at the normalisation of the spatial component of the field, u^i, given by eq. 6.4 and by the condition given immediately below eq. 6.19 in [21], which read respectively:

$$u_i = \frac{\phi}{a_0} n_i \tag{3.2.10}$$

$$|n_i|^2 = n_i n^i = 1. \tag{3.2.11}$$

[3] This is the key claim used by Hossenfelder to fix the coefficients of the kinetic term of u^μ.

[4] This excludes a priori the possibility of a spacelike quantity to avoid acausal behaviour such as superluminal propagation.

By combining these two relations, it is easy to see that:

$$u_i u^i = \left(\frac{\phi}{a_0}\right)^2 n_i n^i = \left(\frac{\phi}{a_0}\right)^2. \tag{3.2.12}$$

After identifying the Hubble radius L as the inverse of the low acceleration scale constant a_0 as:

$$a_0 = \frac{1}{L}, \tag{3.2.13}$$

(3.2.12) becomes:

$$u_i u^i = \phi^2 L^2 \implies \phi = \frac{\sqrt{u_i u^i}}{L}. \tag{3.2.14}$$

Hossenfelder uses the normalisation:

$$\phi = \frac{\sqrt{-u_\alpha u^\alpha}}{L}. \tag{3.2.15}$$

When compared to what was just found above, this would imply, when ignoring spacetime curvature and raising indices with the Minkowski metric:[5]

$$-u_\alpha u^\alpha = u_i u^i \implies \left(u^0\right)^2 = 2u_i u^i = 2\left(\phi L\right)^2. \tag{3.2.16}$$

All that we have done so far has been to simply compare the information given in [21] to the assumptions made in [26] by equating eq. 3.2.14 to eq. 3.2.15. However, a crucial observation can now be made: in the derivation of the non-relativistic limit of the E.O.M.s in [26] from which the MOND equation is retrieved, it is assumed that it is possible to work in the rest frame of the imposter field, in which the following has to hold:

$$n^\mu \approx \left(-1, \vec{0}\right). \tag{3.2.17}$$

If this holds true, then automatically one also has:

$$n_i n^i = 0 \implies u_i u^i = 0,\ \forall\, \phi \neq 0 \tag{3.2.18}$$

directly implying, by the use of (3.2.16), that the following has to hold:

$$\left(u^0\right)^2 = 0 \implies \left(n^0\right)^2 = 0 \rightarrow n^0 = 0. \tag{3.2.19}$$

Subsequently, this translates into the statement:

As the space and time components of both n^α and u^α are proportional, one cannot vanish without the other, and a rest frame such as the one in [26] cannot be defined.

[5] This corresponds to the case of the limit in which Verlinde works, of a perturbed flat spacetime.

This is an indication that either the time-like normalisation is incorrect, or one has to accept that there is no rest frame for the imposter field in the first place.[6] Both conditions can be satisfied simultaneously: for a light-like field there is no rest frame, and the normalisation would instead be:

$$u_\alpha u^\alpha = 0. \tag{3.2.20}$$

One immediate advantage of having a light-like field is the ability to uniquely determine the time component by using the spatial component. In this particular case Verlinde provides the value for u_i, and one can then determine the time component using (3.2.20) for the case of a Minkowksi background:[7]

$$u_0 u^0 = u_i u^i \rightarrow u^0 = \phi L. \tag{3.2.21}$$

This makes it possible to explicitly write out the imposter field as:

$$u^\mu = \left(u^0, u^i\right) = \left(\phi L, \phi n^i L\right) = \phi L \left(1, n^i\right) \tag{3.2.22}$$

This explicit form makes it clear once again that, since the spatial and time component have the same magnitude, one cannot neglect either one.

3.3. Interaction Term and Metric for a Lightlike Field

When choosing a light-like normalisation for the imposter field, a problem immediately arises if one wants to use the same Lagrangian as the one adopted in [26]. The interaction term has the form:

$$L_{int} = -\frac{u^\mu u^\nu}{Lu} T_{\mu\nu}, \tag{3.3.23}$$

with $T_{\mu\nu}$ the energy momentum tensor of normal matter and $u = \sqrt{-u^\nu u_\nu}$.

> The original interaction term for CEG is incompatible with the Lagrangian for a light-like field, as it would imply division by zero.

It is thus necessary to further investigate the origin of this term and search for alternatives for the interaction between the imposter field and matter.[8]

[6] It is possible to be consistent with the spatial component given by Verlinde and keep timelike normalisation at the same time. This option will be discussed in the next chapter.

[7] One could similarly consider a Minkowski background perturbed by the same potential appearing in the vector field. Such an approach is discussed in the next chapter when exploring Generalised Einstein Aether theories.

[8] It has already been stated that having a direct interaction with matter of the form given in CEG implies the presence of two non conformally related metrics. This is incompatible with observations, and has to be avoided.

As explained for (2.2.35), the interaction Lagrangian originates from the use of an effective spatial metric $\tilde{h}_{ij}$, defined as:

$$\tilde{h}_{ij} = g_{ij} - \frac{u_i n_j}{L}. \tag{3.3.24}$$

In the above expression, g_{ij} is a generic spatial metric.[9] As has been noted already for (2.2.35), this expression stems from eq. 6.5 in [21]:

$$h_{ij} = \delta_{ij} - \frac{1}{L}\left(u_i n_j + n_i u_j\right). \tag{3.3.25}$$

However, this is not the first appearance of a spatial metric in [21], as already in eq. 5.8 the following definition was given:

$$h_{ij} = \delta_{ij} - 2\phi\left(x\right) n_i n_j. \tag{3.3.26}$$

In both cases the vector n_i is defined as:

$$n_i \equiv \frac{x_i}{|x|}. \tag{3.3.27}$$

x_i is the position vector with respect to the center of a sphere, and $|x|$ its magnitude, making n_i the unit direction vector. It is fairly straightforward to show that the two equations, 5.8 and 6.5, are in fact equivalent by using (3.2.10):

$$u_i = \phi L n_i \implies \frac{1}{L}\left(u_i n_j + n_i u_j\right) = \frac{1}{L}\phi L\left(n_i n_j + n_i n_j\right) = 2\phi n_i n_j. \tag{3.3.28}$$

It is at this point that the generalisation of h_{ij} to a spacetime metric has to deviate from the approach followed by Hossenfelder. Instead of simply promoting each 3-vector to a 4-vector (as done for $u^i \to u^\mu$, thus causing the conflict for the normalisation), one can note that the spatial metric used is, in fact, simply obtained by the general spacetime metric given in eq. 5.7 of [21], with spacetime interval:

$$\mathrm{d}s^2 = -\mathrm{d}t^2 + \mathrm{d}x_i^2 - 2\phi\left(\mathrm{d}t^2 + \frac{\left(x_i \mathrm{d}x^i\right)^2}{|x|^2}\right) = \tag{3.3.29}$$

$$-\mathrm{d}t^2\left(1+2\phi\right) + \mathrm{d}x_i^2 - 2\phi\frac{\left(x_i \mathrm{d}x^i\right)^2}{|x|^2}. \tag{3.3.30}$$

In order to arrive at (3.3.30), (3.3.27) was used. To obtain the explicit form of the metric, one can rewrite it in general form as:

$$\mathrm{d}s^2 = g_{\mu\nu}\mathrm{d}x^\mu \mathrm{d}x^\nu. \tag{3.3.31}$$

[9] Hossenfelder uses the notation h_{ij} rather than g_{ij}, but this does not correspond to the quantity h_{ij} defined by Verlinde.

This can be simplified by rewriting the coordinate component of the last term in Cartesian coordinates as:

$$\frac{\left(x_i \mathrm{d}x^i\right)^2}{|x|^2} = \frac{\left(x\mathrm{d}x + y\mathrm{d}y + z\mathrm{d}z\right)^2}{x^2+y^2+z^2}. \tag{3.3.32}$$

The equivalent matrix form reads:

$$\frac{1}{x^2+y^2+z^2} \begin{array}{c} \\ \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \end{array} \!\!\begin{array}{c} \begin{array}{ccc} \mathrm{d}x & \mathrm{d}y & \mathrm{d}z \end{array} \\ \begin{pmatrix} x^2 & xy & xz \\ yx & y^2 & yz \\ zx & zy & z^2 \end{pmatrix} \end{array}. \tag{3.3.33}$$

As the eigenvalues e_n are given by:

$$e_1 = 0, \quad e_2 = 0, \quad e_3 = 1, \tag{3.3.34}$$

the matrix in the basis of the eigenvector is:

$$\begin{array}{c} \\ \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \end{array} \!\!\begin{array}{c} \begin{array}{ccc} \mathrm{d}x & \mathrm{d}y & \mathrm{d}z \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{3.3.35}$$

After identifying the z-axis with the radial direction in spherical coordinates, one obtains:

$$\begin{array}{c} \\ \mathrm{d}r \\ \mathrm{d}\theta \\ \mathrm{d}\varphi \end{array} \!\!\begin{array}{c} \begin{array}{ccc} \mathrm{d}r & \mathrm{d}\theta & \mathrm{d}\varphi \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{3.3.36}$$

Finally, through the use of the above, we can rewrite the metric as:

$$\mathrm{d}s^2 = -\mathrm{d}t^2 \left(1+2\phi\right) + \left(1-2\phi\right)\mathrm{d}r^2 + r^2\mathrm{d}\theta^2 + r^2\sin^2\left(\theta\right)\mathrm{d}\phi^2. \tag{3.3.37}$$

It is now simple to identify the components of the metric by comparison with (3.3.30):

$$g_{00} = -\left(1+2\phi\right), \quad g_{rr} = 1-2\phi, \quad g_{\theta\theta} = r^2, \quad g_{\varphi\varphi} = r^2\sin\left(\theta\right). \tag{3.3.38}$$

This is an interesting result: the metric describes a Minkowski spacetime perturbed by a symmetric second rank tensor $k_{\mu\nu}$:

$$g_{\mu\nu} = \eta_{\mu\nu} + k_{\mu\nu}, \quad |k_{\mu\nu}| \ll 1, \tag{3.3.39}$$

as is explained e.g. in [6]. This procedure corresponds to a linearisation of gravity from the full general relativistic approach, and is useful, among other things, for treating relativistic phenomena such as light bending and gravitational radiation in the Newtonian limit. It is clear, by looking at (3.2.10), that the metric perturbation can be fully expressed in terms of the imposter field:

$$k_{00} = k_{rr} = -2u^0 a_0; \quad k_{\theta\theta} = k_{\varphi\varphi} = 0. \tag{3.3.40}$$

This apparently obvious statement has an important implication. In the work of Hossenfelder an effective metric is given, and the metric perturbation couples to matter only. As made clear in section 2.2, the effective metric is not conformally related to the background metric, which implies that there are two sets of distinct lightlike geodesics in the spacetime. This inevitably leads to one of two problematic consequences. The first is:

> If the imposter field couples to all nongravitational energy universally, one has a bimetric theory, in which photons and gravitational waves travel along separate geodesics.

Any theory predicting this behaviour can be falsified by observations indicating that photons and gravitational waves have the same propagation speed. The discussion in the next chapter will prove that such observations have indeed occurred, falsifying CEG if one chooses to so interpret the interaction between energy and the imposter field. The second alternative is the following:

> Due to an unknown physical mechanism, the imposter field couples only to matter, but not to the energy momentum tensor of the EM field, and hence photons.

This second interpretation would exempt the theory from being falsified based on observations on the equivalence of propagation speed between light and gravity. Nonetheless, the following paradox follows directly from this approach: In CEG no transition exists between Newton and MOND regimes, and the potential is always the sum of the two contributions. We can translate this statement to the following infinitesimal transformation of the Newton potential ϕ when adding the deep MOND potential ϕ_M:

$$\phi \to \tilde{\phi} = \phi + \zeta\phi_M. \tag{3.3.41}$$

Here ζ functions as a bookkeeping parameter to indicate the infinitesimal change. We can then note that, for small r, ϕ and ϕ_M are both negative. This directly implies that, for small r, $\tilde{\phi} < \phi$. Writing the spacetime interval for the Schwarzschild metric as a function of the Newton potential plus infinitesimal addition yields:[10]

$$\mathrm{d}s^2 = -\left(1 + 2\tilde{\phi}\right)\mathrm{d}t^2 + \frac{1}{1 + 2\tilde{\phi}}\mathrm{d}r^2 + r^2\mathrm{d}\Omega^2. \tag{3.3.42}$$

It follows that the Schwarzschild radius $\tilde{r}_s$ is shifted in this spacetime:

$$\boxed{r_s = 2GM = -2r\phi \to \tilde{r}_s = -2r\tilde{\phi} > r_s.} \tag{3.3.43}$$

[10] This is the exact form of the Schwarzschild metric. Only a cosmetic change was carried out, $-GM/r \to \phi$.

> As the Schwarzschild radius represents the event horizon, the spacetime would possess two event horizons. A photon could probe the region $r_s < r < \tilde{r}_s$ inside of the matter horizon, emerge and carry the quantum mechanical information describing the spacetime beyond the horizon. Through the interaction with matter, this quantum mechanical information could be passed on to a massive particle, which could have ventured past the matter horizon $\tilde{r}_s$ by itself.

The identification of a single spacetime metric makes it possible to avoid the problems stemming from a non conformal effective metric.[11] All gravitational phenomena can then be treated on the same footing. Moreover, a self interaction for the imposter field is automatically introduced, with the field EOM evolving on a background that is perturbed by the field itself.

The formalism for linearised gravity from [6] will now be followed to identify several crucial quantities which will be needed for the treatment of gravitational lensing. The next step is to separate the metric perturbation $h_{\mu\nu}$ into irreducible representations of the rotation group, which are tensors of various rank that are transformed into themselves under spatial rotations:

$$h_{00} = -2\phi \tag{3.3.44}$$

$$h_{0i} = w_i = 0 \tag{3.3.45}$$

$$h_{ij} = 2s_{ij} - 2\psi\delta_{ij} \to \psi = -\frac{1}{6}\delta^{ij}h_{ij};\; s_{ij} = \frac{1}{2}\left(h_{ij} - \frac{1}{3}\delta^{kl}h_{kl}\delta_{ij}\right) = 0. \tag{3.3.46}$$

In the last line, s_{ij} vanishes as the metric is diagonal and s_{ij} encodes the traceless components of h_{ij}.

3.3.1. An Important Observation on Linearised Gravity

Using the formalism in which one linearises the metric and resulting potentials describing the spacetime might seem contradictory. This is due to the fact that the redundant degrees of freedom allow the choice of a gauge. In the transverse gauge, for example, the Einstein equations become $\nabla^2\phi = 4\pi G\rho$.[12] This implies that the potential has to satisfy the Poisson equation. However, the above relation only holds when the only terms present in the action are the Ricci scalar and the Lagrangian for the source fields.

[11] One could explore the possibility of a conformal metric, but that route is not followed throughout this work.

[12] The expression for the transverse gauge is given in a simplified form, where one can observe that the scalar degree of freedom Ψ corresponds to the gravitational potential and the T_{00} component of the energy momentum is given by the matter density.

The linearity constraint on the potential arises when deriving the Einstein equations from the GR action with sources. Adding a vector field to the action modifies the Einstein equations and the PDE for the ϕ. Linearity of the ϕ is no longer generally imposed.

3.4. Geodesics of the Perturbed Metric

One phenomenon which is used for probing the (apparent) mass density of galaxies in clusters is gravitational lensing. As photons always travel along geodesics, it is necessary to derive the geodesic equations for the metric (3.3.29). As described in section 1.7.4, these can be obtained by varying the action:

$$S = \int |\mathrm{d}s| = \int \mathrm{d}\lambda \left(-g_{\mu\nu}\frac{\mathrm{d}x^{\mu}}{\mathrm{d}\lambda}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\lambda}\right)^{\frac{1}{2}}. \tag{3.4.47}$$

A helpful simplification can be made: since for timelike paths $|\mathrm{d}s| \neq 0$, extremising $|\mathrm{d}s|$ also extremises $\mathrm{d}s^2$, which means the action can be chosen to be:[13]

$$S = \int \mathrm{d}\lambda \left(\frac{\mathrm{d}s}{\mathrm{d}\lambda}\right)^{2} = \int \mathrm{d}\lambda \left(g_{\mu\nu}\frac{\mathrm{d}x^{\mu}}{\mathrm{d}\lambda}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\lambda}\right) \rightarrow L = g_{\mu\nu}\frac{\mathrm{d}x^{\mu}}{\mathrm{d}\lambda}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\lambda}. \tag{3.4.48}$$

For massive particles, the affine parameter is $\lambda = \tau$, the proper time. The geodesic equations are then the **Euler Lagrange** (**EL**) equations, one for each coordinate. It is convenient to work in spherical coordinates to simplify the calculations. The resulting Lagrangian is:

$$L = -\left(1+2\phi\right)\dot{t}^{2} + \left(1-2\phi\right)\dot{r}^{2} + r^{2}\dot{\theta}^{2} + r^{2}\sin^{2}\left(\theta\right)\dot{\varphi}^{2}. \tag{3.4.49}$$

It is important note once again the difference between the field ϕ and the coordinate φ. As shown in section 1.7.4, the EL equations read:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\left(\frac{\partial L}{\partial \dot{x}^{\mu}}\right) = \frac{\partial L}{\partial x^{\mu}}. \tag{3.4.50}$$

The geodesic equation is given as:

$$\frac{\mathrm{d}^{2}x^{\mu}}{\mathrm{d}\tau} + \Gamma^{\mu}_{\nu\rho}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\tau}\frac{\mathrm{d}x^{\rho}}{\mathrm{d}\tau}. \tag{3.4.51}$$

[13] It may seem that this procedure is going against the goal of obtaining geodesics for photons, which always travel on paths made of points separated by a zero spacetime interval. However, it can be shown that the result of the calculations for timelike geodesics is also valid for lightlike geodesics.

As previously mentioned, the Christoffel symbols can be obtained directly from the geodesic equation for each coordinate. Assuming spherical symmetry, ϕ only depends on r and we obtain:[14]

$$\frac{\mathrm{d}\phi}{\mathrm{d}\tau} = \dot{r}\partial_r\phi. \tag{3.4.52}$$

Working out (3.4.50) then yields the four equations:[15]

$$\ddot{t} + \dot{t}\dot{r}\frac{2\partial_r\phi}{1+2\phi} = 0 \tag{3.4.53}$$

$$\ddot{r} - \dot{r}\dot{r}\frac{\partial_r\phi}{1-2\phi} - \dot{\theta}\dot{\theta}\frac{r}{1-2\phi} - \dot{\varphi}\dot{\varphi}\frac{r\sin^2(\theta)}{1-2\phi} + \dot{t}\dot{t}\frac{\partial_r\phi}{1-2\phi} = 0 \tag{3.4.54}$$

$$\ddot{\theta} + \dot{\theta}\dot{r}\frac{2}{r} - \dot{\varphi}\dot{\varphi}\sin(\theta)\cos(\theta) = 0 \tag{3.4.55}$$

$$\ddot{\varphi} + \dot{r}\dot{\varphi}\frac{2}{r} + \dot{\varphi}\dot{\theta}\frac{2\cos(\theta)}{\sin(\theta)} = 0. \tag{3.4.56}$$

We can hence write down, after defining $k \equiv 1 - 2\phi$, in order to tidy up the expressions:[16]

$$\Gamma^t_{tr} = \frac{\partial_r\phi}{1+2\phi} \quad \Gamma^r_{rr} = -\Gamma^r_{tt} = -\frac{\partial_r\phi}{k} \quad \Gamma^r_{\theta\theta} = -\frac{r}{k} \tag{3.4.57}$$

$$\Gamma^r_{\varphi\varphi} = -\frac{\sin^2(\theta)\,r}{k} \quad \Gamma^\theta_{\theta r} = \Gamma^\varphi_{\varphi r} = \frac{1}{r} \tag{3.4.58}$$

$$\Gamma^\theta_{\varphi\varphi} = -\sin(\theta)\cos(\theta) \quad \Gamma^\varphi_{\varphi\theta} = \frac{\cos(\theta)}{\sin\theta} \tag{3.4.59}$$

An important aspect of a spacetime is given by the symmetries it exhibits.[17] Two types of constants of motion can be identified and used to simplify the geodesic equations. The first is the magnitude of the four-velocity v^μ:[18]

$$-g_{\mu\nu}\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}\frac{\mathrm{d}x^\nu}{\mathrm{d}\lambda} = -v_\mu v^\mu = \varepsilon \rightarrow \begin{cases} 0 & \text{lightlike path} \\ 1 & \text{timelike path} \end{cases}. \tag{3.4.60}$$

The others are given by the Killing equation:

$$K_\mu\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda} = C. \tag{3.4.61}$$

[14] One should remember that all of Verlinde's conclusions are only valid for a spherically symmetric potential.

[15] As before, the notation $\dot{x}$ represents differentiation w.r.t. the proper time τ.

[16] One must bear in mind that Christoffel symbols are symmetric in the lower two indices. Therefore, a factor of two has to be accounted for when the two lower indices are not equal (as they are summed over, and the two combinations are equivalent).

[17] As in the case of Noether's theorem in classical field theory, symmetries in GR are often related to conserved quantities, although definitions for energy and momentum, for example, do not always match their flat spacetime counterparts.

[18] The notation v^μ is not consistent with the literature, where the 4-velocity is usually u^μ. v^μ is used to avoid confusion with the vector field treated so far. For a lightlike particle, the velocity w.r.t. the affine parameter λ is equal to the 4-momentum, $v^\mu = \mathrm{d}x^\mu/\mathrm{d}\lambda$.

C is a constant, three spatial Killing vectors are present for spherical symmetry, and another one is found due to invariance under time translation (see e.g. [6]). The three spatial Killing vectors can be identified with the magnitude and direction of the angular momentum in a plane, while the one related to time translation can be identified with the energy of a test particle.

The spacetime thus has five conserved quantities: the magnitude of v^α, the two directions of angular momentum, the magnitude of angular momentum and the total energy.

The conservation of the direction of angular momentum allows one to limit the analysis to that of the motion of a test body in a plane. Without loss of generality, one can hence set:

$$\theta = \frac{\pi}{2} \implies \sin\left(\theta\right) = 1 \implies \dot{\theta} = 0. \tag{3.4.62}$$

The two remaining Killing vectors can be found in two equivalent ways: by use of the Killing equation or by identifying the constants of the motion from the relevant geodesic equations for t and φ.[19] From the geodesic equations, we observe the following:

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\partial L}{\partial \dot{t}}\right) = \frac{\mathrm{d}}{\mathrm{d}\lambda}\left(-2\overbrace{\left(1+2\phi\right)\dot{t}}^{E}\right) = 0 \tag{3.4.63}$$

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\partial L}{\partial \dot{\varphi}}\right) = \frac{\mathrm{d}}{\mathrm{d}\lambda}\left(2\sin^2\left(\theta\right)\overbrace{\dot{\varphi}r^2}^{L}\right) = 0. \tag{3.4.64}$$

E is the total energy and L is the magnitude of the angular momentum. In the same qay, by noting that the metric does not depend on t or φ, one can identify the Killing equations for the related Killing vectors, K_μ and R_μ:

$$K^\mu = \left(\partial_t\right)^\mu = \left(1,0,0,0\right) \to K_\mu = \left(-\left(1+2\phi\right),0,0,0\right) \to -K_\mu\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda} = \left(1+2\phi\right)\dot{t} = E \tag{3.4.65}$$

$$R^\mu = \left(\partial_\varphi\right)^\mu = \left(0,0,0,1\right) \to R_\mu = \left(0,0,0,r^2\right) \to R_\mu\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda} = r^2\dot{\varphi} = L. \tag{3.4.66}$$

In both approaches the conservation of angular momentum was used to impose $\theta = \frac{\pi}{2}$. To arrive at a unique equation of motion by using the obtained constants, (3.4.60) can be used and it becomes:

$$g_{\mu\nu}\frac{\mathrm{d}x^\mu}{\mathrm{d}\lambda}\frac{\mathrm{d}x^\nu}{\mathrm{d}\lambda} = -\left(1+2\phi\right)\dot{t}^2 + \left(1-2\phi\right)\dot{r}^2 + r^2\dot{\varphi}^2 = -\varepsilon. \tag{3.4.67}$$

By multiplying each side by $1+2\phi$ and plugging in E and L, this can be simplified to:

$$-E^2 + \left(1-4\phi^2\right)\dot{r}^2 + r^2\left(1+2\phi\right)\dot{\varphi}^2 = -\varepsilon\left(1+2\phi\right). \tag{3.4.68}$$

[19] These are the directions in which energy is conserved.

The above yields, after isolating the $\dot{r}^2$ term:

$$\dot{r}^2 + \frac{L^2}{r^2\,(1-2\phi)} + \frac{\varepsilon}{1-2\phi} = \frac{E^2}{1-4\phi^2}. \tag{3.4.69}$$

As it was assumed in [21] that the metric is used in the regime in which $\phi \ll 1$, it is a sensible approximation to expand the expression to first order in ϕ. By doing so, one obtains (for $\varepsilon = 1$, giving a timelike path):

$$\frac{\dot{r}^2}{2} + \frac{1}{2} + \frac{L^2}{2r^2} + \left(1 + \frac{L^2}{r^2}\right)\phi = \frac{E^2}{2}. \tag{3.4.70}$$

Again following [6], with $\varepsilon = 1$ to analyse timelike paths, such as those of celestial bodies in galaxies and clusters, one can interpret this equation as describing a particle of unit mass and energy $\mathcal{E}$ (different from the conserved quantity ε) in a one dimensional potential $V(r)$,[20] where one identifies:

$$\mathcal{E} = \frac{E^2}{2} \tag{3.4.71}$$

$$V\,(r) = \frac{1}{2} + \frac{L^2}{2r^2} + \left(1 + \frac{L^2}{r^2}\right)\phi. \tag{3.4.72}$$

The first term is a constant, the following two are contributions already accounted for by Newtonian gravity, and the last term is the general relativistic contribution. Expression (3.4.70) can then be put in the simplified form:

$$\frac{1}{2}\dot{r}^2 + V = \mathcal{E}. \tag{3.4.73}$$

The Schwarzschild equivalent reads:

$$\boxed{V = \frac{1}{2} + \frac{L^2}{2r^2} - \frac{GM}{r} - \frac{L^2 GM}{r^3} = \frac{1}{2} + \frac{L^2}{2r^2} + \left(1 + \frac{L^2}{r^2}\right)\phi_N.} \tag{3.4.74}$$

The only difference between (3.4.74) and (3.4.72) is given by the use of the Newtonian potential ϕ_N rather than the modified potential ϕ obtained by adding a vector field $u^\alpha = \phi L\,(1, n^i)$ to the GR Lagrangian.

This similarity should not be surprising since by rewriting the Schwarzschild metric as a function of the Newtonian potential ϕ_N, one obtains:[21]

$$\mathrm{d}s^2 = -\,(1 + 2\phi_N)\,\mathrm{d}t^2 + \frac{1}{1+2\phi_N}\mathrm{d}r^2 + r^2\mathrm{d}\Omega^2. \tag{3.4.75}$$

[20] It needs to be pointed out that the potential V is not related to gravity, but emulates the potential of a 1D harmonic oscillator.

[21] Compare to eq. 3.4.67 for the MOND-like potential ϕ.

After expanding to 1^{st} order in ϕ_N the above becomes:

$$\mathrm{d}s^2 \approx -\left(1+2\phi_N\right)\mathrm{d}t^2 + \left(1-2\phi_N\right)\mathrm{d}r^2 + r^2\mathrm{d}\Omega^2. \tag{3.4.76}$$

In this approximation, the two metrics coincide, with the difference given by the gravitational potential present. It is important to then conclude the following:

> The analysis just carried out for the geodesics of the spacetime used to derive EG is only valid when only one metric is present. In this case, it can be seen that the results for the Schwarzschild background can be recovered to first order, and in the Newtonian regime in which $\phi = \phi_N$.

3.5. E.O.M.s directly from Verlinde's equations

An alternative way to obtain a PDE describing the behaviour of the potential associated with the imposter field is to look at the previously used eq. (7.37) from [21], and directly follow the definition of the quantities given by Verlinde, rather than coming up with an independent Lagrangian. The equation is given again for convenience:

$$\left(\frac{8\pi G}{a_0}\Sigma_D\right)^2 = -\left(\frac{d-2}{d-1}\right)\nabla_i\left(\frac{\phi_B}{a_0}n_i\right) \xrightarrow{\Sigma_D=\frac{a_0}{8\pi G}\tilde{\epsilon}} \tilde{\epsilon}^2 = -\frac{2}{3}\nabla_i\left(\frac{\phi_B}{a_0}n_i\right). \tag{3.5.77}$$

If one is coherent with the use of partial derivatives with no Christoffels, the RHS is proportional to:

$$-\boxed{\nabla_i\left(\frac{\phi_B}{a_0}n_i\right) = -\left(\nabla_i\phi_B\right)n_i - \left(\nabla_i n_i\right)\phi_B = -\frac{Gm}{r^2} + 2\frac{Gm}{r^2} = \frac{Gm}{r^2}.} \tag{3.5.78}$$

In addition, to simplify (3.5.77) it was used that the spacetime has $d=4$ dimensions, and the notation kept the same as in [21] to facilitate comparison. In [26], it is assumed that $\tilde{\epsilon}$ is an unknown quantity, and its value is not fixed a priori from the information contained in [21]. However, that is not necessarily the case: as noted in section 3.1, Verlinde gives the definition of $\tilde{\epsilon}$ as the largest eigenvalue of the deviatoric strain tensor ϵ'_{ij}:

$$\epsilon'_{ij}\tilde{n}_j = \tilde{\epsilon}\tilde{n}_i. \tag{3.5.79}$$

A distinction has to be made between $\tilde{n}_i$, the normalised eigenvector of the deviatoric strain tensor corresponding to its largest eigenvalue, and n_i used on the RHS of (3.5.77), which is instead defined as $n_i = x_i/\left|x_i\right|$, although the same notation is used to refer to both quantities in different sections of [21].[22] We can then observe the following: Verlinde

[22] The ambiguity does not cause major problems, as both quantities can ultimately be represented by unit vectors.

never explicitly makes use of Christoffel symbols. It should then be possible, at least in principle, to treat (3.5.77) through the use of partial derivatives only, albeit in Minkowski spacetime and with spherical coordinates.[23] In order to make it explicit that the expression will be treated with partial derivatives, and in Minkowski spacetime, it is convenient to re-express (3.5.77) through the usual notation:

$$\tilde{\epsilon}^2 = -\frac{2}{3}\partial_\mu \left(\frac{\phi_B}{a_0} n^\mu\right). \tag{3.5.80}$$

As explained in section 3.2, using the normalisation[24] $u_\mu u^\mu = -(\phi/a_0)^2$ is inconsistent with the spatial normalisation $u_i u^i = (\phi/a_0)^2$. To remain consistent with the spatial normalisation of EG, one instead considers a lightlike imposter field. The generalisation of the spatial vector n^i is then given to satisfy.

$$n_\mu n^\mu = 0 \implies n_0 n^0 = -n_i n^i \implies \left(n^0\right)^2 = \left|n^i\right| = 1 \implies n^0 = \pm 1, \tag{3.5.81}$$

In the above expression, it was noted that n^i is a spatial unit vector, and the negative root was chosen. Choosing the direction of n^i to be radial gives:

$$n^\mu = (-1, 1, 0, 0) \implies n_\mu = (1, 1, 0, 0) \implies u_\mu = \left(\frac{\phi}{a_0}, \frac{\phi}{a_0}, 0, 0\right). \tag{3.5.82}$$

Hence, contrary to what was claimed by Hossenfelder in [26], the quantity n^i, and hence n^μ can be calculated explicitly.[25] Therefore, the RHS of (3.5.77) is obtained as:

$$-\frac{2}{3}\frac{1}{a_0}\partial_\mu \left(\phi_B n^\mu\right) = \frac{2}{3}\frac{1}{a_0}\partial_r \phi_B. \tag{3.5.83}$$

At this point it is important to realise that ϕ_B is the known Newtonian potential for ordinary baryonic matter. Therefore, for the limit of a spherically symmetric potential with a mass m in the origin, as treated in both [21] and [26], the expression can be explicitly evaluated as:

$$\boxed{\frac{2}{3}\frac{1}{a_0}\partial_r \phi_B = \frac{2}{3}\frac{1}{a_0}\frac{Gm}{r^2}.} \tag{3.5.84}$$

Next, the LHS of (3.5.77) has to be evaluated. As we have to find the eigenvalue of the deviatoric stress tensor, it is most convenient to express this quantity as a 4x4 matrix. To

[23] This would of course be incorrect. As mentioned before, non vanishing Christoffels exist for flat spacetime in curvilinear coordinates.

[24] It should again be noted that the a_0 used in Verlinde, and hence in Hossenfelder, does not correspond to the value of Milgrom's constant, as it is larger by a factor of 6.

[25] One can uniquely (up to a sign difference) define the timelike component of a 4-vector if the magnitude and spatial components are known. As the magnitude of a lightlike 4-vector is always zero, the timelike component depends exclusively on the chosen spacetime.

do so, one rewrites the expression:[26]

$$\partial_\mu u_\nu = \begin{array}{c} \\ \mu=0 \\ \mu=1 \\ \mu=2 \\ \mu=3 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \nu=0 & \nu=1 & \nu=2 & \nu=3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{\partial_r\phi}{a_0} & \frac{\partial_r\phi}{a_0} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{3.5.85}$$

Now, as in matrix form $\partial_\mu u^\nu$ is the transpose of $\partial_\nu u^\mu$, we find for the strain tensor:

$$\epsilon_{\mu\nu} = \frac{1}{2}\left(\partial_\mu u_\nu + \partial_\nu u_\mu\right) = \frac{\partial_r\phi}{a_0}\begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.5.86}$$

Finally, to find the deviatoric strain tensor we have to take the contraction of the strain tensor, which in matrix form simply amounts to taking its trace. Hence, the result is:[27]

$$\epsilon'_{\mu\nu} = \epsilon_{\mu\nu} - \frac{1}{4}g_{\mu\nu}\epsilon^\kappa_\kappa = \frac{\partial_r\phi}{a_0}\begin{pmatrix} \frac{1}{4} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{3}{4} & 0 & 0 \\ 0 & 0 & -\frac{1}{4}r^2 & 0 \\ 0 & 0 & 0 & -\frac{1}{4}r^2\sin(\theta) \end{pmatrix}. \tag{3.5.87}$$

The largest positive eigenvalue of the matrix[28] is given by:

$$\tilde{\epsilon} = \frac{1}{4}\left(2+\sqrt{5}\right)\frac{\partial_r\phi}{a_0} \to \tilde{\epsilon}^2 = \frac{9+4\sqrt{5}}{16}\left(\frac{\partial_r\phi}{a_0}\right)^2. \tag{3.5.88}$$

Using this result, one can finally rewrite (3.5.77) as a PDE relating the potential stemming from the imposter field to the Newtonian potential originating exclusively from baryonic matter:

$$\frac{9+4\sqrt{5}}{16}\left(\frac{\partial_r\phi}{a_0}\right)^2 = \frac{2}{3}\frac{1}{a_0}\frac{GM}{r^2}. \tag{3.5.89}$$

Choosing the positive root, the solution to (3.5.89) is given by

$$\boxed{\phi = 4\sqrt{6-\frac{8\sqrt{5}}{3}}\sqrt{GMa_0}\ln(r) + C \approx \frac{3}{4}\sqrt{GMa_0}\ln(r) + C.} \tag{3.5.90}$$

[26] In this formulation, the deviator stress tensor is not a properly defined tensor, due to the use of partial rather than covariant derivatives, which differ in the case of Minkowski space in spherical coordinates.

[27] As in the previous chapters, the metric signature is taken as $(-+++)$.

[28] The eigenvalue is taken after fixing $\theta = \frac{\pi}{2}$, which is allowed in the case of spherical symmetry.

C is a yet undetermined constant. At this point, it is important to note the striking resemblance between the equation just found for the potential and the solution to the deep-MOND regime given in [12], which is:

$$\phi_M = \sqrt{GMa_0}\ln\left(\frac{r}{r_0}\right) + \mathcal{O}\left(r^{-1}\right). \tag{3.5.91}$$

In the above, r_0 is an arbitrary radius. (3.5.90) also resembles the potential found in [26] by solving the nonrelativistic EOM in CEG through the approximation of covariant derivatives by partial derivatives:[29]

$$\phi_H = \sqrt{GMa_0}\ln\left(\frac{r}{GM}\right) + C. \tag{3.5.92}$$

Another important factor must be taken into account when comparing the solution obtained to the one from [12]: (3.5.91) is not the solution of the full theory but only of the limit in which the acceleration falls below the empirically defined critical value a_0. In MOND the transition is smooth, and given by an interpolation function that reduces to the deep-MOND limit in the $a \ll a_0$ case and to standard Newtonian gravity for $a \gg a_0$. Conversely, in EG the transition to the so called "dark gravity" limit happens abruptly for $\tilde{\epsilon} < 1$, corresponding to:

$$\frac{1}{4}\left(2+\sqrt{5}\right)\frac{\partial_r\phi}{a_0} < 1 \rightarrow \partial_r\phi = a < \frac{4}{2+\sqrt{5}}a_0 \approx a_0. \tag{3.5.93}$$

Therefore, apart from the lack of a smooth transition, the "dark gravity" regime can be identified with the deep-MOND limit. Therefore, not only was the deep-MOND solution recovered from [21] without any additional assumption not contained in the original work, but its domain of validity was also independently found.

However, repeating the same procedure while treating covariant derivatives as such, hence utilising the nonzero Christoffels of Minkowski spacetime in spherical coordinates, does not lead to a logarithmic potential. Consequently, although the initial results seem encouraging, the following must be recognised:

> The spacetime generalisation of eq. 7.37 of [21], with the required - sign on the RHS, yields the desired potential.

As the fundamental quantity describing EG is the displacement field u^i, it seems reasonable to build a theory through a sensible spacetime generalisation of this quantity.

The next chapter will show how the displacement vector can be generalised to a class of vector tensor theories capable of retrieving both MOND and Newton limits, as well as reproducing the predictions of GR. These theories are known as **GEA** (Generalised Einstein Aether).

[29] However, one should remember that this expression was obtained by assuming a timelike field and neglecting the mass term.

Chapter 4

Einstein-Aether theory

As was stated throughout the previous chapter, a relativistic generalisation of EG should not follow the framework of CEG, due to the numerous inconsistencies with the theory. Nonetheless, introducing a 4-vector to modify the Einstein equations of GR is a possibility that has been widely explored in the literature. Although various theories can match many of the predictions of GR while introducing a new field in the Lagrangian, most do not jointly introduce a solid theoretical foundation motivating the addition of the field.

It is hence interesting to analyse some of these theories, to see if a generalised EG paradigm can be used to coherently explain the structure of the chosen Lagrangians, and also to account for the success these theories have in matching observations.

This chapter will begin by describing the general theoretical frameworks that will be used, and makes the connection between EG and GEA theories. It will be shown that the identification of a 4-vector generalisation of the displacement vector u^i with the aether vector is consistent and matches all observational constraints placed on GEA so far.

4.1. CEG in the framework of Tensor-Vector theories

The analysis of CEG has revolved around the initial formulation given by Hossenfelder in [26], describing a Lagrangian of the form:

$$L_{tot} = \overbrace{GR}^{A} + \overbrace{L_M}^{B} + \overbrace{L_{int}}^{C} + \overbrace{L_\theta}^{D} \ . \tag{4.1.1}$$

Terms A and B are familiar. When used to build an action, the first yields[1] the well known Einstein-Hilbert action (see, for example, [6]):

$$S_H = \int \sqrt{-g}\, R \,\mathrm{d}^4x. \tag{4.1.2}$$

[1] The expression is valid up to a constant factor varying between conventions.

R is the Ricci curvature scalar described in section 1.5. The variation of the action w.r.t. to the metric $g^{\mu\nu}$ can then be set to zero,[2] equivalent to the vanishing of the corresponding variational derivative:

$$\frac{1}{\sqrt{-g}}\frac{\delta S_H}{\delta g^{\mu\nu}} = 0. \tag{4.1.3}$$

The above generates the Einstein equations in vacuum:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = G_{\mu\nu} = 0. \tag{4.1.4}$$

To obtain the RHS of (4.1.4), we vary the action describing the relevant matter fields, again w.r.t. the metric $g^{\mu\nu}$, obtaining the energy-momentum tensor $T_{\mu\nu}$ as:

$$T_{\mu\nu} = -2\frac{1}{\sqrt{-g}}\frac{\delta S_M}{\delta g^{\mu\nu}}. \tag{4.1.5}$$

When varying the total action comprising the Einstein-Hilbert term and the mass term, with the correct constant of proportionality, we then retrieve:

$$S_{GR} = \frac{1}{16\pi G}S_H + S_M \implies \frac{1}{\sqrt{-g}}\frac{\delta S_{GR}}{\delta g^{\mu\nu}} = 0 \implies G_{\mu\nu} = 8\pi G T_{\mu\nu}. \tag{4.1.6}$$

These are the Einstein equations in the presence of matter, which relate the curvature of spacetime to the presence of mass (and energy), here for a massive matter field. Thus, the following holds:

In GR, the only field coupling to energy is the metric tensor $g_{\mu\nu}$.

However, in CEG the two extra terms C and D introduce a new field to which matter is coupled. As also noted directly in [26], the inclusion of the interaction term C is equivalent to replacing, only in the matter Lagrangian, the background metric of the theory, satisfying the Einstein equations in the absence of external fields, by an effective metric $\tilde{g}_{\mu\nu}$, given as:

$$\tilde{g}_{\mu\nu} = g_{\mu\nu} - a_0\frac{u^\mu u^\nu}{u}. \tag{4.1.7}$$

As usual, u is the magnitude of the vector field u^μ.

There are now two separate metric tensors in the theory, one used to derive the curvature of the underlying spacetime and used in the Einstein-Hilbert action, and the other coupling to energy sources. There are, therefore, two distinct spacetime geometries, each defining its connections and geodesics. As discussed in section 2.2, the two metrics are not conformally related, ergo:[3]

[2] Throughout this chapter, variations are taken w.r.t. the inverse metric. It can be shown that this is equivalent to variations w.r.t. to the metric itself, as $\delta g_{\mu\nu} = -g_{\mu\rho}g_{\nu\sigma}\delta^{\rho\sigma}$, see, for example, [6].

[3] The two interpretations for the coupling of the imposter field to matter were given in section 3.3. Here, the assumption is made that the imposter couples to all energy sources, but it was, however, shown that an exclusive coupling to matter, and not other fields carrying energy, is not viable.

Massive and massless particles travel on the geodesics defined by the effective metric $\tilde{g}_{\mu\nu}$, whereas gravitational perturbations, such as gravitational waves and gravitons, travel along the geodesics of the background metric defined in the Einstein-Hilbert action, $g_{\mu\nu}$.

This is a very important factor which will be referred to in the following section. The last term, D, represents the Lagrangian of the vector field itself, giving its kinetic and mass terms, as extensively discussed in the previous sections. One of the least convincing aspects of CEG is its apparent arbitrariness in determining the parameters of the theory, such as the coefficients in the kinetic term and the normalisation of the field itself. It was shown in sections 3.1-3.3 that more natural choices of coefficients differ from those of CEG, but also fail to produce a logarithmic potential. It should be noted that, for a massive field such as u^μ, the invariant scalar can be interpreted as the rest mass, equal in any reference frame. This identification is absent in CEG. Furthermore the mass term is ignored when achieving the only quantifiable connection of the theory to observational evidence, namely, the reproduction of a MOND-like potential in the non-relativistic limit.

4.2. GW170817 and Constraints to modified gravity theories

At the time of the redaction and publication of [26], there was no experimentally sound way to constrain the parameters of modified gravity theories. That is no longer the case. In August of 2017, a gravitational wave (GW) signal was detected by the LIGO-VIRGO interferometers, and inferred to belong to the merger of a system of binary neutron stars [33]. From a gravitational point of view, the event did not carry more relevance than, say, the GW detection from a merger of black holes, but it did allow for an extremely relevant set of observations, belonging to the field of **multi-messenger astronomy**.

While the result of the merger of a pair of black holes gives rise to a black hole,[4] the collision of the binary neutron stars responsible for the GW179817[5] generated a variety of EM phenomena which were detected across the spectrum [34], for example **Gamma Ray Bursts** (GRB). The gravitational and EM events were traced back to the same cosmological source [35]. Accordingly:

The arrival times of the corresponding radiation provided the opportunity to test to very high accuracy the difference in propagation speed between gravitational and EM waves.

This could be used to further constrain the WEP, which essentially states that all test particles with given initial position and velocity travel along the same path.

[4] Any debris directly involved in the collision would be in proximity of one of the two event horizons, hence unable to escape after the merger. This goes for both matter fields and radiation.

[5] This is to be read as: Gravitational Wave event *yymmdd*, with *y*ear *m*onth and *d*ay respectively

It is now important to introduce the calculation of the time taken for a lightlike particle propagating along a geodesic to reach an observer, for a given metric. This is called the **Shapiro delay** [36], and it is calculated by dividing the distance from the source to the observer by the speed of light along the path connecting the two. At first glance, it may seem that the Shapiro delay would be equal for any source. However, this is not the case, as one has to take into the account the gravitational time delay due to the energy distribution determining the spacetime curvature along the line of sight[6].

Modified gravity is used, first and foremost, to explain the phenomena which cannot be accounted for by the gravitational properties of observable matter in the framework of GR.[7] Accordingly, the use of a direct coupling or effective metric must determine geodesics which reflect the (apparent) dark matter density at a given point in spacetime, which differs from the visible baryonic mass distribution.

Thus, for given baryonic and (apparent) dark matter densities, one can calculate the difference in Shapiro delay of particles travelling along the geodesics set by the two separate sources.

Throughout the observations borne by the GW10817 event, it was shown in [35] that the expected difference in Shapiro delay between GW and photons was of 400 ± 90 days. On the contrary, the first GRB signal was detected [37] only 1.7s after the GW detection. During the following hours, days and months, multiple EM signals were detected from a source consistent with the location in space pinpointed by GW170817, ranging across the EM spectrum from X-Ray to InfraRed and traceable to behaviour consistent with that of a binary neutron star merger [38].

The data from GW170817 is therefore sufficient to falsify CEG, given its reliance on a direct coupling between matter and the extra vector field introduced, equivalent to the existence of two non conformally related metrics.

Nevertheless, GW170817 cannot provide a precise suggestion as to how to build a healthy theory with the CEG approach. This is due to fact that the parameters of CEG, such as vector field normalisation and matter interaction term, do not fall within the framework of other tensor-vector gravity theories.

Therefore, it is important to identify one of the possible frameworks present in the literature as a candidate on which to rebuild a truly covariant theory of emergent gravity, not plagued by the experimentally falsified predictions of CEG.

[6] This phenomenon was first described as an experimental test of GR in 1964. Its full potential can now be exploited thanks to multi messenger astronomy.

[7] Many relevant systems can be approximated by GR's nonrelativistic-weak field Newtonian limit.

4.3. Generalised Einstein-Aether theories

4.3.1. A Strong Hint: the TeVeS Aether

It would be of interest to consider theories which have been shown to reproduce both MONDian and Newtonian behaviour in the correct regimes. One of the most representative relativistic expansions of MOND was introduced by one of the founding fathers of MOND itself, Jacob Bekenstein.

This theory, called **Tensor Vector Scalar** (**TeVeS**) gravity, added a scalar field and a vector field to the traditional Einstein-Hilbert action, in order to retrieve the correct behaviours of GR and MOND in the respective regimes [39]. Although incapable of explaining certain cosmological observations such as the power spectrum of the **Cosmic Microwave Background** (**CMB**), the theory was successful in recovering the required limits of Newtonian and MONDian behaviour. However, in the light of the GW170817 observations, a clear problem for the theory was the presence of two distinct metrics, with one coupling to matter and the other representing the spacetime background. Nonetheless, it was shown in [40] that the theory could in fact be rewritten entirely as a vector-tensor theory, thus avoiding the problematic nature of bi-metric theories,[8] and making a strong connection with another type of vector-tensor gravity: **Generalised Einstein-Aether** (**GEA**) theories.

This connection between TeVeS and GEA theories is a strong signal that a GEA theory can reproduce MOND phenomenology, starting from a covariant approach with an action in a fully relativistic setting. It is hence important to investigate the properties of this class of theories before proceeding further.

4.3.2. Structure and Formalism

The **Einstein Aether** (**EA**) theory was initially introduced by adding a scalar field to the Einstein-Hilbert action [41]. However, it was later reworked in [42] to to include a unit timelike vector field defining a preferred frame instead.[9]

The addition of a preferred frame defining a timelike direction leaves 3D space isotropic, but it breaks Lorentz invariance in the same way that an anisotropic space breaks rotational symmetry, because a Lorentz boost can be interpreted as a rotation around the time axis. This characteristic of the theory makes it possible to falsify its predictions with experimental data on the validity of Lorentz invariance, which so far has held true to high precision. As stated above, the EA theory has an action which includes the Einstein-Hilbert action and extra terms characterising the novel unit timelike vector field, conventionally named A^μ. The action for the EA theory described in [44] can be given as

[8] As previously explained, the disagreement with observational data only occurs when the two metrics present are non conformally related.

[9] The timelike unit vector field is named aether. It is different from the field disproved by the Michelson Morley experiment [43], as it defines a preferred time, not spatial direction, and it is hence consistent with special relativity.

follows, using the notation u^μ for the vector field:

$$S_{EA} = \frac{1}{16\pi G} \int \sqrt{-g}\, L_{EA}\, \mathrm{d}^4 x. \tag{4.3.8}$$

The Lagrangian has the form:

$$L_{EA} = \overbrace{R}^{A} + \overbrace{c_1 \left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) + c_2 \left(\nabla_\mu u^\mu\right)\left(\nabla_\nu u^\nu\right) + c_3 \left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right)}^{B} + \overbrace{\lambda \left(u_\mu u^\mu + 1\right)}^{C}. \tag{4.3.9}$$

The components can be identified as follows:

- A is the standard Ricci scalar, yielding the L.H.S. of the Einstein equations;
- B has the role of a kinetic term with free coefficients c_i, equivalent to χ in CEG;
- C is a Lagrange multiplier which forces the vector field u^μ to be unit timelike upon variation of the action w.r.t. λ. It has the same role as the a priori assumption of $u_\mu u^\mu = -\phi^2/a_0^2$ in CEG.

It can be seen that this family of theories belongs to the enlarged group of vector-tensor (VT) theories, of the form [45]:

$$L_{VT} = \overbrace{R + \omega u_\mu u^\mu R}^{\tilde{A}} + \overbrace{c_1 \left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) + c_2 \left(\nabla_\mu u^\mu\right)\left(\nabla_\nu u^\nu\right) + c_3 \left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right)}^{\tilde{B}} + \overbrace{\lambda \left(u_\mu u^\mu + 1\right)}^{\tilde{C}}, \tag{4.3.10}$$

which clearly reduces to (4.3.9) in the special case of no direct coupling of the vector field u^μ to the Ricci scalar R, with $\omega = 0$. In constrained theories such as EA, a non-zero ω could be replaced by a rescaling of the Newton constant G, as $\tilde{G} = G/\left(1-\omega\right)$.

A modification to (4.3.9) which allows non-canonical kinetic terms in the action is of particular interest, as TeVeS was shown to belong to this Generalised Einstein Aether (GEA) group . This specific case was extensively treated in [46], and its most general form is given as:

$$L_{GEA} = \frac{1}{16\pi G}\left(M^2 \mathcal{F}\left(\mathcal{K}\right) + \lambda \left(u_\mu u^\mu + 1\right)\right), \tag{4.3.11}$$

where the following should be noted:

- a new constant M of the dimension of mass was introduced;
- $\mathcal{K}$ is a rescaled version of the kinetic term present in L_{EA} of (4.3.9), $\mathcal{K} = B/M^2$;
- $\mathcal{F}\left(\mathcal{K}\right)$ is an arbitrary function of the kinetic term.

As shown in [46], the field equations (corresponding to the Einstein Equations in GR) are given as:

$$G_{\alpha\beta} = \tilde{T}_{\alpha\beta} + 8\pi G T_{\alpha\beta}, \tag{4.3.12}$$

where $G_{\alpha\beta}$ and $T_{\alpha\beta}$ are the Einstein tensor and matter energy-momentum tensor from GR respectively. This reflects the presence of a single metric tensor, and the coupling of matter is to this metric only, and not directly to the vector field u^μ. The new term, $\tilde{T}_{\alpha\beta}$ is the energy-momentum tensor of the vector field, and is given as:

$$\tilde{T}_{\alpha\beta} = \frac{1}{2}\nabla_\sigma \left[\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\left(J_{(\alpha}{}^{\sigma}u_{\beta)} - J^{\sigma}{}_{(\alpha}u_{\beta)} - J_{(\alpha\beta)}u^{\sigma)}\right)\right] - \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}Y_{(\alpha\beta)} + \frac{1}{2}g_{\alpha\beta}M^2\mathcal{F} + \lambda u_\alpha u_\beta. \tag{4.3.13}$$

The bracket notation for the indices indicates the symmetric component of a (combination of) tensor(s), e.g.:

$$J_{(\alpha}{}^{\sigma}u_{\beta)} = \frac{1}{2}\left(J_\alpha{}^{\sigma}u_\beta + J_\beta{}^{\sigma}u_\alpha\right). \tag{4.3.14}$$

An auxiliary tensor was introduced as:

$$J^{\alpha}{}_{\sigma} = \left(\mathcal{K}^{\alpha\beta}{}_{\sigma\gamma} + \mathcal{K}^{\beta\alpha}{}_{\gamma\sigma}\right)\nabla_\beta A^\gamma, \tag{4.3.15}$$

and $\mathcal{K}^{\alpha\beta}{}_{\sigma\gamma}$ is the tensor defining the coefficients for the kinetic term $\mathcal{K}$ through:

$$M^2\mathcal{K} = \mathcal{K}^{\alpha\beta}{}_{\sigma\gamma}\nabla_\alpha u^\gamma \nabla_\beta u^\sigma = c_1\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) + c_2\left(\nabla_\mu u^\mu\right)\left(\nabla_\nu u^\nu\right) + c_3\left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right). \tag{4.3.16}$$

In the above, the dummy indices where replaced to match those of (4.3.9). Finally, $Y_{\alpha\beta}$ gives the functional derivative of the kinetic component $\mathcal{K}$ w.r.t. to the metric tensor:[10]

$$Y_{\alpha\beta} = \nabla_\sigma u^\eta \nabla_\gamma u^\xi \frac{\delta\left(\mathcal{K}^{\sigma\gamma}{}_{\eta\xi}\right)}{\delta g^{\alpha\beta}} = -c_1\left[\left(\nabla_\nu u_\alpha\right)\left(\nabla^\nu u_\beta\right) - \left(\nabla_\alpha u_\nu\right)\left(\nabla_\beta u^\nu\right)\right]. \tag{4.3.17}$$

The EOM for the vector field are obtained in [46] through varying the action w.r.t. u^α. The result is equivalent to directly calculating the EL equations, which, however, yields a clearer form if one intends to mimic the structure of the CEG analysis carried out in previous chapters, thus allowing one to make considerations on the symmetry of quantities built from the derivatives of the vector field:

$$\nabla_\mu\left(\frac{\partial L}{\partial\left(\nabla_\mu u^\nu\right)}\right) = \frac{\partial L}{\partial u^n}. \tag{4.3.18}$$

[10] To be exact, the functional derivative is again taken w.r.t. the inverse metric tensor, since as usual the field equations (4.3.12) are obtained by varying the action w.r.t. the inverse metric.

4.3.3. Including the Constraints from GW170817

A separate observation must be made before (4.3.18) is calculated explicitly: as shown in [47], strong restrictions are put on GEA theories based on GW170817. Specifically, the measured time delay in arrival between EM and gravitational radiation of 1.7s places a strict upper bound on the propagation speed of gravitational waves. Using units with a constant, unit speed of light, $c = 1$, the speed of gravitational waves c_T is parametrised as:

$$c_T = 1 + \alpha_T, \quad \alpha_T = 2\frac{\Delta t}{d_s} \tag{4.3.19}$$

where $d_s \approx 40\,\mathrm{Mpc}$ is the estimated distance to the source, and $\Delta t \approx 1.7s$ is the delay in arrival time. With these values, one then has the constraint:

$$|\alpha_T| \lessapprox 10^{-15}. \tag{4.3.20}$$

To excellent approximation then, models should satisfy the condition $\alpha_T = 0$. For a GEA model, this condition implies:

$$\alpha_T = -\frac{\left(c_1 + c_3\right)\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}}{\left[1 + \left(c_1 + c_3\right)\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\right]} = 0, \tag{4.3.21}$$

which, assuming the free function $\mathcal{F}(\mathcal{K})$ is in fact not a constant, implies on a Minkowski background:

$$c_1 = -c_3. \tag{4.3.22}$$

When enforcing (4.3.22), the kinetic term becomes:

$$\mathcal{K} = c_1\left[\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) - \left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right)\right] + c_2\left(\nabla_\mu u^\mu\right)\left(\nabla_\nu u^\nu\right) \tag{4.3.23}$$

where, in analogy with the EM strength tensor, one can introduce the quantity:

$$F_{\mu\nu} = \nabla_\mu u_\nu - \nabla_\nu u_\mu, \tag{4.3.24}$$

whose contraction,

$$F_{\mu\nu}F^{\mu\nu} = 2\left[\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) - \left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right)\right] \tag{4.3.25}$$

simplifies the allowed form for the kinetic term:

$$\mathcal{K} = \frac{c_1}{2}F_{\mu\nu}F^{\mu\nu} + c_2\left(\nabla_\alpha u^\alpha\right). \tag{4.3.26}$$

The LHS of (4.3.18) can now be rewritten as:

$$\nabla_\mu\left(\frac{\partial L}{\partial\left(\nabla_\mu u^\nu\right)}\right) = 2\nabla_\mu\left\{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\left[c_1 F^\mu_\nu + c_2\left(\nabla_\alpha u^\alpha\right)\delta^\mu_\nu\right]\right\}, \tag{4.3.27}$$

whereas the RHS of (4.3.18) simply gives:

$$\frac{\partial L}{\partial u^{\nu}} = \frac{\partial}{\partial u^{\nu}}\lambda\left(u_{\mu}u^{\mu}+1\right) = 2\lambda u_{\nu}, \tag{4.3.28}$$

yielding the full EOMs:

$$\nabla_{\mu}\left\{\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\left[c_1 F^{\mu}_{\nu} + c_2\left(\nabla_{\alpha}u^{\alpha}\right)\delta^{\mu}_{\nu}\right]\right\} = \lambda u_{\nu}. \tag{4.3.29}$$

This can easily be shown to be equivalent to the expression found in [46]

$$\nabla_{\alpha}\left(\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}J^{\alpha}_{\beta}\right) + \frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}y_{\beta} = 2\lambda u_{\beta}, \tag{4.3.30}$$

as one can express J^{α}_{β} as:

$$J^{\alpha}_{\beta} = 2\left[c_1 F^{\alpha}_{\sigma} + c_2\delta^{\alpha}_{\sigma}\left(\nabla_{\mu}u^{\mu}\right)\right] \tag{4.3.31}$$

and the functional derivative of the action w.r.t. u^{ν} is shown to vanish,

$$y_{\beta} = \nabla_{\sigma}u^{\eta}\nabla_{\gamma}u^{\xi}\frac{\delta\left(\mathcal{K}^{\sigma\gamma}{}_{\eta\xi}\right)}{\delta u^{\beta}} = 0. \tag{4.3.32}$$

4.3.4. The Weak Field, Non-Relativistic Limit

The constraints that were imposed on the EOMs were based on a Minkowski background. In the weak field limit, it is interesting to work in a Minkowski spacetime perturbed by a potential ϕ. Following [46], one can work in the Poisson Gauge, with a metric given by (it is important to note that the perturbation parameter ϵ is in no way related to the strain quantities of CEG described in previous chapters):

$$g_{\alpha\beta} = \eta_{\alpha\beta} + \epsilon h_{\alpha\beta}, \tag{4.3.33}$$

where the metric perturbation $h_{\alpha\beta}$ can be taken in the Poisson Gauge as:

$$h_{\mu\nu} = -2\phi\delta_{\mu\nu}. \tag{4.3.34}$$

The same approach is taken to expand the vector field around a lowest order unit timelike approximation, chosen as δ^{α}_{0}, which to first order in the same parameter ϵ gives:

$$u^{\alpha} = \delta^{\alpha}_{0} + \epsilon B^{\alpha}. \tag{4.3.35}$$

4.3.5. Connection to Emergent Gravity

In the previous chapter, we showed that the metric used by Verlinde (which explicitly analyses the nonrelativistic, spherically symmetric case) can be rewritten as a Minkowski background plus a perturbation of the same form of that used in [46] (see (3.3.29)). It therefore seems reasonable to assume that the displacement field 3-vector, which describes the modified potential that can be shown to reproduce MOND phenomenology, can be cast in a 4-vector form on the same background on which the Einstein-Aether perturbation is being analysed. In CEG, Hossenfelder arbitrarily gives the same normalisation to her 4-vector u^α and to that of the 3-vector appearing in [21], a choice that was shown to be inconsistent with a timelike vector in the previous chapter, where a lightlike normalisation was instead suggested.

Here, a different approach is taken. To identify the displacement 3-vector from [21] with the timelike vector u^α, we have to ensure that the normalisation is consistent on a perturbed Minkowski background. The task is simplified by the observation that the potential perturbing both spacetimes in [21] and [46] is the same as the one which appears in the 3-vector in Emergent Gravity and the unit timelike vector in GEA.

Before analysing the normalisation, it is important to be consistent with a unique system of units. Following the use of natural units from [21] and [26], we set speed to be dimensionless, energy to have inverse time dimension, and infer the other necessary physical units through the formula for the Compton wavelength, relating relativistic (the speed of light c) and quantum mechanical (the energy of a quantum of the EM field $\hbar$) scales through the equivalence in energy between the wavelength λ of a photon and the mass m of a massive particle:

$$\lambda = \frac{\hbar}{mc}. \tag{4.3.36}$$

When setting $c = \hbar = 1$ and taking mass as the fundamental dimension, the following can be inferred:

$$\lambda = \frac{1}{m} \rightarrow [\lambda] = [D] = [M]^{-1}. \tag{4.3.37}$$

It is then easy to see that the dimensions of acceleration (such as the acceleration scale a_0) are given as:

$$[a_0] = \left[\frac{c}{t}\right] = \left[\frac{1}{D}\right] = [M]\,, \tag{4.3.38}$$

implying that the mass scale can be interpreted as an acceleration scale. That said, it is interesting to note that the GEA Lagrangian (4.3.11) scales the kinetic term by a constant of the dimension of mass, M. Without modifying the theory, we can decide to absorb the mass constant into the field itself, by the substitution:

$$V^\mu = \frac{u^\mu}{M}, \tag{4.3.39}$$

which leads to a new, equivalent Lagrangian formulation in terms of the new field V^μ:

$$L_V = \frac{1}{16\pi G}\left(M^2\mathcal{F}\left(\mathcal{K}\left(V^\mu\right)\right) + \lambda\left(M^2 V_\mu V^\mu + 1\right)\right). \tag{4.3.40}$$

It is now clear that the EOM for λ yield a new constraint on V^μ:

$$V_\mu V^\mu = -\frac{1}{M^2}. \tag{4.3.41}$$

For future reference, the vector V^μ will be called the **dressed** potential, since it contains an additional prefactor when compared to the expression from [21], and the original u^μ with unit magnitude the **naked** potential as it lacks any of the additional prefactors of the dressed potential. The easiest way to define the correct dimension for the normalisation factor is of course to look at the dimensions of the spatial component of the displacement vector, which are of the form:

$$\left[u^i\right] = \left[\frac{\phi}{a_0}\right] = \left[\frac{\phi}{M}\right]. \tag{4.3.42}$$

As the gravitational potential is the energy per unit mass, with both quantities having the same dimension in natural units as $E = mc^2$, we can conclude that:

$$\left[u^i\right] = [M]^{-1}. \tag{4.3.43}$$

Therefore, it would be inconsistent to identify the 4-vector of the displacement vector as the naked potential. Instead, it has units coinciding with the dressed potential, as shown in (4.3.41), so it is consistent to make the identification:

$$V^i = \frac{\phi}{a_0} n^i, \tag{4.3.44}$$

where it is appropriate to now equate the two mass scales involved in the calculation:

$$a_0 = M. \tag{4.3.45}$$

An immediate result of (4.3.44) is that the spatial component of the naked potential can be given as:

$$u^i = a_0 V^i = \phi n^i. \tag{4.3.46}$$

The two cases which are now of most interest in analysing the consequences of the connection between the aether and the displacement field of [21] are:

- The full form of the field in a static Minkowski background;
- The behaviour of the field as a vector perturbation over a perturbed Minkowski background.

For the first case, the normalisation to match the constraint given by the Lagrange multiplier in (4.3.9) is straightforward:

$$u_\mu u^\mu = -\left(u^0\right)^2 + \left(u^i\right)^2 = -1 \rightarrow u^0 = \pm\sqrt{1+\phi^2}. \tag{4.3.47}$$

For the second case, perturbations around an already timelike vector δ_0^α as in (4.3.35), it is useful to look at the normalisation necessary for the sum of the zeroth and first order corrections to remain unit timelike:

$$A^\alpha = \delta_0^\alpha + \epsilon B^\alpha + \mathcal{O}\left(\epsilon^2\right) \rightarrow A_\alpha A^\alpha = -1. \tag{4.3.48}$$

Working out the above we have:

$$\left(\eta_{\mu\nu}+\epsilon h_{\mu\nu}\right)\left(\delta_0^\mu\delta_0^\nu+\epsilon\left(\delta_0^\mu B^\nu+\delta_0^\nu B^\mu\right)+\epsilon^2 B^\mu B^\nu\right)\approx\eta_{00}\left(1+2\epsilon B^0\right)+\epsilon h_{00}=-1, \tag{4.3.49}$$

from which it is implied that:

$$B^0=-\phi. \tag{4.3.50}$$

It is shown in [46] that the generalised Einstein equations (4.3.12) and the EOMs for the vector field (4.3.27) yield respectively:

$$2\nabla^2\phi-\left(c_1-c_3\right)\nabla\cdot\left(\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\nabla B^0\right)-\lambda=8\pi G\rho \tag{4.3.51}$$

$$2c_3\nabla\cdot\left(\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\nabla\left(\phi\right)\right)+2c_1\nabla\cdot\left(\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\nabla\left(\phi+B^0\right)\right)=-2\lambda. \tag{4.3.52}$$

By now plugging in the constraints from GW170817, $c_1=-c_3$ and (4.3.50), we can easily obtain a single PDE for the potential:

$$\nabla\cdot\left[\left(2+c_1\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\mathcal{K}}\right)\nabla\phi\right]=8\pi G\rho. \tag{4.3.53}$$

It is important to realise that the above can only be obtained if (4.3.50) holds. Therefore, it is important to ensure that this is, in fact, the case when the perturbation is given by the chosen naked potential. To this end, we can use the field normalised in Minkowski spacetime (as each element of u^μ is $\propto\phi$, the time component is the same in a perturbed Minkowski background up to first order in the perturbation), and set:

$$B^0=\pm\sqrt{1+\phi^2}\approx-\phi, \tag{4.3.54}$$

where the minus sign was chosen to obtain the correct normalisation with $B^0=-\phi$, and the spatial terms can be ignored as they do not enter (4.3.53).

$$u^\mu=\delta_0^m+\epsilon\left(-\phi,\ \phi n^i\right). \tag{4.3.55}$$

Computing the magnitude yields:

$$u_\mu u^\mu=-\left(1+2\epsilon\phi\right)\left(1-\epsilon\phi\right)^2+\left(1-2\epsilon\phi\right)u_i u^i, \tag{4.3.56}$$

and discarding nonlinear terms in ϵ we obtain:

$$-\left(1+2\epsilon\phi\right)\left(1-2\epsilon\phi\right)+\mathcal{O}\left(\epsilon^2\right)=-1+\mathcal{O}\left(\epsilon^2\right), \tag{4.3.57}$$

which shows that, provided that $\phi^2\gg1$, it is consistent to use the lightlike approximation of the naked potential as the perturbation of the timelike field. In [46], the correct limits for both MONDian and Newtonian gravity can only be retrieved if we accept that M is of the same order of magnitude as a_0.

The above calculations have shown that M and a_0 are in fact the same quantity, and have given a solid theoretical foundation to the Einstein-Aether theory, as well as a truly covariant expansion for both MOND and Emergent Gravity. However, it should be noted that the a_0 utilised in the above calculations is not equal to Milgrom's constant a_M, but related to it through $a_0 = 6a_M$. The Lagrangian for GEA with the dressed potential and $M = a_0$ can ultimately be given as:

$$\boxed{L_V = \frac{1}{16\pi G}\left({a_0}^2\mathcal{F}\left(\mathcal{K}\left(V^\mu\right)\right) + \lambda\left({a_0}^2 V_\mu V^\mu + 1\right)\right), \quad V^i = \frac{\phi}{a_0}n^i} \tag{4.3.58}$$

Chapter 5

Numerical MOND and Galaxy Clusters

5.1. Baryonic and Apparent Mass

When inferring the mass distribution from a gravitational potential, we must solve the PDE which models the system. In the Newtonian case, this PDE is the Poisson equation for gravity, whereas for MOND it is the non-linear MOND PDE. For the case of galaxy clusters in Newtonian gravity, the key assumption is that the total mass distribution ρ_T is given by the sum of the baryonic contribution ρ_B and a dark matter contribution ρ_D. Then, the Poisson equation for the Newtonian potential ϕ_N reads:

$$\boxed{\nabla \cdot (\nabla \phi_N) = 4\pi G \rho_T = 4\pi G \left(\rho_B + \rho_D\right).} \tag{5.1.1}$$

One can integrate eq. 5.1.1 to obtain the mass contained within a finite volume. For a spherical volume, which is of interest for galaxy clusters, the integration over the domain Ω yields:

$$\int_\Omega \nabla \cdot (\nabla \phi_N) \ \mathrm{d}\Omega = \int_\Omega 4\pi G \left(\rho_B + \rho_D\right) \ \mathrm{d}\Omega. \tag{5.1.2}$$

The RHS is proportional to the total mass, whereas the LHS can be simplified using the divergence theorem. Denoting the spherical surface bounding the domain by S, $\hat{n}$ the unit normal vector, M_B and M_D the masses of the baryonic and dark matter contained in the volume respectively, eq. 5.1.2 results in:

$$\int_S \nabla \phi_N \cdot \hat{n} \, \mathrm{d}S = 4\pi G \left(M_B + M_D\right). \tag{5.1.3}$$

Plugging in the the normal unit vector for the spherical surface $\hat{n} = \hat{r}$, and the surface element $\mathrm{d}S = r^2 \sin(\theta) \, \mathrm{d}\theta \, \mathrm{d}\phi$, eq. 5.1.3 becomes:

$$r^2 \int_S \frac{\partial \phi_N}{\partial r} \sin(\theta) \ \mathrm{d}\theta \, \mathrm{d}\phi = 4\pi G \left(M_B + M_D\right). \tag{5.1.4}$$

The following important observation can then be made:

In Newtonian gravity, the mass contained in a spherical volume is uniquely determined by the behaviour of the potential on the surface bounding the volume.

For a spherically symmetric mass distribution, both the potential and gravitational acceleration $g_N = \partial\phi_N/\partial r$ are also spherically symmetric. For a mass distribution contained within a radius R, eq. 5.1.5 simplifies to:

$$4\pi g_N r^2\Big|_{r=R} = 4\pi G\left(M_B + M_D\right). \tag{5.1.5}$$

When no mass is present for $r > R$, the gravitational acceleration at any point outside the mass distribution is:

$$g_N\left(r\right) = \frac{G\left(M_B + M_D\right)}{r^2}. \tag{5.1.6}$$

Eq. 5.1.6 describes the same acceleration as the one produced by a point source. This statement is part of the **Shell Theorem**, and it implies the following:

The gravitational acceleration outside of a spherically symmetric mass distribution with total mass M_T is indistinguishable from the acceleration generated by a point source of mass M_T located in the center of the distribution.

On the other hand, in MOND, the mass distribution present in galaxy clusters is exclusively made up of baryons. The PDE for the MOND potential ϕ_M, where $\mu\left(x\right)$ is an arbitrary interpolation function satisfying the correct asymptotic behaviour, is expressed as:

$$\nabla\cdot\left[\mu\left(\frac{|\nabla\phi_M|}{a_0}\right)\nabla\phi_M\right] = 4\pi G\rho_B. \tag{5.1.7}$$

Following the same procedure as for the Newtonian case, the baryonic mass contained in a spherical volume can be found by integration:

$$\int_\Omega \nabla\cdot\left[\mu\left(\frac{|\nabla\phi_M|}{a_0}\right)\nabla\phi_M\right]\,\mathrm{d}\Omega = 4\pi G M_B. \tag{5.1.8}$$

After applying the divergence theorem this yields:

$$r^2\int_S \mu\left(\frac{|\nabla\phi_M|}{a_0}\right)\frac{\partial\phi_M}{\partial r}\,\sin\left(\theta\right)\,\mathrm{d}\theta\,\mathrm{d}\phi = 4\pi G M_B. \tag{5.1.9}$$

As in Newtonian gravity, the total mass enclosed in the spherical volume is defined by the behaviour of the potential on the boundary. If the mass distribution is spherically symmetric, this reduces to:

$$\mu\left(\frac{1}{a_0}\frac{\partial\phi_M}{\partial r}\right)\frac{\partial\phi_M}{\partial r}r^2\Bigg|_{r=R} = G M_B. \tag{5.1.10}$$

In the case of galaxy clusters, points with $r \geq R$ are in the deep MOND regime, and on the boundary the interpolation function is approximately equal to its asymptotic value, $\mu(x) \approx x$. When no mass is present for $r > R$, eq. 5.1.10 becomes:

$$\frac{1}{a_0}\left(\frac{\partial \phi_M}{\partial r}\right)^2 r^2 = GM_B. \tag{5.1.11}$$

The gravitational acceleration $g_M = \partial \phi_M / \partial r$ is then given by:

$$\boxed{g_M = \frac{\sqrt{GM_B a_0}}{r}.} \tag{5.1.12}$$

The acceleration in eq. 5.1.12 is equivalent to the acceleration produced by a point source of mass M_B in the deep MOND limit, as derived in eq. 1.4.30 of Chapter 1. However, to obtain this result it was assumed that the boundary of the domain was in the deep MOND regime. Instead, if this is not the case, eq. 5.1.10 becomes:

$$g_M = \frac{GM_B}{r^2}\left[\mu\left(\frac{g_M}{a_0}\right)\right]^{-1} \neq \frac{\sqrt{GM_B a_0}}{r}. \tag{5.1.13}$$

This results in a restriction on the use of the shell theorem in MOND:

> In MOND, the shell theorem can only be applied outside a spherical domain containing a mass M_B if the boundary of the domain is in the deep MOND regime. In this case, the acceleration is the same as that produced by a point source in the deep MOND regime with the same mass M_B, at the center of the mass distribution.

The above statement implies that the mass contained in a volume, the boundary of which is not in the deep MOND regime, depends on the interpolation function chosen and on the value of the acceleration at the boundary. This also holds for spherically symmetric distributions. In the next section we will treat the phenomenon of gravitational lensing, to show that the assumption that the gravitational potential is always Newtonian can lead to a form of apparent matter which is non-physical, named **Phantom Dark Matter** (**PDM**).

5.2. Gravitational Lensing and Phantom Dark Matter

5.2.1. Lensing on a Flat Background

The data gathered from gravitational lensing measurements only gives an indirect measure of the mass present in systems such as galaxies and galaxy clusters. In general, gravitational lensing is considered to be an effect of the perturbation of a Minkowski background by the Newtonian potential, with the spacetime interval:

$$\mathrm{d}s^2 = -\left(1 + 2\phi_N\right)\mathrm{d}t^2 + \left(1 - 2\phi_N\right)\left(\mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2\right). \tag{5.2.14}$$

Here we denote the Minkowski metric by $\eta_{\mu\nu}$ and the perturbation by $h_{\mu\nu}$. The metric from eq. 5.2.14 can be decomposed as:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}. \tag{5.2.15}$$

This implies that the perturbation is a 2^{nd} rank tensor fully defined by its trace, namely, a diagonal matrix of the form:

$$h_{\mu\nu} = -2\phi_N \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1. \end{pmatrix} \tag{5.2.16}$$

Following the treatment given in [6], we make the assumption that the perturbation is small enough that the geodesics of the perturbed spacetime can be calculated by a slight deviation from the geodesic of the Minkowski spacetime, which are straight lines. The perturbed geodesic is $x^{\mu}\left(\lambda\right)$, with respect to an affine parameter λ. The Minkowski geodesic is $x^{(0)\mu}\left(\lambda\right)$ and the geodesic perturbation is $x^{(1)\mu}\left(\lambda\right)$. We then have:

$$x^{\mu}\left(\lambda\right) = x^{(0)\mu}\left(\lambda\right) + x^{10)\mu}\left(\lambda\right). \tag{5.2.17}$$

Gravitational lensing considers the paths of photons, which are geodesics, for which the spacetime interval vanishes. For the interval of eq. 5.2.14, this implies:

$$\mathrm{d}s^2 = g_{\mu\nu}\frac{\mathrm{d}x^{\mu}}{\mathrm{d}\lambda}\frac{\mathrm{d}x^{\nu}}{\mathrm{d}\lambda} = 0. \tag{5.2.18}$$

It is useful to denote the wave vectors for the Minkowski spacetime and the perturbation:

$$k^{\mu} = \frac{\mathrm{d}x^{(0)\mu}}{\mathrm{d}\lambda}, \quad l^{\mu} = \frac{\mathrm{d}x^{(1)\mu}}{\mathrm{d}\lambda}. \tag{5.2.19}$$

With $\left(k^0\right)^2 = \left(\vec{k}\right)^2 = k^2$, eq. 5.2.18 yields, to first order in $x^{(0)\mu}$:

$$-kl^0 + \vec{l}\cdot\vec{k} = 2k^2\phi_N. \tag{5.2.20}$$

The analysis of the perturbed geodesic equation to first order yields expressions for the time and space components of the change of the wave vector perturbation w.r.t. the affine parameter λ. This represents the deviation from a Minkowski geodesic (which is the 0^{th} order approximation of eq. 5.2.18), and it reads:[1]

$$\frac{\mathrm{d}l^0}{\mathrm{d}\lambda} = -2k\left(\vec{k}\cdot\nabla\phi_N\right), \quad \frac{\mathrm{d}\vec{l}}{\mathrm{d}\lambda} = -2k^2\nabla_{\perp}\phi_N. \tag{5.2.21}$$

[1] To obtain these results, the Christoffels need to be calculated. In the interest of conciseness, we skip the details, which can be found in [6].

In the above expression the transverse gradient was introduced, which quantifies the gradient that is not parallel to the path:

$$\nabla_\perp \phi_N = \nabla \phi_N - \left(\vec{k} \cdot \nabla \phi_N\right) \vec{k}/k^2. \tag{5.2.22}$$

Two steps are now required to find the spatial deviation of the geodesic from a straight line. First, l^0 can be obtained by integrating the expression in eq. 5.2.21 w.r.t. the affine parameter λ:[2]

$$l^0 = \int_\lambda -2k \left(\vec{k} \cdot \nabla \phi_N\right) \mathrm{d}\lambda = -2k\phi_N. \tag{5.2.23}$$

Following the same procedure to find the change in $\vec{l}$ yields:

$$\Delta \vec{l} = -2k^2 \int_\lambda \nabla_\perp \phi_N \, \mathrm{d}\lambda \tag{5.2.24}$$

By inserting eq. 5.2.23 into eq. 5.2.20, it can also be seen that spatial component of the perturbation to the wave vector is normal to the spatial component of the background geodesic:

$$\vec{l} \cdot \vec{k} = 2k^2\phi_N + kl^0 = 0. \tag{5.2.25}$$

Ultimately, the quantity of interest is the deflection of photons due to the gravitational potential ϕ_N. This is measured by the deflection angle $\hat{\alpha}$. Identifying the spatial length s of the path along which the angle is measured as $s = k\lambda$, the deflection angle is computed as follows:

$$\hat{\alpha} = -\frac{\Delta\lambda}{k} = 2 \int_s \nabla_\perp \phi_N \, \mathrm{d}s. \tag{5.2.26}$$

5.2.2. Cosmological Backgrounds

The discussion so far has assumed that the background metric is given by Minkowski spacetime. However, the actual study of gravitational lensing must take into account the fact that, according to observations, the universe is expanding. We again will follow the notation from [6], for which the following concepts must be defined:

- At any given moment in time, the spacetime describing the universe is, spatially, maximally symmetric. This means that it is spatially homogeneous and isotropic: space looks the same everywhere and in every direction.
- The universe evolves with time, with the spatial distance between objects varying accordingly.

[2] The integration constant is set to zero, as one uses the fact that the deviation will vanish when the potential is 0, $\phi_N = 0 \implies l^0 = 0$.

- Introducing the space interval in 3D space as $\mathrm{d}\sigma^2$, and the function determining its variation over time as $R(t)$, the two previous statements give rise to metrics of the form:
$$\mathrm{d}s^2 = -\mathrm{d}t^2 + R^2(t)\,\mathrm{d}\sigma^2. \tag{5.2.27}$$
It is evident that, for a given value of t, the spatial component of the spacetime can be made homogeneous and isotropic by an appropriate choice of $\mathrm{d}\sigma^2$. At the same time, spatial distances will increase (decrease) over time for an increasing (decreasing) $R(t)$, which is called the **scale factor**.

- The curvature of the 3D space defined by $\mathrm{d}\sigma$ is described by the Riemann tensor $\tilde{R}_{ijkl}$.[3] In section 1.5 it was stated that the Riemann tensor can be decomposed into a trace component, the Ricci tensor $\tilde{R}_{ij}$, and a traceless component, the Weyl tensor C_{ijkl}. For a 3D manifold, the Weyl tensor always vanishes [6], so the curvature is fully described by the Ricci tensor. The Ricci scalar $\tilde{R} = \tilde{R}^i_i$ can then be used to quantify the curvature. It is useful to introduce the scaled curvature scalar $k = \tilde{R}/6$.

- The scaled curvature k can be normalised to the values $k \in \{+1, 0, -1\}$. The sign of k determines the curvature of the space: $k = -1$ defines a space with constant negative curvature, in which initially parallel geodesics will diverge; $k = 0$ corresponds to flat space; $k = +1$ defines a space with constant positive curvature, where geodesics will converge.

- When rescaling k to the $\{+1, 0, -1\}$ range, the scale factor acquires the dimension of distance. It is useful to work with a dimensionless scale factor $a(t)/R_0$, where R_0 is a length scale. The curvature scalar is similarly redefined as $\kappa = k/R_0^2$, and so is the radial direction $\bar{r}$ of the spatial metric from $\mathrm{d}\sigma^2$, as $r = R_0\bar{r}$.

Taking into account all the observations made above, the metric describing a spatially isotropic and homogeneous universe is given by the **Friedmann-Lemaitre-Robertson-Walker** (**FLRW**) metric, with spacetime interval:

$$\mathrm{d}s^2 = -\mathrm{d}t^2 + a^2(t)\left[\frac{\mathrm{d}r^2}{1-kr^2} + r^2\mathrm{d}\Omega\right]. \tag{5.2.28}$$

In eq. 5.2.28 the scale factor $a(t)$ does not yet have an explicit form, which is instead fixed by the Einstein equations for a given energy distribution. By postulating the energy content of the universe at different cosmological times (for example, from immediately after the Big Bang to the modern day), an expression can be obtained for $a(t)$, that makes it possible to define the spatial distances between points in spacetime for a given value of t. Therefore, once $a(t)$ has been found, one can utilise the FLRW metric as a background metric.

[3] We use $\tilde{R}_{ijkl}$ rather than R_{ijkl} to avoid confusion between the resulting Ricci scalar $\tilde{R}$ and the scale factor $R(t)$.

Naturally, the treatment of gravitational lensing will be different when considering the FLRW background rather than Minkowski spacetime. Gravitational lensing on an FLRW background is generally treated through the weak lensing formalism, where the assumption is made that the gravitational potential ϕ_N originates from a body of a size which is negligible when compared to the distance D_{LS} between the source and the lens, as well as the distance d_L between the lens and the observer. However, as explained in [24], the thin lens approximation cannot be applied to MOND due to the theory's inherent nonlinearity. For example, the deflection angle that would be produced by N equal bodies along a line of sight would be $\sqrt{N}$ times larger than if the total mass belonged to a single object. Therefore, rather than deriving the relations for the deflection angles in the thin-lens approximatio, we give the expression for weak lensing on an FLRW background in terms of the observed angle θ_i and the unlensed angle β_i. The angles are 2D vectors which define the angular position of the source in the sky. The expression relating these two angles, as shown in [48], is:

$$\frac{\partial \beta_i}{\partial \theta_j} = \delta_{ij} + \int_r g\left(r\right) \frac{\partial^2 \phi_N}{\partial x^i \partial x^j}. \tag{5.2.29}$$

In eq. 5.2.29, $g\left(r\right)$ is the lensing kernel, which contains information about the mass distribution, but the exact form of which is not important for this discussion. In addition, x^i represents the 2D coordinates in the plane normal to the line of sight, and r is the distance between two points at a given time t, namely, the comoving distance, which corresponds to the scaled radial coordinate from the FLRW metric. The integral is calculated along the line of sight between the source and observer. Although a more in-depth discussion could certainly be carried out on weak lensing on an FLRW background, there is one key aspect to note:

> As in the case of a Minkowski background, lensing on an FLRW spacetime is determined by the Newtonian potential ϕ_N.

The mass distribution inferred by lensing measurements is always based on the assumption that gravity is Newtonian. This leads us to ask the important question: if MOND is correct, what mass distribution would be inferred from the observed potential, in the case that the potential was exclusively generated by baryonic matter? To answer this question, we recall that, both in the case of Newtonian gravity and MOND, the mass contained in a finite volume can be determined by the behaviour of the potential at the boundary of said volume. We previously pointed out that the boundaries of galaxy clusters are in the deep MOND regime. Since the mass distribution in clusters is generally concentrated in the central region [3], we can also to good approximation assume that the potential at the boundary resembles the potential generated by a spherical mass distribution. Therefore, if we make the (erroneous) assumption that the potential is Newtonian, and instead work with the MOND potential, we find, for the total Newtonian mass M_N contained in a spherical volume:

$$M_N = \frac{1}{4\pi G} \int_\Omega \nabla^2 \phi_M \; \mathrm{d}\Omega. \tag{5.2.30}$$

Applying the divergence theorem we find:

$$M_N = \frac{1}{4\pi G}\int_S \frac{\partial \phi_M}{\partial r}\mathrm{d}S. \tag{5.2.31}$$

Assuming that at the boundary, $r = R$, the potential is approximately that of a spherical mass distribution centered at the origin, $\phi_M \,|_{r=R} = \sqrt{GM_B a_0}\ln r$, where M_B is the baryonic mass generating the potential ϕ_M, we find:

$$\boxed{M_N = \frac{4\pi}{4\pi G}\frac{\sqrt{GM_B a_0}}{r}r^2\bigg|_{r=R} = \sqrt{\frac{a_0 M_B}{G}}R} \tag{5.2.32}$$

A number of important conclusions can be derived from the above expression:

- The inferred Newtonian mass grows linearly with the radius R of the volume over which integration is performed, even if the baryonic mass M_B becomes constant at a given radius.
- There is a critical radius r_c for which the inferred Newtonian mass M_N and the true baryonic mass M_B coincide. This radius is determined by the baryonic mass:

$$\boxed{M_N = M_B \implies r_c = \sqrt{\frac{GM_B}{a_0}}.} \tag{5.2.33}$$

- There is a corresponding critical surface mass density $\Sigma_m = a_0/G$ which determines the transition between the Newtonian and MOND regimes. The importance of Σ_m in the study of the rotation curves of galaxies in explained in [49].
- The total Newtonian mass includes the baryonic and dark components. Hence we can re-write:

$$M_B + M_D = \sqrt{\frac{a_0 M_B}{G}}R \implies M_D = \sqrt{\frac{a_0 M_B}{G}}R - M_B. \tag{5.2.34}$$

This implies that for radii $R < r_c$, the inferred mass of the dark matter will be negative, $M_D < 0$. Equivalently, we note the following: in the above expression, a negative value for M_D is obtained when $M_B < a_0/GR^2 = \Sigma_m R^2$. Therefore, negative mass should be observed for all systems with surface mass density $\Sigma < \Sigma_m$. Although this relation corresponds to the MOND regime, the radius R at which this phenomenon is measured should be larger than the radius of the system, as it is well known that the region close to the core of galaxies is in the Newtonian regime, and baryonic mass is dominant, so the effect could not be observed.

5.3. Simulation results

In the previous section, the analysis of the relationship between the apparent Newtonian mass M_N, phantom dark matter mass M_D and baryonic mass M_B was carried out under the assumption of spherical symmetry. However, in the case of galaxy clusters, spherical symmetry is only a good approximation at the boundary of the system, as the mass distribution generally includes a continuous component, the ICM, and a discrete component, given by the galaxies. In MOND, three important aspects cannot be treated analytically:

1. In order to compute the mass contained in a volume through the behaviour of the potential at the boundary, we have so far assumed that the boundary of the system is in the deep MOND regime. Given a potential, this allows us to infer the PDM contribution to the total mass. However, it is not true that the whole cluster is in the deep MOND regime. This means that we can not analytically compute the value of the potential at an arbitrary surface contained in the volume, whether this is centered at the origin or at another arbitrary location. Therefore, we are interested in analysing the behaviour for the integrated mass for spheres of increasing radii centered at the origin, both for the case of PDM and baryonic matter.

2. We have shown in the previous section that, at the critical radius r_c of a given system, the apparent mass equals the baryonic mass, implying that the PDM mass is 0. Therefore, we are interested in understanding how the relationship between baryonic mass and PDM changes when the galaxies embedded in the ICM. We want to explore how the PDM distribution increases around the galaxies, and determine what fraction of the total dark matter in the cluster is clumped around the galaxies, rather than following the gas distribution.

3. We showed in eq. 5.2.34 that it is possible for the PDM mass to take on negative values. This necessarily implies that there will be regions in which the PDM density transitions from positive to negative, crossing zero. We predict that this should happen close to the critical radius r_c of each galaxy. However, given that the gas component is non-negligible for all clusters analysed, we want to determine the regions in which the PDM has a negative mass distribution, and compare the results to the analytical case for two isolated masses which is provided in [50].

5.3.1. The Sample

In order to simulate a galaxy cluster, it is necessary to make a selection of observational data which can provide detailed information for both the ICM and galaxy components of the baryonic distribution. The search for valid results is made challenging in MOND by the fact that raw data is rarely available. Instead, the observations gathered by astronomers are processed according to a particular cosmological model, which almost always assumes that gravity is Newtonian at all scales. For this work, we required a sample that did not include any inferred dark matter contribution, and that could give a precise account of the distribution of the baryonic matter present in the clusters. The catalog by Reiprich and Bohringer [51] was used to obtain the ICM distribution. This catalog is particularly suitable since the data is obtained via X-Ray measurements rather than gravitational lensing, therefore giving an account of the radiation-emitting hot gas in the ICM without including inferred massed from weak lensing. Moreover, the catalog has been utilised in the study of modified gravity theories in [52], giving a further indication of its suitability for the study of alternative gravitational theories that do not predict the existence of non-baryonic dark matter. The ICM distribution is given as a **King β model**. This distribution makes the assumption that the gas is close to being in an isothermal state, and fully defines its distribution in 3D space through three parameters:

1. The central mass density ρ_0;
2. The critical gas radius r_k[4];
3. The dimensionless exponent β.

The distribution is spherically symmetric, and is given as:

$$\boxed{\rho\left(r\right) = \frac{\rho_0}{\left(1+\left(r/r_k\right)^2\right)^{-3\beta/2}}.} \tag{5.3.35}$$

However, the catalog given in [51] does not provide any information about the galaxy content of each cluster. Moreover, most galaxy mass data in the literature is given by including the assumed halo of dark matter surrounding the baryonic mass distribution. However, although the Reiprich catalog does not provide information on galaxy populations directly, it includes several clusters from the Abell catalog. The Abell catalog is a separate catalog of galaxy clusters which can complement the Reiprich catalog, as it gives information on the number of galaxies which belong to each cluster, by dividing the clusters into groups according to galactic population [53]. Certainly, it would be interesting to study every cluster, regardless of the number of galaxies contained within them. Nonetheless, as computation time is an important factor, we limited our study to the two less populous cluster groups, namely:

- Group 0, containing between 30 and 49 galaxies;
- Group 1, containing between 50 and 79 galaxies.

We found 15 galaxy clusters belonging to Groups 0 and 1 that were contained in the Reiprich catalog. Moreover, a fraction of these galaxies is gas dominated, whereas the rest is galaxy dominated, allowing us to test the difference in the behaviour of the PDM

[4] This should not be confused with the critical radius determining the volume in which the baryonic and apparent masses are equivalent for the case of a galaxy.

when the baryonic mass is more continuously or discretely distributed. The last step was to define a mass and size scale for the galaxies, as well as determining exactly how many galaxies to include in the simulation for each group. First, as there is an inherent uncertainty associated with the exact number of galaxies belonging to each cluster, due to the possibility of misclassification for both background and foreground galaxies, it was decided to assign the median value for both groups, namely, 40 galaxies for group 0 and 65 galaxies for group 1. Finally, it was decided to assign the same mass and size to each galaxy for either group. The best way to compute a mass range for galaxies without any assumption regarding dark matter is through the baryonic Tully-Fisher relation and rotation curve measurements, which can both consistently explained by MOND, based solely on baryonic matter. The Tully-Fisher relation, introduced in chapter 1, directly relates the asymptotic rotational speed of a galaxy to the baryonic mass contained within it:

$$M_b = \frac{V_r^4}{Ga_0}. \tag{5.3.36}$$

By analysing the SPARC collection of rotation curve data, we individuated $300km/s$ as a realistic value for our simulations and set the baryonic mass from eq. 5.3.36. Finally, we set the radius of each galaxy to $20kp$. For comparison, the Milky Way galaxy has an asymptotic rotational velocity of $V_r \approx 220km/s$, and hence a mass smaller than the the one of the galaxies used for the simulations, M_g, by a factor of ≈ 3.5. Moreover, the Milky Way radius is $\approx 30kp$, hence larger than the one of the galaxies in our simulation, R_g, by a factor of ≈ 1.5. Moreover, the galaxies are simulated as a 3D Gaussian pulse with a standard deviation of $\sigma = R_g/3$, so that 99.7% of the total mass M_g is contained within R_g. Furthermore, there is a reason why the chosen values for the galaxy mass and radius are interesting. For these values, the critical radius r_c, within which the apparent and baryonic masses are predicted to be equivalent, is only slightly larger than the galaxy radius R_g, by a factor of ≈ 1.3. This implies that we should be able to observe the formation of the PDM halo outside of each galaxy, as the critical radius is located outside of the baryonic mass distribution.

5.3.2. Integrated Mass Around the Center

The first phenomenon we analysed was the radial behaviour of the integrated mass for the baryonic, PDM, and total apparent mass distributions. To study this property, we calculated the masses contained within 20 spheres centered at the origin, with radii r_s given by the sequence:

$$\boxed{r_s = \{r_t/n\}_{n=1}^{10}.} \tag{5.3.37}$$

The results are shown in fig. 5.3.1. As can be seen from fig. 5.3.1, all of the integrated masses are increasing functions. Moreover, for all the clusters that were simulated, there is a radius within which the integrated dark matter M_D is smaller than the baryonic mass M_B. However, we also see that, at the boundary, $M_D > M_B$ for all cases considered. The radius at which $M_D = M_B$ varies between clusters, and it reaches a maximum value of

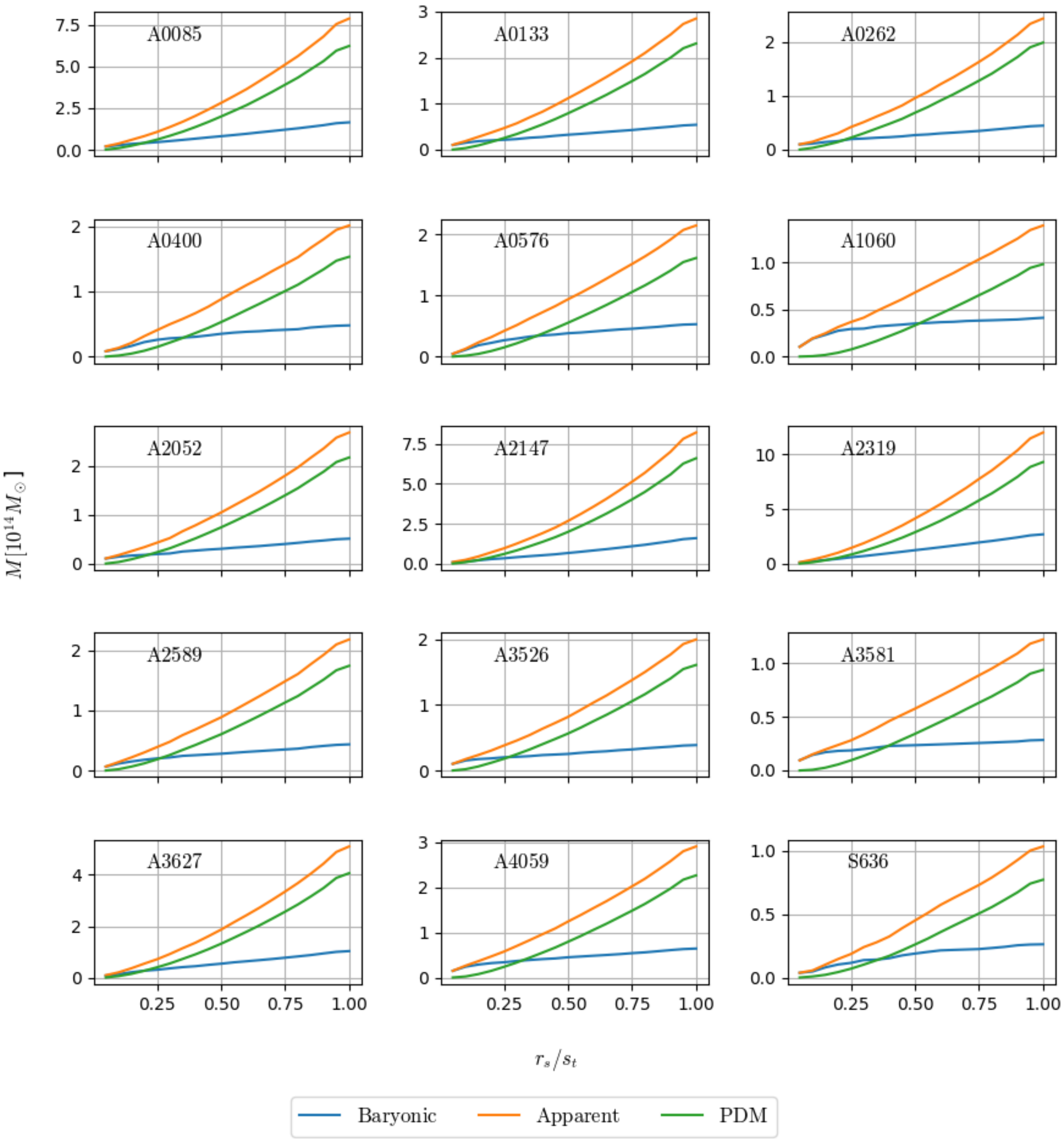

Figure 5.3.1: Integrated mass contained within spheres of successively larger radii around the origin. The blue, green and orange curves denote the baryonic, PDM and apparent masses respectively. The x axis gives the scaled domain size. The y axis gives the mass in 10^{14} solar masses. It can be seen that the baryonic mass still has a slope at the boundary of the domain. This occurs because the gas distribution is divergent, and grows linearly in the radial direction.

Name	$\rho_0\,(10^{-25}g/cm^3)$	β	$\mathrm{r}_k(kpc)$	$\mathrm{r}_{out}(kpc)$	Richness
A0085	0.34	0.532	58.4	2241	1
A0133	0.42	0.530	31.7	1417	0
A0262	0.16	0.443	29.6	1334	0
A0400	0.04	0.534	108.4	1062	1
A0576	0.03	0.825	277.5	1076	1
A1060	0.09	0.607	66.2	790	1
A2052	0.52	0.526	26.1	1373	0
A2147	0.03	0.444	167.6	2360	1
A2319	0.1	0.591	200.7	2657	1
A2589	0.12	0.596	83.1	1206	0
A3526	0.29	0.495	26.1	1175	0
A3581	0.31	0.543	24.6	840	0
A3627	0.04	0.555	210.6	1830	1
A4059	0.2	0.582	63.4	1324	1
S636	0.01	0.752	242.3	742	0

Table 5.1: List of the galaxy clusters simulated. All clusters belong to both the Reiprich and Abell catalogs. The richness comes from the Abell catalog, whereas the other parameters, which define the ICM distribution, originate from the Reiprich catalog. ρ_0 gives the central density of the ICM. β is a dimensionless exponent which appears in the ICM distribution. r_c and r_out are the critical radius and total radius of integration for the ICM respectively. The Richness column indicates the richness group each cluste belongs to: for richness 1, we simulated 65 galaxies, while for richness 0 we simulated 40.

$r_s = 0.5s_t$ for A1060. Both the total values of M_B and M_D clusters shown in fig. 5.3.1 vary over an order of mangitude between the different clusters. However, the radial scaling of each integrated mass is not considerably affected by the total mass contained in each cluster. From eq. 5.2.32, we know that the apparent mass grows as $M_A \propto \sqrt{M_B}R$ if the system is approximately spherically symmetric and is in the deep MOND regime at the integration boundary. Since we know that the baryonic distribution grows linearly with R, this implies that, in this case, the growth in M_A will be given by $M_A \propto R^{3/2}$. Deviation from this behaviour can be caused by the boundary of the sphere not being in the deep MOND regime, by the distribution not being approximately spherically symmetric, or both. We can see that, for the clusters A0400, A0576, A1060, A3581 and S636, the apparent mass grows, to good approximation, linearly. We can understand this behaviour by noting that these five clusters have the five smallest values for r_{out} in our sample. As r_{out} is the radius that bounds the cluster, we can see that it is more likely for these clusters that the deep MOND limit approximation on the boundary is not as good an approximation as it is for the other clusters. These five clusters also have very different values for the central mass density ρ_0 and galaxy population group amongst each other, so we see that the behaviour of the apparent mass is in this case not strictly tied to the nature of the baryonic distribution, but to the fact that the boundaries of these clusters are not precisely in the deep MOND regime.

Regardless of the total baryonic and apparent mass contained in the entire cluster, it is clear that all mass contributions scale similarly for different clusters (with the exceptions described above for the cluster with small bounding radii r_c). Therefore, to compare the behaviour of the clusters to each other more directly, we show the ratio between the dark and baryonic integrated masses M_D/M_B, again as a function of the distance from the origin, in fig.5.3.2. As can be seen in fig. 5.3.2, all clusters show a similar behaviour for the ratio M_D/M_B for low values of r_s/r_t. To make the analysis more quantitative, we point out that we can rewrite eq. 5.2.34 in the following way:

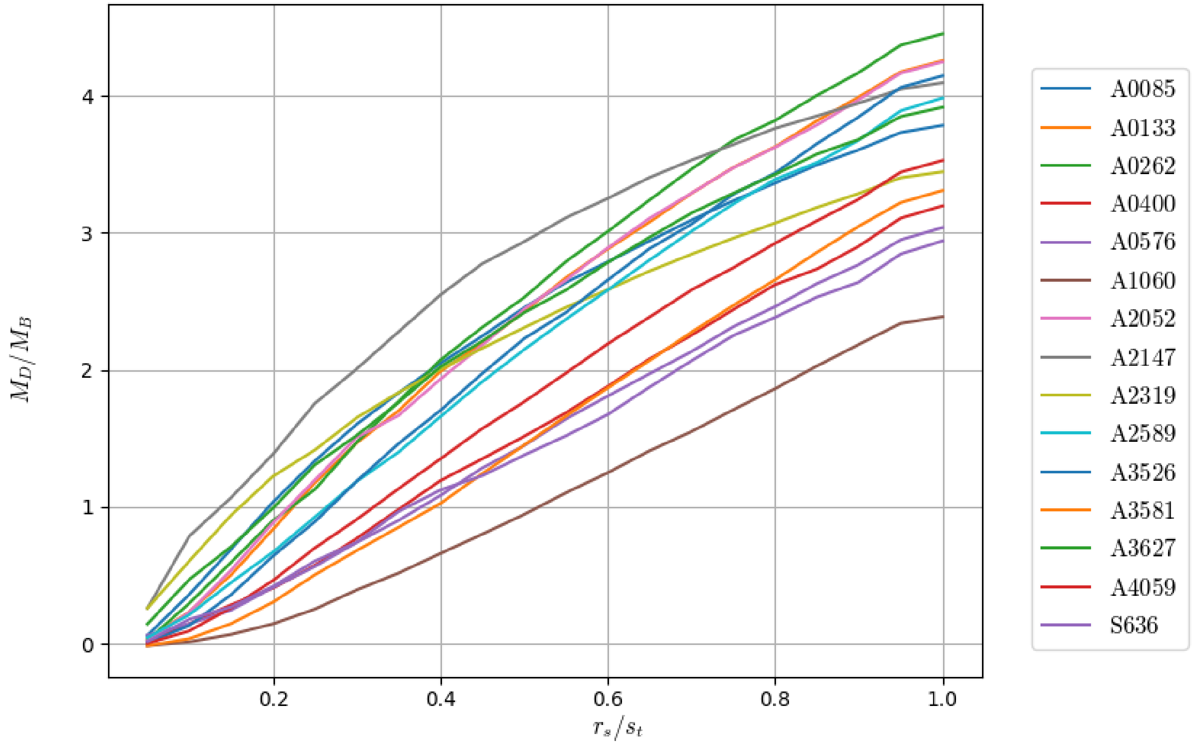

Figure 5.3.2: Ratio between the integrated mass for the dark mass and baryonic mass, for the integrated mass of the spheres centered at the origin. The x-axis gives the normalised radial distance, while the y-axis is the dimensionless ratio M_D/M_B. We see that the ratio grows linearly for clusters for which most of the domain is in not in the deep MOND regime, and grows with r_s/s_t

$$M_D/M_B = \sqrt{\frac{a_0}{GM_B}}R - 1. \tag{5.3.38}$$

However, the above expression only holds in the case in which the boundary of the sphere over which the volume is integrated is in the deep MOND regime. Since for smaller values of r the cluster is still in the Newtonian regime, this is not expected to hold. Moreover, in the central region, the abundance of galaxies implies that the mass distribution cannot be approximated by a spherical distribution. As can be seen from fig. 5.3.1, far from the central region the baryonic mass M_B grows approximately linearly. Therefore we expect that the integrated dark mass M_D will grow $\approx M_B/\sqrt{M_B} = \sqrt{M_B}$ away from the central region. We see that this is in fact the case for a number of clusters, although other clusters exhibit a linear growth at all values of r_s. This effect appears because the mass distribution is more spread out, implying that there is no spherical symmetry in the inner regions of the cluster. At the boundary, the ratio varies between $2.4 \leq M_D/M_B \leq 4.3$. Overall, we see that the ratio between PDM and baryonic matter does not follow the expected square root behaviour. Instead, in many clusters we see a linear growth in M_D/M_B, which is simply determined by the linear growth w.r.t. the radius r_s. Similarly to the result from fig. 5.3.1, this shows that if the system is not in the deep MOND regime, the usual square root dependence between M_D and M_B is no longer valid, and the behaviour of the PDM is no longer predictably related to the baryon density.

5.3.3. Integrated Mass Around the Galaxies

In the previous subsection we showed that, in general, the growth of the PDM mass w.r.t. the baryonic mass inside the clusters does not follow the square root relation, when the systems are not spherically symmetric and/or in the deep MOND regime. Another aspect of interest is the behaviour of the PDM around the galaxies. It was shown in [54] that recent measurements of the weak lensing in galaxy clusters implied a consistently larger number of small scale lensing effects when compared to simulations using the standard cosmological model with Newtonian gravity and cold dark matter. It is then of great interest to analyse the behaviour of the dark matter around the galaxies in the cluster. By doing so, we can see if the PDM can indeed form clumps around the galaxies. This would imply that a higher fraction of the PDM is concentrated away from the ICM, which in the majority of the clusters analysed accounts for most of the baryonic mass. To do so, we integrate the distribution of baryonic matter, PDM and total apparent matter in the neighbourhood of each galaxy, and add up all the contributions. The distance scale of interest for the concentration of PDM around the galaxies is the critical radius r_c. For the case of an isolated galaxy, the critical radius $r_c = \sqrt{GM_B/a_0}$ is the radius for which $M_A = M_B$. As all the galaxies simulated have the same mass and radius, we know that the critical radius is larger than the galactic radius r_g by a factor of ≈ 1.3. Therefore, the value of the integral at the smallest radius r_c should result in a PDM mass close to 0. More specifically, we evaluate the integrated masses in spherical volumes surrounding each galaxy, with the sphere radii r_s being multiples of the critical radius r_c:

$$\boxed{r_s = \{r_c \cdot n\}_{n=1,2,4,6,8,10}.} \tag{5.3.39}$$

First, we analyse the integrated masses for baryonic, PDM and apparent masses for all clusters. The results are shown in fig. 5.3.3. As shown in fig. 5.3.3, the baryonic mass

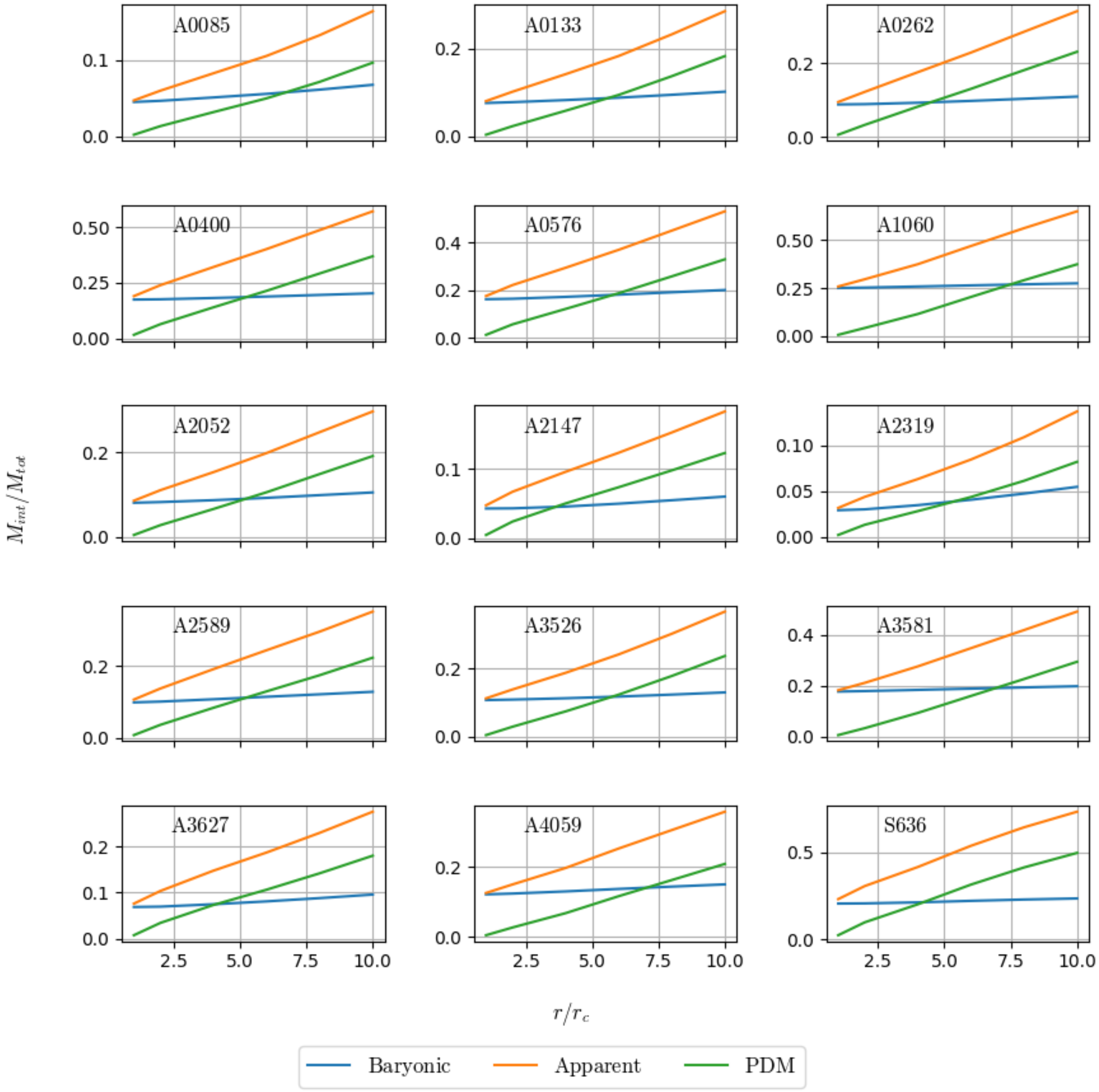

Figure 5.3.3: Baryonic, PDM and apparent integrated masses within the spheres around each galaxy. The x-axis gives the size of the spheres around each galaxy, as multiples of the critical radius of the galaxy. The y-axis gives the the integrated masses divided by the total apparent mass M_A contained in the cluster. Therefore, the y-axis is dimensionless. All curves start at the critical radius r_c, so that $r/r_c = 1$.

contained within increasing radii varies only slightly. This is to be expected, as the galaxies are generally separated from each other by distances which are large compared to their critical radius, and that the ICM surrounding them has a low mass density away from the central region. It can be seen that, for all clusters, the PDM density is close to zero at the critical radius, as expected. Moreover, the integrated PDM mass increases approximately linearly for all clusters. Instead, the baryon mass is approximately constant. A weak linear growth in the baryonic mass can be understood when we take into account that, for large values of r/r_c, we should also consider the ICM mass, which might give a non-negligible contribution. We also see that the radius at which the integrated PDM mass overtakes the baryonic contribution is always in the range $4r_c \leq r_s \leq 8r_c$. These results confirm that a vast portion of the PDM is contained in spheres around the galaxies, although the contribution only becomes dominant for $r_s > 4r_c$ in all cases. We can make another important observation when comparing fig. 5.3.3 to fig. 5.3.1. The apparent mass M_A grows linearly in the radius for all clusters. This can be understood by utilising eq. 5.2.32 for constant M_B. It is natural to assume that there is spherical symmetry around each galaxy, but the fact that eq. 5.2.32 gives such a precise fit tells us that, even in the presence of the external acceleration field generated by the ICM, the sphere contained within $r_s = 10r_c$ is in the deep MOND regime. This is in contrast to fig. 5.3.1, where we have established that the central region of the cluster is generally neither in the deep MOND regime, nor spherically symmetric. We have then established that the region around each galaxy is in general spherically symmetric and in the deep MOND regime. However, we should investigate whether the above result is affected by how much of the total baryonic and PDM mass of the cluster is contained within these spheres. Therefore, we plot the ratio of integrated mass to total, for both baryonic and PDM components, for the same radii as in fig. 5.3.3. This ratio can be seen in fig. 5.3.4. From fig. 5.3.4, we see that the fraction of the total baryon mass that is contained within a radius r_c around the galaxies ranges from a negligible amount of $\approx 13\%$, to the vast majority of the baryons in the cluster, at $\approx 85\%$. In addition, the baryons contained within the largest radius tested, $r_s = 10r_c$, also range from 25% to $\approx 92\%$. The variety in the mass contained within each sphere for different clusters stems from the fact that certain clusters have a high concentration of gas and galaxies in the central region, while others have broader distributions, due to which the mass captured in the shells around the galaxies only includes a negligible amount of the total ICM mass, and hence of the total baryon mass. On the contrary, the PDM distribution shows an approximately linear growth for all clusters w.r.t. the radius of the sphere r_s. For all clusters, at the critical radius r_c the PDM accounts for less than $\approx 3\%$ of the total PDM contained in the cluster, with clusters in which this value is lower than $\approx 1\%$. The clusters in which $\approx 3\%$ of the PDM is contained within r_c are those for which both gas and galaxies are concentrated in the central region, so that the PDM associated with the ICM is included within the smallest sphere. Moreover, the total PDM mass which is contained within $r_s = 10r_c$ varies from $\approx 10\%$ to $\approx 70\%$, with most clusters in the range between $\approx 20\%$ and $\approx 30\%$. Once again, we have then shown that the distribution of the PDM does not trivially follow that of the baryons, while instead it converges towards the discrete distributions, and that

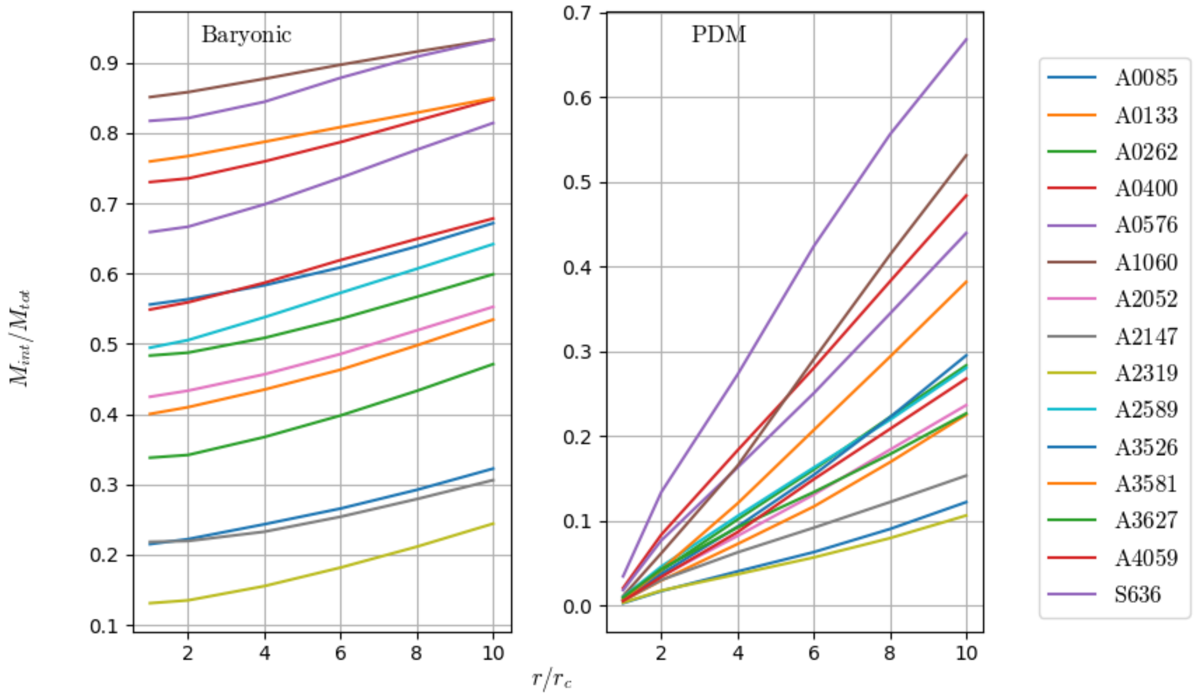

Figure 5.3.4: Fraction between the integrated and total mass, for baryonic (left) and PDM (right) mass distributions, contained in the spheres around the galaxies. The x-axis gives the radius of the spheres around each galaxy in units of the critical radius r_c. The y-axis gives the fraction of the integrated to total mass and is dimensionless. We see that the baryonic mass contained within the critical radius strongly varies, while the PDM is always close to zero within the critical radius r_c.

regardless of the baryon fraction contained in the spheres, the PDM is negligible close to each galaxy, as predicted in the previous section. Fig. 5.3.5 shows more clearly that the PDM distribution follows a common behaviour in all clusters, regardless of the width of the mass distribution which determines the density of the ICM and galaxies. As can be seen in fig. 5.3.5, the ratio between the baryonic mass and the total apparent mass within the spheres around the galaxies decreases approximately as $1/\sqrt{r_s}$. This is expected, since far from the galaxies we are in the deep MOND regime, and we can hence use the inverse of eq. 5.2.32. Furthermore, we see that the PDM mass becomes comparable to the baryonic mass between $r_s = 4r_c$ and $r_s = 8r_c$ for all clusters, regardless of the total fraction of baryons or PDM contained in any of the spheres. Although we have shown the relation between the baryonic and apparent integrated masses inside each of the shells, it is useful to analyse the ratio of the PDM to the baryons directly. This is shown in fig. 5.3.6. As can be seen in fig. 5.3.6, the PDM mass has a very similar integrated mass to that shown for the case of spheres centered at the origin, showing that the PDM distribution

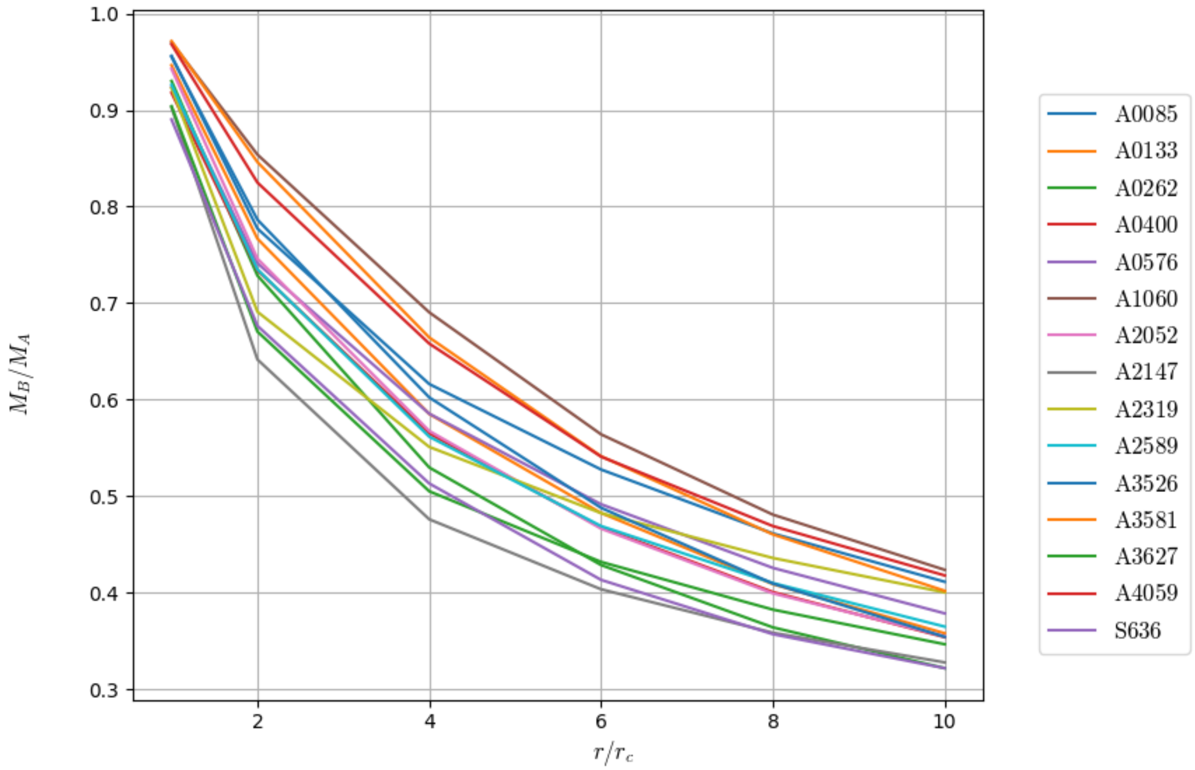

Figure 5.3.5: Ratio of the baryonic matter M_B contained in each shell around the galaxies to the apparent matter M_A contained in the same shell. The x-axis gives the radius of the spheres around each galaxy, scaled by the critical radius r_c, while the y-axis is a dimensionless fraction.

around the galaxies gives the dominant contribution to the total distribution of the PDM in each of the clusters. Up until this moment, we have made the assumption that the total contributions to baryonic, PDM and apparent masses are uniquely defined for a given spherical volume. Therefore, we have not tried to examine whether the baryonic mass we utilised is sufficient to account for all the dark matter contained in the cluster. This is because we are aware that our answer would not differ from that given in the literature when using the pristine formulation of MOND, which is equivalent to AQUAL in the case of spherical symmetry. However, there is one result that could theoretically be predicted without the need for a simulation, but is interesting nonetheless. This result is shown in fig. 5.3.7. From fig. 5.3.7, we clearly see that there are two linear scaling behaviours for the ratio M_D/N_B, one increasing and the other decreasing. These two behaviours correspond exactly to clusters where the baryonic mass distribution is either dominated by galaxies or by the ICM. However, this apparent relation does not depend on the MOND regime or the distribution being spherical. Instead, it is embedded in the baryonic fractions that

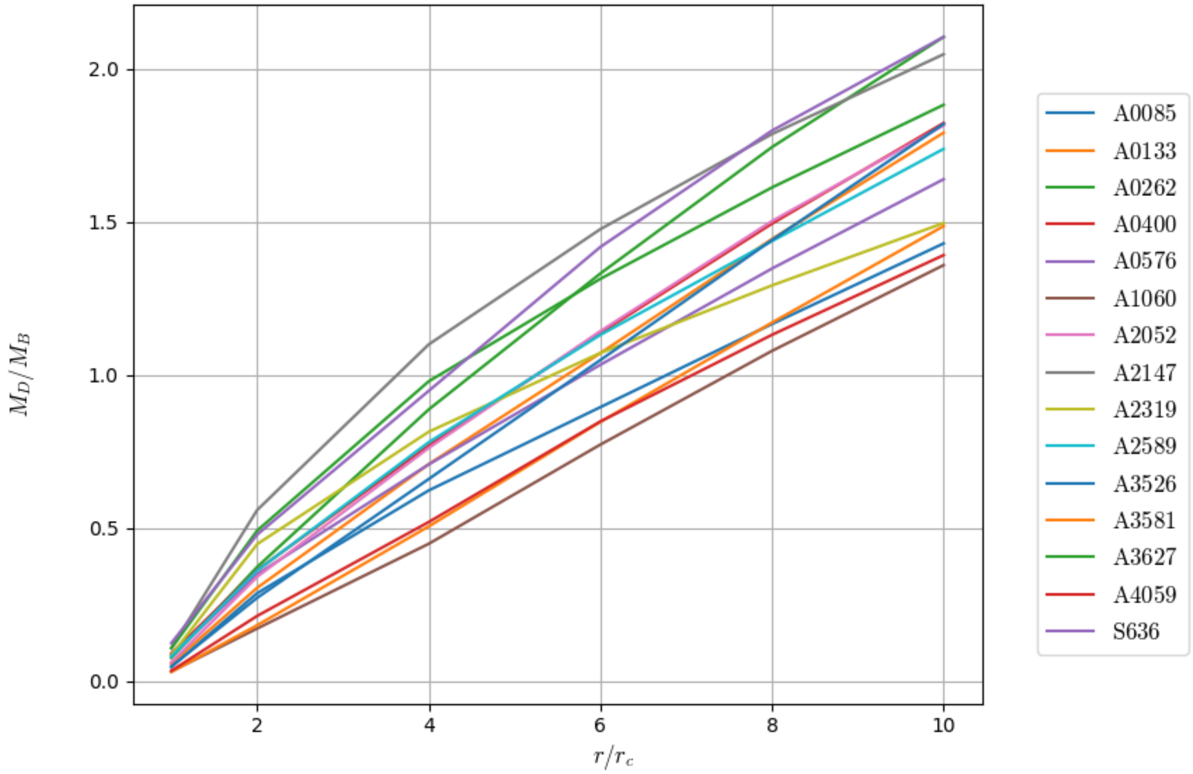

Figure 5.3.6: Ratio of dark matter to baryonic matter for the spheres around the galaxies. The x-axis gives the radius of the spheres around each galaxy, scaled by the critical radius r_c, while the y-axis is a dimensionless fraction.

we input into the simulation. Although we cannot fully account for the origin of this relationship, it is clear that the ratio between PDM and baryonic mass increases linearly as long as the galaxies make up most of the baryonic matter, and decreases linearly if the dominant component is the ICM.

5.3.4. Negative Mass Distributions

In general, the phenomena that can be described in the MOND framework through the exclusive use of baryonic matter can also be reproduced in standard cosmology by postulating an ad hoc matter distribution. Even the presence of small gravitational lenses inside of galaxy clusters, which we showed in the previous section to be an inevitable consequence of inferring an apparent distribution from a MOND potential, can be accommodated by simply postulating the presence of dark natter wherever needed. However, there is one phenomenon that cannot be explained by any widely adopted dark matter model: negative mass distributions. Of course, no such distribution would be inferred if the assumption is

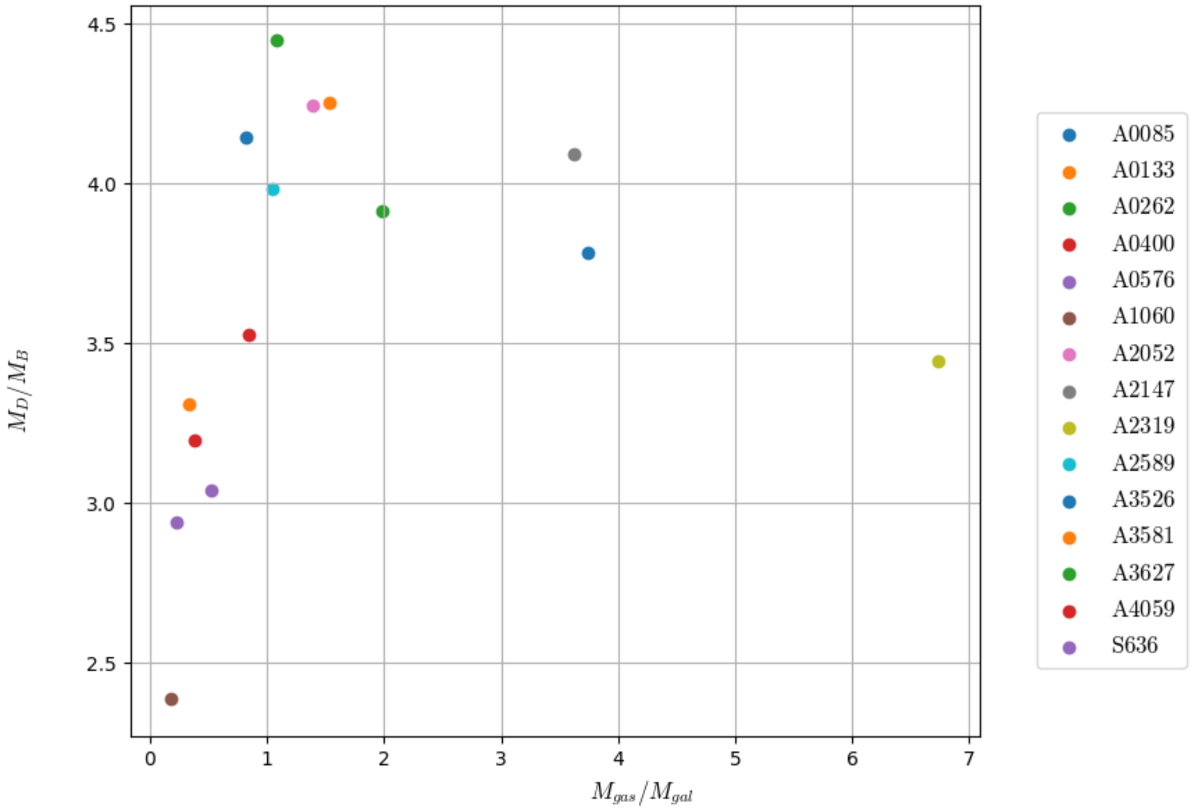

Figure 5.3.7: Dark-baryonic ratio against gas-galaxy ratio. All masses refer to the total masses integrated over the cluster. Each point corresponds to a different cluster, as given in the legend. There are clearly two different regimes, for which the ratio M_D/M_B increases (decreases) linearly. The two regimes meet where the baryonic mass is evenly split between gas and galaxies.

made that the potential measured is Newtonian. However, there is nothing that impedes the PDM stemming from a MOND potential, for a completely arbitrary baryonic mass distribution, from being negative at some points in space. A clear example can be taken from the existence of the critical radius r_c for a galaxy. If we assume that the baryonic mass distribution of the galaxy drops off extremely fast after a certain radius r_g, and that we have $r_c > r_g$, then for a given radius $r_g < r < r_c$ it is possible that there is a negative PDM component, since:

$$\boxed{M_D = \sqrt{\frac{a_0 M_B}{G}}R - M_B \to R < r_c = \sqrt{\frac{GM_B}{a_0}} \implies M_D < 0.} \tag{5.3.40}$$

The possibility of PDM with negative density has been considered by Milgrom in [50], where he analytically derived the geometry of a region of negative density around a discrete object of low surface density (equivalent to a radius smaller than r_c when using the definition for the critical surface density $\Sigma_m = a_0/G$ given after eq. 5.2.34) in an external acceleration field. We now want to make qualitative statements about the presence of negative mass PDM distributions in our simulations. Without loss of generality, as all clusters follow the same distribution with different parameters, we consider the cluster A0085. Its baryonic mass density is shown as a collection of isosurfaces, which are made transparent to show the galaxies within the ICM. The baryonic isosurfaces are shown in fig. 5.3.8: It can be seen that most of the galaxies are concentrated in the central region of the cluster, where the ICM mass density is highest. Nonetheless, every galaxy is in an external gravitational field pulling it towards the center of the ICM distribution. It is useful to now show an illustration from [50] depicting the predicted region of negative mass density. The illustration is in fig. 5.3.9: We now present the same situation as in fig. 5.3.8, except we now show the isosurface for a vanishing apparent mass, $M_A = 0$, implying a negative dark matter $M_D < 0$ in the region with positive baryon mass density: It can be seen from fig. 5.3.10 that the negative PDM accumulates around the galaxies exclusively, in the form of a torus. The approximation devised by Milgrom and shown in fig. 5.3.9 can hence be confirmed in the case of the external gravitational field generated by the ICM, with the galaxies acting as the sources. Due to the galaxies filling the gaps left by the negative PDM densities, it is not clear that the geometries shown in blue in fig. 5.3.10 actually correspond to tori. A clearer perspective, showing the predicted gaps in

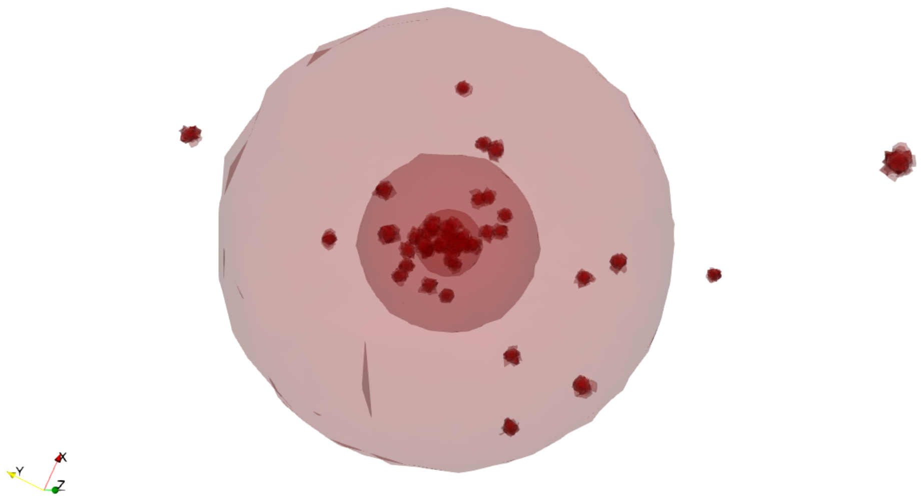

Figure 5.3.8: 3D view of the isosurfaces for the baryonic mass distribution for the A0085 cluster. For scale, each of the smaller surfaces gives the spatial extension of a galaxy inside the cluster. The larger isosurfaces around the origin represent the ICM distribution.

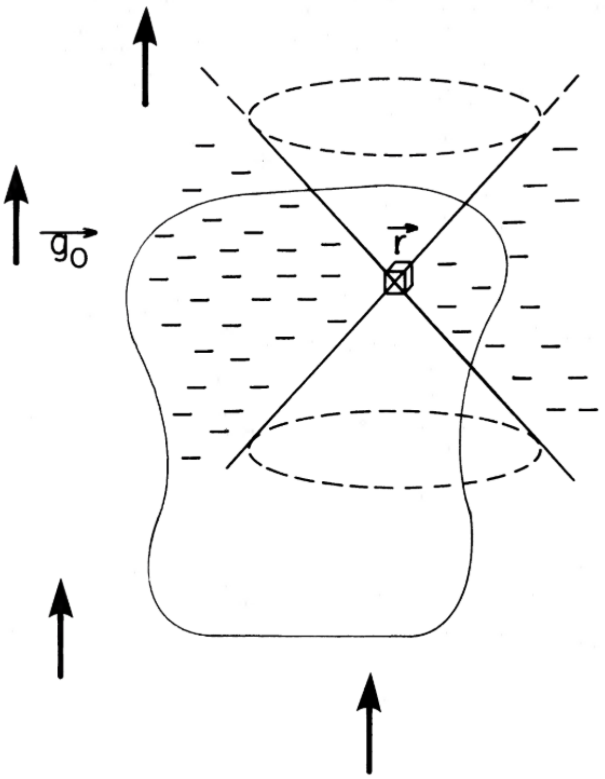

Figure 5.3.9: Body in an external gravitational field giving rise to a negative mass distribution, marked by the - signs to the sides of the extended cone geometry.

the negative PDM density in the direction of the external field lines, is given in fig. 5.3.11. We also show some isosurfaces for the positive apparent mass distribution, in order to give a more straightforward indication of the clumping effect that takes place close to the galaxies, after the critical radius r_c. From fig. 5.3.12 we see that, although some of the apparent mass density is present around the ICM, it is evident that each galaxy has a positive distribution surrounding it, starting after its critical radius r_c. Finally, we show the same system with all the isosurfaces for the separate contributions at once in fig. 5.3.13. From fig. 5.3.13, we can see that the galaxies are surrounded by a positive, spherical PDM distribution, and contained in the torus-shaped negative PDM distribution further away from its center. As a smooth transition is required between the PDM having a positive and negative distributions, the green and blue contributions in fig. 5.3.13 transform into each other, and become equivalent close to r_c for every galaxy, where $M_D = 0$.

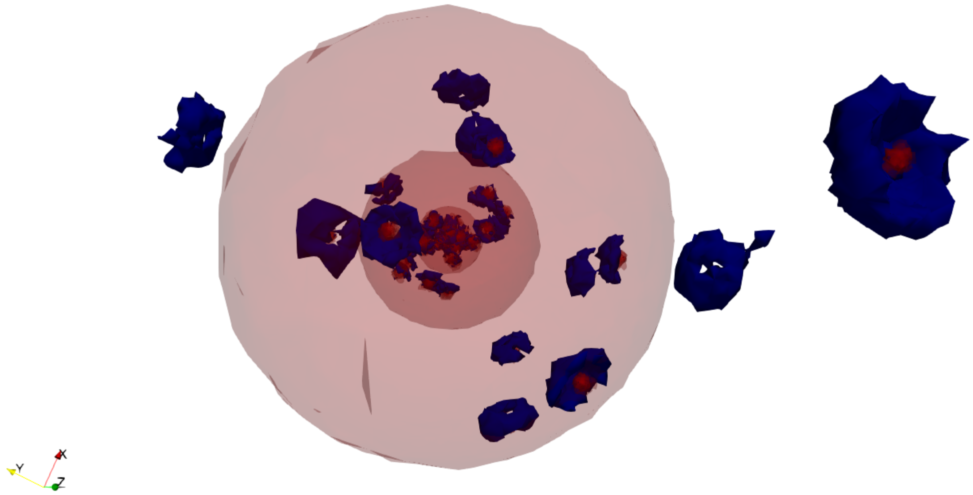

Figure 5.3.10: 3D view of the isosurfaces for the baryonic mass distribution for the A0085 cluster. The red surfaces represent the positive baryonic mass density, while the blue isosurfaces represent the points at which the apparent mass distribution vanishes, implying a negative PDM density $M_D = -M_B$.

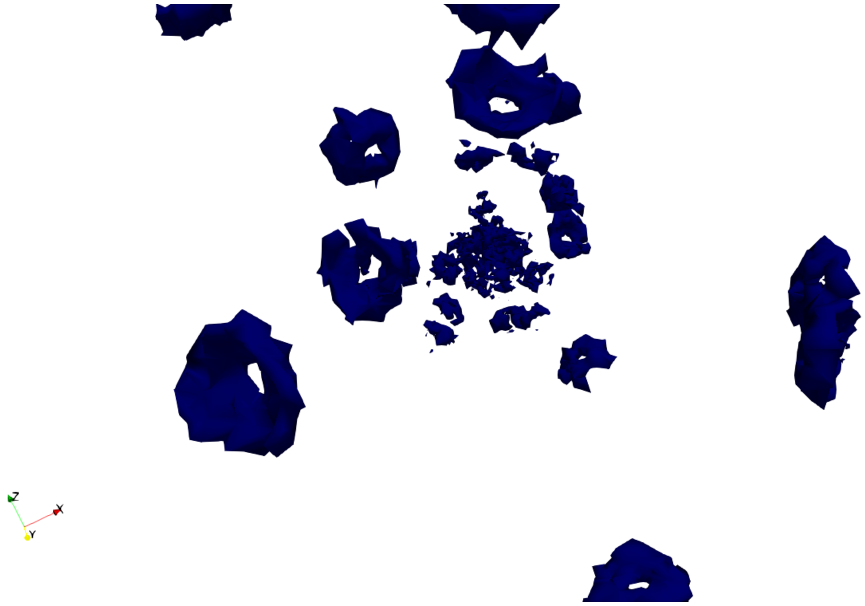

Figure 5.3.11: 3D view of the isosurfaces in A0085 for the negative PDM density $M_D = -M_B$. The angle of this 3D view does not correspond to the one of the other plots, since it is pointing towards the center of the ICM, in the direction of thee external acceleration field. Due to this, the torus shapes are more clearly visible.

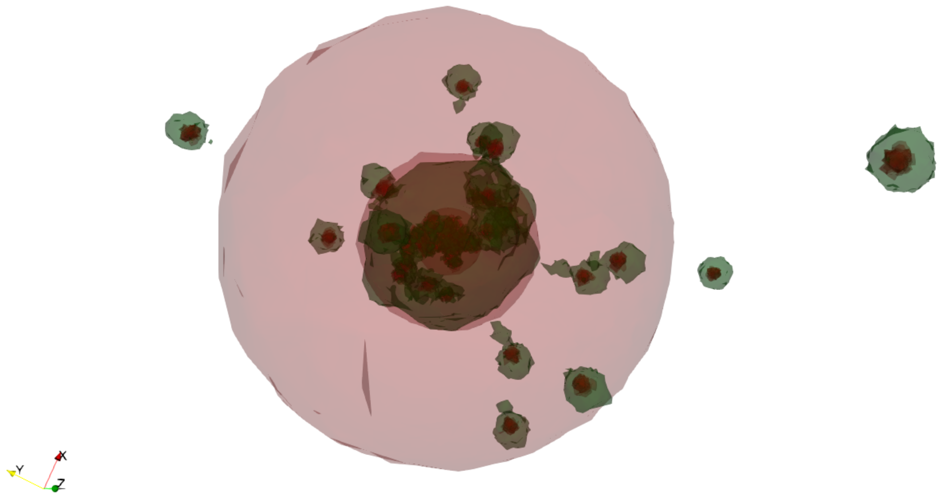

Figure 5.3.12: 3D view of the isosurfaces in A0085 for the positive contributions of the baryonic distribution (red) and the apparent distribution (green).

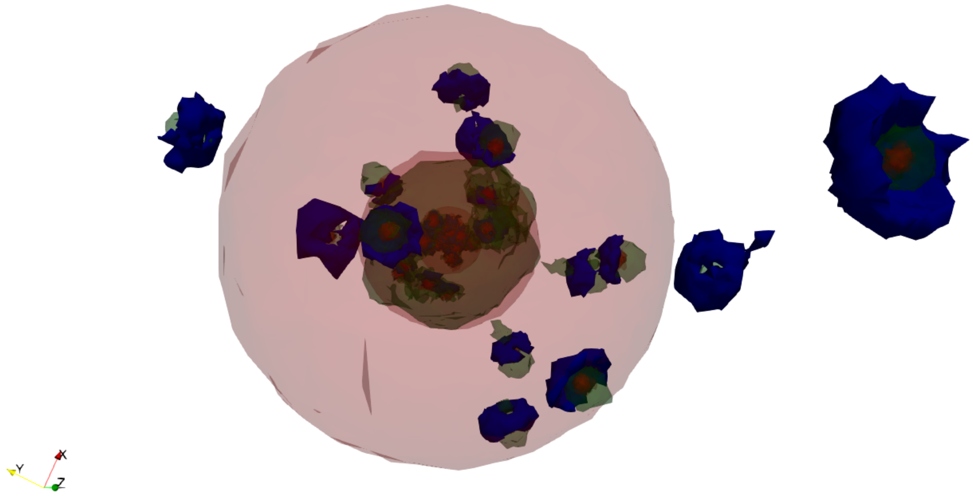

Figure 5.3.13: 3D view of the isosurfaces in A0085 including the positive contributions of the baryonic distribution (red) and the apparent distribution (green), and the negative PDM contributions (blue).

Part II

Computer Engineering

Chapter 6

Finite Element Method

This chapter gives an introduction to the **Finite Element Method** (**FEM**). FEM is the numerical technique which was used to compute the solution to the MOND **Boundary Value Problem** (**BVP**), through the use of the `FEniCS` software, which will be introduced in chapter 7. This chapter will begin by describing the basic structure of FEM, explaining why it is a popular numerical method and giving its working principles from a non-mathematical point of view. Next, the concepts of the weak form and test function will be introduced, alongside the most common FEM implementation, namely the Galerkin method. Examples of the weak form will then be given for the Poisson and MOND equations. Subsequently, the discretisation procedure will be treated. Three main concepts are discussed: the elements, basis functions and the assembly of the integrals into element matrix and element vector. For each concept, a brief mathematical overview will be given, after which implementations will be described, utilising results from the literature.

6.1. The Need for a Numerical Solution

In an ideal world, the solution to equations describing physical problems would be given in terms of known analytic functions, such as polynomials, exponentials, trigonometric functions and similar. Unfortunately, analytic solutions to PDEs are not always available, for reasons ranging from non-linearity to the use of complex domain geometries.

In the case of the MOND PDE, exact general solutions[1] are not known even for the simplest 3D domains such as spheres or cylinders. Therefore, numerical methods need to be utilised to understand the behaviour of any physical system of interest which, for MOND, are mostly galaxies and galaxy clusters.

This section introduces the work that has been carried out in the past to obtain both exact and numerical solutions to the MOND equations. Through the description of previous results, we will argue for the need for a more robust numerical treatment, particularly

[1] By general solution here we mean solutions which do not make use of principles of symmetry. For example, a spherically symmetric solution is not a general solution, and is only applicable when there is a spherically symmetric mass distribution.

in the case of discrete mass distributions. Ultimately, we will explain why the Finite Element Method is the most suitable numerical scheme for the purpose.

The MOND PDE with interpolation function $\mu(x)$ has no analytic solution. One can find approximate solutions for simple mass distributions, which either possess strong symmetries (spherical or axial), or have a physical extension which is negligible when compared to the chosen domain, such as point sources. However, these solutions are of limited use in physically interesting situations, due to the assumptions which have to be imposed to find a closed form.

Analytic solutions of toy models were treated in [13] and [55], with a focus on cylindrically symmetric distributions, which are of great relevance in the study of galaxies. Non-exact solutions were also treated in [13] through perturbation theory and the use of the Finite Difference Method. Nonetheless, none of these implementations can be used to model the irregular mass distributions which is necessary for the analysis of galaxy clusters. Galaxy clusters are particularly important because the MOND predictions do not seem to match observational evidence when simplified mass models are used [56]. Even more importantly, previous work on the behaviour of MOND in galaxy clusters [49] did not directly compute a solution to the full MOND PDE. Instead, it calculated the Newtonian force and inferred the resulting MOND force by the a posteriori use of the interpolation function $\mu(x)$. This approach neglects the crucial non-linearity of the MOND PDE, and by construction cannot make a distinction between the potential stemming from a continuous distribution and a collection of discrete sources, if the total mass of the two systems is equal.

In galaxy clusters, continuous distributions correspond to the gas in the **Intra Cluster Medium** (**ICM**), whereas the galaxies can be considered as discrete sources with a radius smaller than that of the cluster by at least two orders of magnitude. Therefore, if one wants to analyse the behaviour of MOND in galaxy clusters, a numerical solution, which has good performance when the source term is non-homogeneous, is needed. Mesh based methods, such as Finite Differences, Finite Volume and Finite Element can be utilised, but in order to obtain both convergence in the solution and low computation times, particular attention must be given to the meshing process.

6.2. Introduction to FEM

When dealing with numerical methods, one has to abandon the concept of continuous functions: both inputs and outputs need to be discretised and expressed with finite precision. This implies the following:

1. The **domain** of a problem can only be described by a finite number of points. For example, these can be identified as the **nodes** of a **mesh**, and any function on the domain will be evaluated at this set of points;

2. The value of any function can only be calculated at a finite number of nodes in the domain. In general, the value of the function will contain a finite error when compared to an exact analytical expression.

The concept of the mesh is fundamental in most numerical schemes, including FEM. The term mesh refers to a collection of objects that can be characterised based on their geometric dimension.[2] As an example, a mesh in n dimensions is composed of objects that have geometric dimensions 0 through n. For MOND one is interested in the 3D case where:

- Points are also referred to as **vertices** or nodes. They have dimension 0. It is on these nodes that one obtains the value of a solution;

- Lines are termed **edges** in 3D, and have dimension 1;

- Surfaces come in the form of **facets**, defining the boundary of each 3D element, and have dimension 2;

- Polyhedra, such as tetrahedra, are the **elements** in 3D and have dimension 3. They are also referred to as **cells**, and the solution is computed over each of them individually.

As mentioned above in the definition of a cell, in FEM the solution to a BVP is calculated separately over each cell, with the value of the solution being approximated on each node that makes up the cell. From a mathematical standpoint, the aspect which makes FEM a powerful numerical method is that the domain over which the BVP is defined is subdivided into smaller subdomains. On these subdomains, the solution can be approximated by combinations of simple functions, such as polynomials. With FEM, solutions to BVPs on complex domain geometries can be obtained by splitting the overall domain into smaller, simpler domains of the chosen shape, such as cubes or tetrahedra in 3D. Overall, obtaining the solution of a BVP using FEM involves the following steps:

1. Choosing a domain of computation and defining the BVP on it, including the PDE and the boundary conditions;

2. Obtaining a **weak form** from the PDE;

3. Deriving a system of equations from the weak form that can be solved on each element;

4. Solving the system of equations on each element to obtain the value of the solution at every node in the domain.

[2] It will be explained in chapter 8 that a topological dimension is also required in the `FEniCS` software. However, for many purposes the geometric dimension is sufficient.

6.3. Weak Form and Test Functions

Certain aspects of the above-mentioned steps need to be described more in detail. The most important features of FEM include the introduction of so-called test functions and the use of the weak form.

Without delving too deeply into the mathematical formalism, the reason why introducing a test function and weak form is advantageous is the following: when defining a PDE over a domain, one must ensure that the PDE is valid at each point of the domain itself. For example, if the PDE contains second derivatives, the solution must have second derivatives at each point of the domain. This is quite a stringent condition.

Instead, a test function, which is a function which shares a given property with the solution, could be introduced. The case of interest is that of a test function which belongs to the same function space as the solution. If one multiplies both sides of the PDE by this test function, one obtains an equivalent expression. The expression can now be integrated. This is where the weak form is advantageous: as one is dealing with a PDE, there will be spatial derivatives of the solution. Since both sides of the PDE were multiplied by a test function, one can use integration by parts to redistribute the differentiation between the solution and the test function, hence decreasing the order of the partial derivative on the solution. For example, if the highest order derivative on the solution is reduced from 2 to 1, the solution is no longer required to have second derivatives throughout the whole domain, but only first derivatives. This is a weaker requirement, hence the name **weak form**.

Moreover, the following important aspect must be noted:

The PDE must hold at all points of the domain. Instead, the integral expression giving the weak form must only be valid for all test functions in the chosen function space.

In synthesis, in the initial PDE the solution is required to satisfy a stringent condition on its differentiability, at all points in the domain. After multiplication by a test function and integration, the resulting expression has less stringent constraints and has to hold for all possible test functions rather than all points in the domain. If the solution of the PDE is sufficiently well behaved, it can be shown that it coincides with the solution of the weak form. Therefore, the weak form can be seen as offering a different formulation for the problem, for which the solution is identical, can be obtained more easily, and can be more effectively computed through a numerical scheme.

There are various ways to arrive at the weak form. The two main approaches are given by the Ritz method and the Galerkin method. Given that for the MOND PDE only the Galerkin method can be used, the description of the Ritz method is given in appendix D. The Galerkin method also includes a prescription to obtain a discretised system from the weak form. for. The latter will be described once the method has been explained.

6.4. Galerkin Method

The strong conditions imposed on the use of the Ritz method (see appendix D) are absent from the Galerkin method. The latter method can, however, be used to solve non-linear PDEs with inhomogeneous boundary conditions such as MOND. Moreover, the procedure requires fewer steps. These are:

1. Obtaining a weak form from the BVP.
2. Discretising the weak form through the use of basis functions and the introduction of elements;
3. Deriving the **element matrix** and **element vector** from the discretised weak form.

In the next subsection, we will find the weak form for two different PDEs. First, for the Poisson equation, one of the simplest PDE formulations, in order to apply the basic concepts without further complications. Subsequently, for MOND with one of the possible interpolation functions, as this is the problem which needs to be solved.

6.5. The Weak Form for the Poisson Equation

To allow for a smooth transition to the description of the MOND PDE, it is useful to begin by treating the Poisson equation for gravity. The solution is the gravitational potential ϕ, the mass density is ρ and G is the gravitational constant. The Poisson equation can then be given as:

$$\nabla \cdot (\nabla\phi) = 4\pi G\rho \in \Omega. \tag{6.5.1}$$

The domain Ω is taken to be the sphere in 3D. In the case of the Poisson equation inside a 3D sphere, the solution is known. This can then be used as the boundary condition, where M is the total mass enclosed in the sphere:

$$\phi = \frac{GM}{r} \in \Gamma. \tag{6.5.2}$$

Γ is the boundary of the domain Ω. For a sphere in 3D, this is the surface of the sphere, and it assumes the role of spatial infinity. When the value of the solution is explicitly prescribed on the boundary, the test function is required to vanish on the boundary:

$$\psi = 0 \in \Gamma. \tag{6.5.3}$$

As mentioned in sections 6.3 and 6.4, one way to obtain the weak form is the following: one multiplies each side of the PDE by a test function (which might be required to satisfy some additional constraints, e.g. on the boundary of the domain), integrates both sides of the expression, and reduces the order of partial derivatives. Applying these steps to eq. 6.5.1 with a test function ψ we obtain:

$$\int_\Omega \nabla \cdot (\nabla\phi)\, \psi \ \mathrm{d}\Omega = \int_\Omega 4\pi G\rho\psi \ \mathrm{d}\Omega. \tag{6.5.4}$$

To reduce the order of differentiation on the solution ϕ, one can use integration by parts to arrive at:

$$\int_\Omega \nabla \cdot (\nabla\phi \cdot \psi)\ \ \mathrm{d}\Omega - \int_\Omega \nabla\phi\nabla\psi \ \mathrm{d}\Omega = \int_\Omega 4\pi G\rho\psi \ \mathrm{d}\Omega. \tag{6.5.5}$$

As the integrand of the first integral on the LHS is a divergence, one can use the divergence theorem to obtain:

$$\int_\Gamma \nabla\phi \cdot \psi \ \mathrm{d}\Gamma - \int_\Omega \nabla\phi\nabla\psi \ \mathrm{d}\Omega = \int_\Omega 4\pi G\rho\psi \ \mathrm{d}\Omega. \tag{6.5.6}$$

The resulting integral over the surface Γ contains ψ. From eq. 6.5.3 we have $\psi = 0$ on the boundary, so this term vanishes. The remaining expression can be used to give the weak form as follows. One must first introduce the function space to which both ϕ and ψ belong to and call it Σ. Then, the weak form is given by the following statement.

Find ϕ such that the following expression holds for all test functions ψ in Σ that satisfy $\psi = 0 \in \Gamma$:

$$\boxed{-\int_\Omega \nabla\phi\nabla\psi \ \mathrm{d}\Omega = \int_\Omega 4\pi G\rho\psi \ \mathrm{d}\Omega.} \tag{6.5.7}$$

6.6. The Weak Form for MOND

In order to find the weak form for MOND, one follows similar steps to the ones for the Poisson equation. However, as there are different interpolation functions in MOND, which give rise to different PDEs, each of these has to be treated separately. In the following, the weak form is obtained for the MOND PDE with the simple interpolation function. One has the PDE:

$$\nabla \cdot \left(\frac{|\nabla\phi|}{|\nabla\phi + a_0|} \nabla\phi \right) = 4\pi G\rho \in \Omega. \tag{6.6.8}$$

Ω is the domain of the solution. The case of interest for Ω is the 3D sphere. In order to find a solution, one must give a boundary condition, such as the behaviour of the potential ϕ at spatial infinity. Following [12] and [13], one can define the Dirichlet Boundary Condition (BC):

$$\phi = \sqrt{MGa_0} \ln\left(\frac{r}{r_0}\right) \in \Gamma. \tag{6.6.9}$$

This type of BC, which has to be imposed on the PDE and also on the weak form, is called essential.

To obtain the weak form, one has to again multiply the PDE by a test function, integrate over the domain, and reduce the order of the partial derivatives. Starting from eq. 6.6.8 and multiplying by the test function ψ we have:

$$-\int_\Omega \psi\nabla\cdot\left(\frac{|\nabla\phi|}{|\nabla\phi|+a_0}\nabla\phi\right)\,\mathrm{d}\Omega = \int_\Omega 4\pi G\psi\rho\,\mathrm{d}\Omega. \tag{6.6.10}$$

Using integration by parts on eq. 6.6.10 we arrive at:

$$\int_\Omega \nabla\cdot\left(\frac{|\nabla\phi|}{|\nabla\phi|+a_0}\psi\nabla\phi\right)\,\mathrm{d}\Omega - \int_\Omega \frac{|\nabla\phi|}{|\nabla\phi|+a_0}\nabla\phi\nabla\psi\,\mathrm{d}\Omega = \int_\Omega 4\pi G\psi\rho\,\mathrm{d}\Omega. \tag{6.6.11}$$

As for the Poisson equation, the test function ψ must vanish on the boundary:

$$\psi = 0 \in \Gamma. \tag{6.6.12}$$

As eq. 6.6.11 contains a divergence term, one can use the divergence theorem to obtain:

$$\int_\Gamma \frac{|\nabla\phi|}{|\nabla\phi|+a_0}\psi\nabla\phi\,\mathrm{d}\Gamma - \int_\Omega \frac{|\nabla\phi|}{|\nabla\phi|+a_0}\nabla\phi\nabla\psi\,\mathrm{d}\Omega = \int_\Omega 4\pi G\psi\rho\,\mathrm{d}\Omega. \tag{6.6.13}$$

Since the first term is now an integral over the boundary Γ, over which $\psi = 0$, the first term cancels, and one is left with the weak form. As stated above, the weak form must hold for all test functions in the chosen function space, for which details are now unimportant, named Σ. Mathematically, the weak form is given by the following statement.

Find ϕ such that the following holds for all test functions ψ in the function space Σ that satisfy $\psi = 0 \in \Gamma$:

$$\boxed{-\int_\Omega \frac{|\nabla\phi|}{|\nabla\phi|+a_0}\nabla\phi\nabla\psi\,\mathrm{d}\Omega = \int_\Omega 4\pi G\psi\rho\,\mathrm{d}\Omega} \tag{6.6.14}$$

As can be seen in eq. 6.6.14, there are only first derivatives of the solution and test function. Hence the requirement is that ϕ is differentiable once, rather than twice differentiable, as was the case with the original PDE formulation. This is a weaker requirement than what was initially set.

6.7. Discretisation and Basis Functions

Once the weak form has been obtained, the problem can be discretised. However, before giving a specific example by carrying out the discretisation of the weak form which was found above for the Poisson equation, it is necessary to introduce some concepts which are generally applicable. Despite the intention to avoid being too technical, it must be pointed out that the functions which have been defined so far belong to an infinite space, namely a Hilbert space. For the purposes of this discussion, it is sufficient to point out that a function belonging to an infinite space can be defined as an infinite sum of basis functions, each coming with a specific coefficient. Using the notation u for a function in an infinite function space, $\tilde{v}_i$ for the basis functions and $\tilde{c}_i$ for the respective coefficients, one can express u through the infinite sum:[3]

$$u = \sum_{i=0}^{\infty} \tilde{c}_i \tilde{v}_i. \tag{6.7.15}$$

As computers cannot calculate infinite quantities, one must define the function as a finite sum, which is an approximation to the exact function u. The function defined through this finite sum thus belongs to a finite subspace of the infinite space one started with. Calling the approximation from the finite subspace u_a, one can write:

$$u_a = \sum_{i=1}^{n} c_i v_i. \tag{6.7.16}$$

As the expression in eq. 6.7.16 is a finite sum, the existence of a finite amount of basis functions v_i is implied. The next aspect to note is the following: the weak form has to hold for every test function ψ. As the test function is required to belong to the same function space as u, its approximation must also belong to the same finite subspace. Therefore, the test function can be expressed as a finite sum over the basis functions, and requiring that the weak form holds for all test functions is equivalent to requiring that it holds for all of the basis functions independently.

Having defined these concepts, it is now possible to express the weak form for the Poisson equation in terms of the approximated solution. Using again the notation v_i for the basis functions and c_i for the respective coefficients, this involves making the substitution in eq. 6.6.14:

$$\phi \approx \sum_{i=1}^{n} c_j v_j. \tag{6.7.17}$$

Furthermore, the weak form must now hold for all basis functions v_j. The basis functions have not yet been defined. In general, one can choose any set of functions which satisfies certain mathematical requirements, for example orthogonality. This means that the product of any two basis functions is 0, unless it is the product of a basis function with itself. However, it is assumed that the basis functions are known, hence the unknowns are now the coefficients c_i. The problem now is to find these coefficients rather than the initial

[3] In the literature, it is common to assign the letter ϕ to the basis functions. However, in the MOND PDE ϕ is the variable of interest, so the notation v_i is used instead for the basis functions.

exact function ϕ. Generally, the discretised version of the weak form follows from making the two simple replacements:

$$\phi \to \sum_{j=1}^{n} c_j v_j, \quad \psi \to v_i \tag{6.7.18}$$

With these substitutions, eq. 6.6.14, which gives the weak form for the Poisson equation, now reads as follows:

Find the coefficients c_j such that the following expression holds for all basis functions v_i:

$$\boxed{-\sum_{j=1}^{n} c_j \int_\Omega \nabla v_j \nabla v_i \, \mathrm{d}\Omega = \int_\Omega 4\pi G \rho v_i \, \mathrm{d}\Omega.} \tag{6.7.19}$$

It can then be seen that eq. 6.7.19 describes a matrix-vector multiplication by making the substitutions:

$$-\int_\Omega \nabla v_j \nabla v_i \, \mathrm{d}\Omega \to A_{ij}, \quad \int_\Omega 4\pi G \rho v_i \, \mathrm{d}\Omega \to f_i. \tag{6.7.20}$$

The newly introduced quantities are the element matrix A_{ij} and the element vector f_i. The discretised weak form then becomes:

$$\boxed{A_{ij} \cdot c_j = f_i.} \tag{6.7.21}$$

It was previously explained that the coefficients c_j uniquely define the solution ϕ. The expressions for A_{ij} and f_i are known once the respective integrals have been solved. After computing the integrals, the approximate solution can be found by solving the system of equations in eq. 6.7.21. In order to arrive at the solution, three steps are in general non-trivial:

1. Dividing the domain into sub-domains, called elements;

2. Choosing a set of suitable basis functions for the problem;

3. Numerically calculating the integrals defining the element matrix and element vector.

These steps will be discussed individually in the subsections below. For each step, first a brief mathematical explanation will be provided. Techniques for their computation will then be given, together with implementations from the literature.

6.8. Elements

6.8.1. Mathematical Problem

Both of the integrals given in eq. 6.7.19 are given over the domain Ω. So far, the basis functions have been treated as continuous functions. However, in order to calculate the integrals numerically, one must define a finite number of points over which the approximate solution will be calculated. These points are called nodes, or vertices. Once a finite number of vertices have been introduced, one can no longer exactly cover the initial domain Ω, but only an approximation of this domain, denoted by $\tilde{\Omega}$. The result of integration over the domain $\tilde{\Omega}$ can be decomposed into the sum of the integration over subdomains e_k which add up to $\tilde{\Omega}$. For example, the integral could be decomposed to calculate A_{ij} over m subdomains:

$$-\int_{\Omega} \nabla v_j \nabla v_i \, \mathrm{d}\Omega = -\sum_{k=1}^{m} \int_{e_k} \nabla v_j \nabla v_i \, \mathrm{d}e_k. \tag{6.8.22}$$

In 3D, the elements define the volume of a cell, and the union of all the cells is equal to the approximated domain $\tilde{\Omega}$. The process of subdividing the domain into subdomains forms a mesh, which is defined by the cells composing it and the nodes over which the solution is calculated. The following important aspect should be noted: in theory, the integral over each element can be calculated independently. However, one would have to calculate the integral of the weak form as a function of the test functions for all values of i and j. For the integrals to be truly independent from each other, one must make a smart choice for the basis functions.

6.8.2. Implementation

The subdivision of the domain into a mesh made of nodes and elements is a step of crucial importance for FEM. As described in [57], element quality can be defined by different criteria, but all definitions agree on one main factor:

> Regardless of the exact geometric shape of the element, elements of higher quality are those which are closest to a regular shape.

For 3D, the element geometry of interest is generally the tetrahedron. A good element is thus given by a tetrahedron whose sides are approximately equal. A more in depth discussion of element quality will be given in chapter 8. It is then important that the nodes are connected in such as way as to obtain elements which all have approximately a regular shape. The most common technique to obtain simplex elements (triangles in 2D, tetrahedra in 3D) is **DT** (**Delaunay triangulation**). Through this technique, each element is defined so that its circumsphere (the smallest sphere containing the element) does not contain the center point of any other element.

In addition to generally producing elements of high quality for complex domain shapes, it was shown in [58] that for the case of 2D meshes the Delaunay triangulation can be improved to achieve a scaling of $O(n\log(n))$ on a sequential machine by utilising a quad-edge data structure, which allows to concurrently store the information of four edges. Moreover, it was recently shown in [59] that on a multi-core system with 16 single-threaded cores, optimisations based on point insertion in 2D meshes allowed the computation of the triangulation with a speedup of ≈ 14 compared to a sequential program with a scaling of $O(n\log(n))$. On the other hand, the implementation from [59] did not provide comparable speedups for the case of 3D meshes.

However, it was shown in [60] that through GPU acceleration, the triangulation for 3D models with $O(10^6)$ nodes could achieve a speedup of ≈ 6 w.r.t. a sequential counterpart running on a CPU. Moreover, for models with $O(10^7)$ nodes, the speedup achieved reached a factor of ≈ 10. As the technique used in [60] was also that of point insertion, these results show that for the case of 3D meshes, GPUs can provide speedup of order $O(10)$ for Delaunay triangulation w.r.t. CPU implementations. It is hence clear that the use of Delaunay triangulation is optimal for the mesh generation required in FEM, because of the possible speedups achievable by multi-core and GPU-accelerated systems in both 2D and 3D. As a result, most mesh generation libraries, such as `tetgen` and `CGAL`, offer Delaunay as the default triangulation backend, as explained in [61].

6.9. Basis Functions

6.9.1. Mathematical Problem

To ensure that the integral over each element e_k can be evaluated independently, each basis function must be defined so that it is only non-zero at a given point in space. As the domain is now given by a discrete set of points, this can be done by using so-called hat functions[4]. Formally, a hat function is defined to be only non-zero on a specific node, and zero everywhere else. Using the indices i for the basis functions and j for the discrete points in the domain, this can be defined as:

$$\boxed{v_i(x_j, y_j, z_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}} \tag{6.9.23}$$

If the elements are small compared to the domain $\tilde{\Omega}$, one can make the approximation that, over each element, every basis function can be approximated as a linear combination of the coordinates. Indicating the value of the basis function v_i on the element e_k as v_i^k,

[4] This description is limited to elements of degree 1, but the same reasoning can be used to make elements of higher order by adding more nodes to each element.

this implies[5]:

$$\boxed{v_i^k = a_i + b_i x + c_i y + d_i z.} \tag{6.9.24}$$

The coefficients a_i, b_i, c_i, d_i are chosen so that on every element e_k, the value of each test function v_i^k at each node of the element follows the form given in eq. 6.9.23. Eq. 6.9.23 implies a crucial feature: as each test function is only non-zero at a given node, test functions which are not directly connected by an edge do not overlap. Given that nodes which are connected by an edge are part of the same element, for all nodes i, j which do not belong to the same element one has:

$$\int_{e_k} v_i v_j \, \mathrm{d}e_k = \int_{e_k} \nabla v_i \nabla v_j \, \mathrm{d}e_k = 0. \tag{6.9.25}$$

When considering the matrix structure of the discretised weak form of eq. 6.7.19, one can then see that for a large number of elements $k \gg 1$, the resulting matrix will be sparse, because the vast majority of the pairs of basis functions with indexes i, j will give zero contributions.

6.9.2. Implementation

The mathematical description provided in the previous subsection only concerned the use of linear elements for reasons of simplicity. However, in real world applications, linear elements often cannot be used. For example, if one is interested in taking the second derivative of the solution, as in the case of galaxy clusters, one must use elements on which the solution is approximated by a higher order polynomial. Moreover, polynomials are not the only choices that can be made for the basis functions on each element. Choosing the best element for the problem at hand can quickly become complex, and for higher order elements the solution of the integrals over each element becomes challenging.

In addition, it is well known that specific problems are best treated with non-standard basis functions, such as the Nedelec elements introduced in [62] for EM problems. The greatest issue is that many libraries for FEM do not offer compatibility with most of the families of basis functions which are needed to achieve the highest precision in the solution of specific problems such as EM simulations. This is because it is difficult to obtain closed forms for even a lower order approximation to these basis functions, as explained in [63]. One approach that allows for the use of most families of basis functions is given by FIAT (FInite element Automatic Tabulator), introduced in [63] and [64].

The approach taken by FIAT is to approximate arbitrary basis functions by linear combinations of polynomials of arbitrarily high order. This can be done for elements of varying geometries, both in 2D and 3D, including triangles and tetrahedra. Once the basis functions have been approximated, their values are stored in a file for future use.

[5] It should again be noted that this is the approximation made for linear elements. If using polynomials, one defines the basis functions on each element as a polynomial function of order n of the coordinates. Degree 3 elements are given by an order 3 polynomial.

Although this implies an overhead in the computation, it allows for a better evaluation of the solution, as well as bridging the gap between the mathematical formulation and the programming implementation. FIAT was first implemented in Python, because of advantages such as automatic memory management and ease of interface with a large number of linear algebra libraries, necessary to compute the basis functions efficiently.

It was also shown in [65] that by integrating the level 3 BLAS library in FIAT, speedups from $O(10)$ up to $O(10^3)$ could be achieved in the tabulation of complex basis functions, thanks to the optimisation of the dense-matrix operations required. FIAT is one of the building blocks of many modern FEM solvers, such as FEniCS, which will be introduced in chapter 7.

6.10. Integration and Assembly

As explained above, the only non-zero contributions to the element matrix come from nodes belonging to the same element. The last step which is necessary to obtain the matrix element is hence the computation of the value of the integral on each element. As one chooses elements of a simple geometry, it is sometimes possible to solve the integrals over the basis functions analytically. When this is not possible, numerical integration must be used. Ultimately, one has to obtain the element matrix for the total system by assembling the matrices obtained for every element e_k, by utilising a **mapping** w.r.t. to the vertices of each element. The same must be done for the element vector. However, in this case, the coefficients for each basis function v_i are known, as they depend on the interpolated value of the source at each discrete point x_j.

6.10.1. Implementation

To obtain the total element matrix and element vector, four steps are necessary:

1. Solving the integral involving the basis functions for both the solution and the test function, v_i and v_j. This gives the element matrix A_{ij}^k for each element e_k;

2. Solving the integral involving the interpolation of the source and the basis function for the test function v_i, The coefficients of v_i are given by the interpolated values of the source f at all nodes x_j, and are known a priori. This gives the element vector f_i^k for each element e_k;

3. Once the integrals for the element matrix have been computed over each element e_k, the results need to be assembled into the total element matrix according to the indices i, j;

4. Once the integrals for the element vector have been computed for each element e_k, the total element vector needs to be assembled according to the index i.

The solution of the integrals depends on the choice of the basis functions, the number of dimensions of the mesh, the geometry of the elements, and the value of the source at each node. Therefore, it is hard to give a standard measure of performance for the general case without specifying the element type and basis functions. However, in all cases, one must have a connectivity list indicating which pairs of nodes i, j are connected. This list is called the **mapping**.

It was shown in [66] that by utilising the MPI standard on a multi-core CPU system, speedups of $O(10)$ can be achieved for the mapping on meshes with $O(10^4)$ nodes. To arrive at this result, instead of having a global mapping, each element is made to hold a list of its own mapping into the total element matrix. Without the need for a global mapping, the mapping computation can be more efficiently distributed for parallel computation. Moreover, it was shown in [67] that a similar method could be paired to a more efficient compression of sparse matrices to obtain speedups of $O(10)$ in the mapping for meshes with $O(10^5)$ nodes. The speedup is defined w.r.t. standard libraries such as uBLAS, in the case of sequential computation. The results from [67] are particularly relevant for this work as they are integrated with **FEniCS Form Compiler** (**FFC**) described in chapter 7.

6.10.2. Solution of the Linear System

6.10.3. Implementation

After the integrals have been computed for the total element matrix and total element vector, one is left with a linear system of the form:

$$A_{ij}c_j = f_i. \tag{6.10.26}$$

The solution is uniquely determined by the coefficients c_j, and there are generally two different methods for obtaining these coefficients.

1. **Direct solvers**: through techniques such as **Lower-Upper** (**LU**) decomposition. In this case, only matrix-vector multiplication is required, once the element matrix A_{ij} has been decomposed into the product of a lower triangular and an upper triangular matrix, L and U respectively:[6]

   $$A_{ij} = LU. \tag{6.10.27}$$

   However, the method grows with $O(n^3)$ in the number of nodes in the mesh. Moreover, if the matrix M is mostly sparse, the computation can be inefficient as empty cells are multiplied together.

2. **Iterative solvers**: rather than solving the problem directly by matrix multiplication, an iterative solver starts with an approximation of the solution. Iterative solvers do not provide an exact solution as in the case of direct solvers, but instead find a best approximation. This is particularly useful when solving a non-linear problem.

[6] Indices for the L and U matrices are suppressed to avoid clutter.

The MOND PDE is non-linear, but in certain regimes it converges to the Poisson equation, which is linear. Moreover, as indicated, for example, in [68], iterative solvers become more effective for meshes with $O(10^6)$ elements. Therefore, in order to determine which solver is most suitable for a problem, one must not only consider its size, but also the distribution of the sources and the non-linearity in the considered regime. A more in-depth study of the speedup which can be provided by various direct and iterative solvers for the MOND PDE will be given in chapter 8.

6.10.4. The FEM Workflow and FEniCS

For the implementatiofn of a complete FEM software, one must provide code to compute each quantity present in the aforementioned section. The fundamental issue is that, from a mathematical point of view, FEM is applicable to most of the problems in engineering and science which can be expressed in PDE form, such as **Computational Fluid Dynamics** (**CFD**), **Electromagnetism** (**EM**), statics, gravity and so on. However, as was explained when describing basis functions, different PDEs generally require different domains, problem sizes, basis functions and element types. Therefore, even though each component of the FEM workflow can be executed with high performance, the lack of a package capable of handling the compatibility between each of the fundamental building blocks greatly hinders the ability to truly exploit the power of the FEM formalism.

For this reason, efforts have recently been made to unite the various high performance libraries for use in FEM into a single framework. The prime example of this category of open source general FEM solvers is given by FEniCS, first introduced in [69]. FEniCS handles all the stages necessary for computing the FEM solution for an arbitrary PDE, including mesh generation, element definition, integral calculation and matrix assembly, and solution to both linear and nonlinear systems of equations. The packages allowing the computation of these steps include the previously mentioned CGAL library for mesh generation, FIAT for element definition, and FFC to obtain element matrices and element vectors. The next chapter is dedicated to an in-depth description of FEniCS, and the following two chapters give the results of its use to find the solution of the MOND PDE, first with a sequential program, and subsequently through parallelisation using the MPI standard.

Chapter 7

Introduction to FEniCS

This chapter serves as an introduction to FEniCS, an open source project aimed at providing a scalable, high performance platform for mathematical modelling through FEM. FEniCS is supported for both C++ and Python. Throughout this work, almost exclusive use was made of Python, although C++ code is present in the definition of a limited amount of objects, such as `Expressions`, which will be described throughout the chapter. The chapter is structured as follows: first, an overview of FEniCS will be given. Next, a general description will be provided for all the main components which form FEniCS. Finally, the main classes and functions from the package will be described, and their relation to the code developed to solve the MOND PDE will be explained.

7.1. Building Blocks of FEniCS

FEniCS can generally be utilised without the need to access the core libraries which separately implement its basic functionalities, such as mesh generation and linear algebra calculations. Nonetheless, it is important to describe its main building blocks because in order to increase the performance of the solution of a PDE, it is often necessary to modify the parameters which define the behaviour of the underlying libraries. The first thing to mention is that there are two separate interfaces to FEniCS:

- A C++ library;
- A Python module.

The C++ library defines all the classes which are necessary for the Finite Element implementation, as well as a number of functions which are used to obtain the solution of a given PDE, such as `assemble` and `solve` (see, for example, [68]). On the other hand, the Python module for FEniCS is mostly generated through the use of **Simplified Wrapper and Interface Generator** (**SWIG**). However, certain features, such as the full integration of **Unified Form Language** (**UFL**), are exclusive to the Python implementation. UFL has an important role in FEniCS, as it allows all relevant quantities for a FEM implementation to be defined with a syntax close to the mathematical notation.

This results in a more automated solving process, as most aspects concerning the element type or explicit assembly of element matrices and element vectors are handled by FCC. Both FCC and UFL will be described more in detail throughout the rest of the chapter. In order to access all functionalities of FEniCS, it is sufficient to import a single module, which can be done as follows for Python:

```python
from dolfin import *
```

In the Pyhton implementation, all separate sub-modules are imported through the above command. It is useful to give an overview of the most important of these, while a schematic depiction of all sub-modules and their relations is given in fig. 7.1.1:

7.1.1. DOLFIN

DOLFIN is the main user interface of FEniCS for both the C++ and Python versions. DOLFIN allows the access to all of the separate sub-modules and is hence a critical component of any FEniCS workflow. It is important to mention that the efficiency of a FEniCS program is the same whether the chosen language is Python or C++. This is due to the fact that, in both cases, functions and classes used are implemented from the C++ library. As previously noted, this is mostly achieved through the use of SWIG.

7.1.2. UFL

As stated above, UFL allows a definition of quantities needed for a FEM solution through a syntax which is remarkably close to the mathematical notation. This is particularly true for weak forms, which can be given directly as integrands over the domain. When one has a weak form and a domain with related mesh, the element matrix and vector need to be assembled through either SyFi or FFC, which then produces UFC code.

7.1.3. SyFi and FFC

At the core of any FEM solver are systems of linear equations, implemented as a matrix-vector multiplication as described in chapter 6. These element matrix and element vector are called the **bi-linear** and **linear** forms respectively. For the system of equations to be solved, it is necessary to first assemble the system, and secondly solve the resulting equations, whether these are linear or non-linear. In most cases, sparse matrices are obtained.

As the overall element matrix needs to include the individual element matrices for each element in the domain, another quantity needs to be computed, namely, the mapping. In short, the mapping indicates the position in the overall matrix element of each individual element's matrix. In many FEM solvers, this process has to be implemented by the user, which can result in unsatisfactory performance and is generally prone to errors. FFC (FEniCS Form Compiler) can automatically produce both the element matrix and the

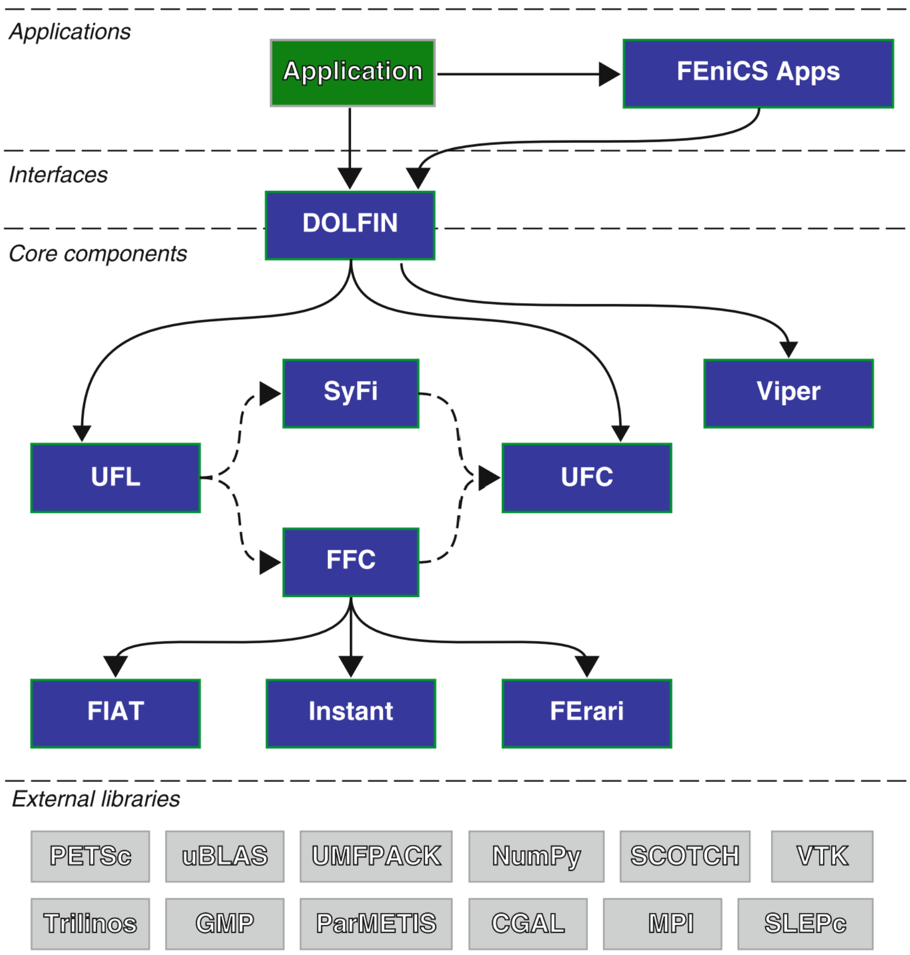

Figure 7.1.1: Schematic representation of the sub-modules present in FEniCS, their connection and role. Also depicted are a number of external libraries that are commonly utilised in connection to a FEniCS implementation. DOLFIN represents the main interface. UFL defines the syntax that must be used to define quantities such as the weak form. FFC provides functions such as the assembly of element matrix and element vector. FIAT is used for element definition, and Instant is the JIT (Just In Time) compiler. Image from [68].

mapping, with a mesh and a weak form as its input. The weak form needs to be defined through UFL. The FFC operation does not have to be explicitly called by the user, as it is handled directly by the DOLFIN sub-module [70]. The output is in the form of UFC code, described next.

7.1.4. UFC

UFC (**Unified Form-assembly code**) is an interface between code that is generally the same for most FEM implementations, and code that is usually problem-specific. UFC code expresses important quantities given a certain weak form, such as:

- **D.O.F.** (**Degree Of Freedom**) map: this indicates the value of the solution at each vertex in the domain, mapping nodes of the mesh to individual element matrices in the full element matrix;

- **Finite Element**: this describes the element used, which for the rest of this work is assumed to be a tetrahedron in 3D, on which the numerical solution is approximated by a polynomial of varying degree.

The above are implemented as C++ classes, and can also be utilised in Python through related classes. In Python, the UFC module is contained in the DOLFIN module.

7.1.5. Viper

Viper is used to implement the plotting capabilities of FEniCS. However, for the work presented here, the `matplotlib` module was utilised instead, because of its flexibility and compatibility with the `numpy` module for scientific computing.

7.1.6. FIAT and FErari

As explained in chapter 6, FIAT is the FEM backend responsible for the definition of the basis functions, which give the expression for the numerical solution of the PDE on each of the elements. FIAT also allows for the definition of elements which are of a higher degree, or non-polynomial. Higher order elements are, for example, needed when it is necessary to take the derivative of the obtained solution, such as in the case of the calculation of an apparent mass distribution for galaxy clusters. FIAT is generally not accessed by the user, and it handles the classes defining the element type, such as the `Lagrange` element class which is used throughout this work. In addition, the optional backend **FErari** gives the option of utilising graph-based optimisation, which occurs at compile-time.

7.1.7. Instant

Instant is another backend for DOLFIN, needed for JIT (Just In Time) compilation and inlining of C++ code. It allows for the use of C++ code in a FEniCS Python program, which is necessary for certain objects such as `Expressions`, which can be used to define initial guesses or boundary conditions for a given PDE. Instant again makes use of SWIG to allow the C++ code to be utilised in a Python program. Instant is particularly necessary because the form compilers such as FFC which generate UFC code, output C++ code. Through instant, form compilers such as FFC can easily be used through Python.

7.1.8. Mshr

Although not shown in fig. 7.1.1, the **mshr** module was used throughout this work. It provides a number of classes which are useful in the creation, analysis and modification of meshes, such as pre-defined meshes, information on the quality of mesh elements, and compatibility with standard mesh file types such as xml.

7.1.9. External Libraries

To optimise performance, DOLFIN utilises various classes and functionalities that belong to external modules. The most relevant for the work described here are the following:

- **NumPy**: An open source Python module providing tools for scientific computing and numerical methods in general. The most used structure of the module is the `ndarray`, which allows for the definition of arrays of arbitrary dimension. Several quantities related to DOLFIN objects come in the form of `ndarrays`, for example the value of the solution or its derivatives at each node in the domain. Moreover, Numpy allows for better integration of plotting modules. In particular, the combination with **Matplotlib**, used throughout this work, provides data visualisation similar to that of **Matlab**, in an open source environment.

- **MPI**: A standard for parallel distributed memory computing, standing for Message Passing Interface. MPI is available as a C++ library as well as a Python module, `mpi4py`. It is not necessary to explicitly import the `mpi4py` module for the Python version, as this is already included in the DOLFIN module.

- SCOTCH: A library that provides mesh partitioning capabilities. This is most useful when implementing the parallelisation of a solver in FEniCS using a distributed memory system. In this case, the mesh is distributed amongst all processes, with each process only storing a sub-mesh on which it performs computation.

7.2. Important Classes and Functions

Certain classes and functions are utilised in virtually every FEniCS program. For example, as one generally deals not only with a PDE, but with a BVP, boundary conditions are needed. Furthermore, if one is to solve a non-linear problem, as in the case of the MOND PDE, a certain initial guess has to be defined for the solver to converge. In every case, one needs to define a domain and respective mesh. In this section the most important classes and functions for the solution of a BVP are listed, with a graphical summary provided in fig. 7.2.1.

7.2.1. Function Space

The `FunctionSpace` class defines the space to which the numerical solution belongs. This space is determined by three inputs:

1. A **mesh**, which provides the spatial domain of computation. This can be given as a built-in mesh, or a more complicated mesh which can be imported from a separate mesh generation library such as `gmsh` or `tetgen`.

2. The **D.O.F. map** defines the family of elements that is chosen. Throughout this work, a Lagrangian element is chosen, which is, in short, equivalent to a polynomial function. This is the type of function that approximates the solution on each cell.

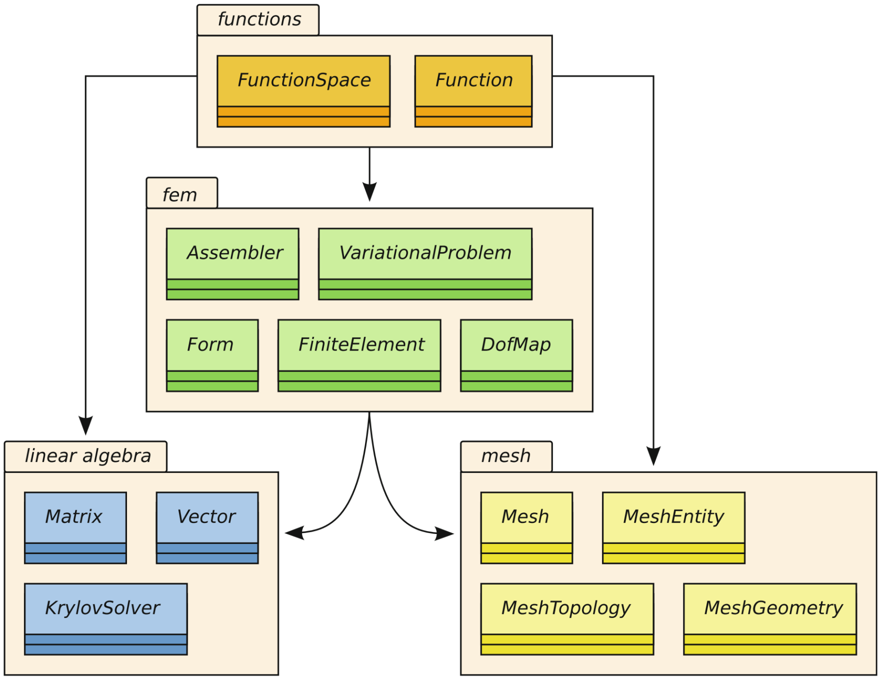

Figure 7.2.1: Schematic representation of the most important classes and functions utilised in the definition and solution of a BVP in FEniCS. The **functions** and **mesh** categories are the most relevant, as it is always necessary to manually define meshes, functions and function spaces. Image from [68].

3. The **degree** of the element: in the case of a Lagrangian element, this gives the degree of the polynomial which will be used to approximate the solution. A higher degree yields a higher accuracy, as well as increasing the computation time and memory required.

7.2.2. Function

A `function` is defined at each point of the `FunctionSpace` to which it belongs. A `function` object stores the value of a solution or initial guess at every discrete point of the domain through a form defined by its D.O.F. map. For example, in the case of a Lagrangian element, a `function` is given by a polynomial of the required degree at each node of the mesh.

7.2.3. Expression

Although not shown in fig. 7.2.1, the `expression` class is of crucial importance in the definition of source terms, as well as initial guesses if one is solving a non-linear problem such as the MOND PDE. Unlike the `function` class, which can only be defined at the nodes of a `FunctionSpace` on a given mesh, an `expression` can be given in terms of spatial coordinates. Moreover, it can be defined in terms of the solution and its derivatives, allowing one to build non-linear PDEs such as the MOND PDE. It is important to note that, when using Python, one can define an `expression` object using C++ syntax. To do so, the input must be a string containing C++ code.

An important advantage of using C++ code in the definition of an `expression` object is the possibility of using inline conditional statements. For example, this can be used to define a spherical source of arbitrary radius, by demanding that its value is non-zero only within a given radius.

7.2.4. FEM Classes

Classes such as `assembler` and `Form` were not directly used in this work, because their definition can be automated when using the `solve` function. However, in order to be able to specify a linear and nonlinear solver different from the FEniCS default, one must use the `VariationalProblem` class. Furthermore, the `FiniteElement` class, which can be used to define the type of cell to be used, such as triangle, square or hexagon in 2D, was also not explicitly used. This is due to the fact that the only elements necessary for basic domain geometries are the so-called simplices, which correspond to triangles in 2D and tetrahedra in 3D. Tetrahedra were used as the domain of interest for galaxy clusters is the sphere.

7.2.5. Linear Algebra

Much like the FEM classes, the linear algebra classes allow for a direct definition of the element matrix and element vector. These are not necessary when using the automated `solve` function, and were hence not explicitly defined throughout this work. On the other hand, the `KrylovSolver` class was utilised to change the linear iterative solver and its preconditioner in order to increase performance for the parallel implementation, as described in chapter 9.

7.2.6. Mesh

The `mesh` class represents the discretised domain over which the solution is obtained. From a `mesh` object, one can obtain information on useful quantities such as the number of cells or vertices present in the mesh, as well as the smallest or largest cell diameter. All of these quantities were used to optimise the performance throughout this work, and will be discussed in detail in the next chapter.

Furthermore, one can define a so=called `meshfunction`, which can set an arbitrary value at each vertex, cell, or facet in the mesh. This can be extremely useful when trying to optimise a solution by mesh refinement, as the mesh can be selectively refined where a criteria is met. Common criteria are the inclusion of a point based on its coordinates, or the error w.r.t. to the anlytical solution being within a given range.

Wide use was made of `meshfunction` functions throughout this work, as will be discussed in the next chapter.

7.2.7. Mesh Entity, Topology and Geometry

Depending on the number of dimensions in which it lives, a mesh can contain various types of `meshentity` objects. The `meshtopology` of the mesh itself defines the connectivity between different `meshentity` objects, for example a list of edges connecting adjacent vertices.

In the case of a 3D domain, the topological objects can be vertices, edges, facets and cells. In short, their topological dimension quantifies the difference in their geometrical dimension from the dimension of the mesh in which they live. As a cell is geometrically 3D, it will have a topological dimension of 0. Similarly, facets, edges and vertices have topological dimensions of 1, 2 and 3 respectively. The use of topological dimensions allows for the definition of a cell independently from the spatial dimension of the domain: in 1D, a line is considered a cell, in 2D a facet is a cell. In all of the aforementioned cases, the cell has topological dimension 0. Through this classification, one can then define local mesh refinement algorithms which can work regardless of the mesh being in 1D, 2D or 3D. On the other hand, the geometrical dimension of a `meshentity` corresponds to the intuitive notion of spatial dimension, with a vertex being 0D, a line 1D, a facet 2D and a cell 3D.

Finally, a `meshentity` can be defined in a general way through its topological dimension. Subtracting the topological dimension of a `meshentity` from the geometrical dimension of the mesh itself will always yield the geometrical dimension of the cell, regardless of the mesh being in 1D, 2D, 3D or above.

Chapter 8

Solving the MOND PDE in FEniCS

This chapter describes the serial implementation of FEM to solve the MOND PDE in FEniCS. First, the notation used in the chapter will be introduced, in terms of computation times and mesh parameters. With the notation in place, the performance will be discussed for meshes in the case of no refinement, uniform refinement and local refinement. We will show that, by utilising an improved version of a local mesh refinement function, we can speed up the total computation time by $O(100)$. All comparisons will be carried out by the use of error measures that will be defined throughout the chapter. Moreover, we will show that the use of local mesh refinement is always advantageous for the number of sources of interest in a galaxy cluster, namely, $O(100)$. Finally, we will analyse the performance of the linear and non-linear solvers available in FEniCS, and compare the speedup they can provide to the one that can be obtained by the use of local mesh refinement.

8.1. Notation

Before giving the description of the results achieved for the different mesh configurations tested, the notation used throughout the chapter will be introduced for clarity. There are five main computation times of interest, all of which are measured in seconds:

- **Generation time**, to generate the mesh: t_{g}.
- **Uniform refinement time**, to uniformly refine the mesh: t_{u};
- **Local refinement time**, to locally refine the mesh: t_{l}
- **Solver time**, to compute the solution of the PDE on the mesh: t_{s};
- **Total time**, including contributions from all times described above, as well as other operations such as explicit communication between processors and plotting of the results: t_{t}.

In addition, there are five quantities which define the scale of the mesh:

- **The number of cells**, equal to the number of elements: n_c;
- **The number of vertices**, equal to the number of points at which the function is evaluated: n_v;
- **The minimum cell diameter**, taken as the minimum diameter of the sphere circumscribing each cell: d_m;
- **The diameter and radius of the domain**, d_t and r_t. Throughout this chapter, the domain size is kept constant.
- We introduce d_t and r_t to obtain the scaled radii for sources and cells.

All the above quantities will be given in relation to a specific mesh. Therefore, they can be thought of as functions of the mesh. As the mesh geometry will be the sphere throughout the report, three quantities are sufficient to uniquely define any mesh. The input (or equally, the argument) to the functions giving the time and scale of the mesh will be the following:

- **Mesh resolution**, the number of mesh elements per spatial direction: α. In the report, the resolution values are in the range $10 \leq \alpha \leq 40$;
- **Uniform mesh refinement**, giving the number of times the mesh is refined uniformly: β. The values for the uniform refinement are in the range $1 \leq \beta \leq 5$;
- **Local mesh refinement**, giving the number of times the mesh is refined locally: γ. The values for the local refinement are in the range $1 \leq \gamma \leq 20$.

Now that all functions and parameters have been defined, an arbitrary function of a given mesh is expressed as:

$$\boxed{f\left(\alpha, \beta, \gamma\right).} \tag{8.1.1}$$

For example, the time required to generate a mesh with resolution $\alpha = 20$, no uniform refinement, and local refinement $\gamma = 6$ is expressed as:

$$t_\mathrm{g}\left(20, 0, 6\right). \tag{8.1.2}$$

To give another example, the number of cells for a mesh with resolution $\alpha = 40$ and no uniform or local mesh refinement is expressed as:

$$n_\mathrm{c}\left(40, 0, 0\right). \tag{8.1.3}$$

8.2. Mesh Parameters

With the notation in place, it is now important to give a more in-depth description of all of the quantities defined above, alongside their contribution to the accuracy of the solution and overall computation time:

1. The generation time t_g: more detailed meshes will generally yield more precise results[1]. However, the time required to generate a mesh[2] will increase with the number of its elements. There are, therefore, different optimal mesh sizes for different design goals, such as the lowest possible error in the solution, or the shortest overall computation time for a given error threshold.

2. The number of vertices n_v: vertices define the degrees of freedom over which the solution is computed. The number of vertices per cell depends on the chosen geometry: in this report, tetrahedra are implied. When using linear elements, one has four vertices, one at each corner of the tetrahedron. However, in certain circumstances a higher order polynomial is needed to express the solution.[3] Higher degree elements were used, for example, when calculating the apparent mass distribution from the potential function described in Part 1. The vertex distribution for tetrahedra of degree 2 and 3 is shown in fig. 8.2.1 [71]. A higher number of vertices for the same geometry generally results in longer computation times and smaller errors.

3. The number of cells n_c: the cells are the elements over which the PDE solution is computed. Their size is of particular relevance, because variations in the solution which are smaller than the cell dimension[4] can generally not be accounted for.

4. The radius of the cell: as mentioned above, features smaller than the cell size are usually not accounted for in the solution. For example, this is the case for source terms with a spatial extension which is considerably smaller than the total domain size. It is hence important that the cells which intersect a source have the correct size. This is an aspect of crucial importance in the study of galaxies and galaxy clusters, where the discrete mass sources are orders of magnitude smaller than the total system.

The aforementioned quantities must be analysed for different meshes in order to obtain simulations which can minimise error, computation time, and required memory.

[1] Strictly speaking, there are exceptions to this behaviour, which will be discussed later in the chapter.

[2] The generation time for a mesh grows with the mesh size, whether the mesh is structured or unstructured. Structured meshes have a predefined layout in space, such as a Cartesian or polar grid. Unstructured meshes have no a priori arrangement, and require a connectivity list to determine adjacent cells. Throughout this work, only unstructured meshes are used.

[3] This is the case if one has to calculate the second derivative of the solution. In that case one needs a degree 3 element.

[4] The dimension of the cell can be given as the diameter of its inscribed or circumscribed sphere, see e.g. [72]. Throughout the rest of this work, the diameter of the cell indicates its circumradius, which is the radius of the sphere within which it is inscribed.

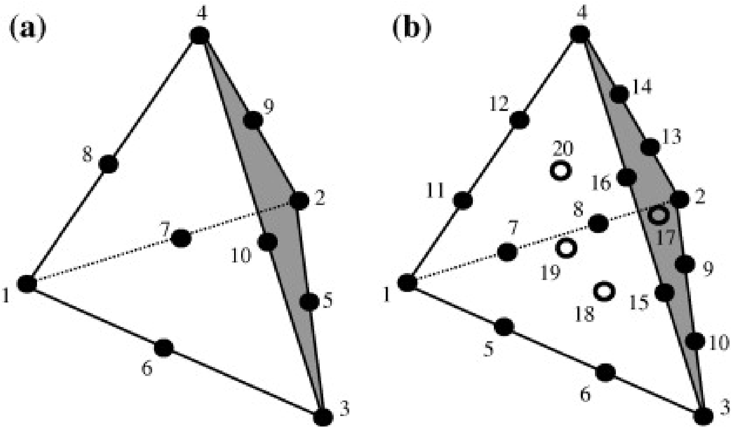

Figure 8.2.1: Second and third order tetrahedra with the respective vertices. Additional vertices are present on the edges for both cases, and also on the faces of the tetrahedron for the third degree element [71].

8.3. Non Refined Meshes

It is important to understand how all necessary quantities evolve when increasing the mesh resolution α. The mesh quantities of interest for a spherical mesh of resolution α are shown in fig. 8.3.1. To minimise the computation time, as well as the error, it is not sufficient to analyse the scaling behaviour of these quantities. Offsets and lower order terms also have to be taken into consideration in order to make realistic comparisons between different configurations.

In order to fit the data points, the `curve_fit` function from the Python `scipy` package was used. The `curve_fit` function utilises an **LS** (**Least Squares**) approximation to fit discrete points to a function of arbitrary form. This allows for the fitting not only of polynomial behaviour, but also of logarithmic and exponential scaling. The latter will be encountered multiple times throughout this chapter. Moreover, the ability to fit data to functions which are not polynomial is particularly advantageous when calculating speedups between different configurations. For example, this allows the fit of a quotient of two polynomials, which in general is not a polynomial. The following functions were found to best fit the data from fig. 8.3.1:

- Generation Time: $t_g \approx 1.658 - 0.228\alpha + 0.013\alpha^2$
- Cell Number (1000s): $n_c \approx 27.092 - 3.949\alpha + 0.172\alpha^2$
- Vertex Number (1000s): $n_v \approx 4.265 - 0.619\alpha + 0.028\alpha^2$
- Smallest Cell Diameter [5]: $d_m/d_t \approx -0.002 + 1.108/\alpha$

It is natural to expect a growth of $O(\alpha^2)$ for the scaling behaviour of t_g, n_c and n_v. This is because a change in resolution for a fixed domain size is analogous to an increase in the

[5] This is a normalised quantity, divided by the domain size.

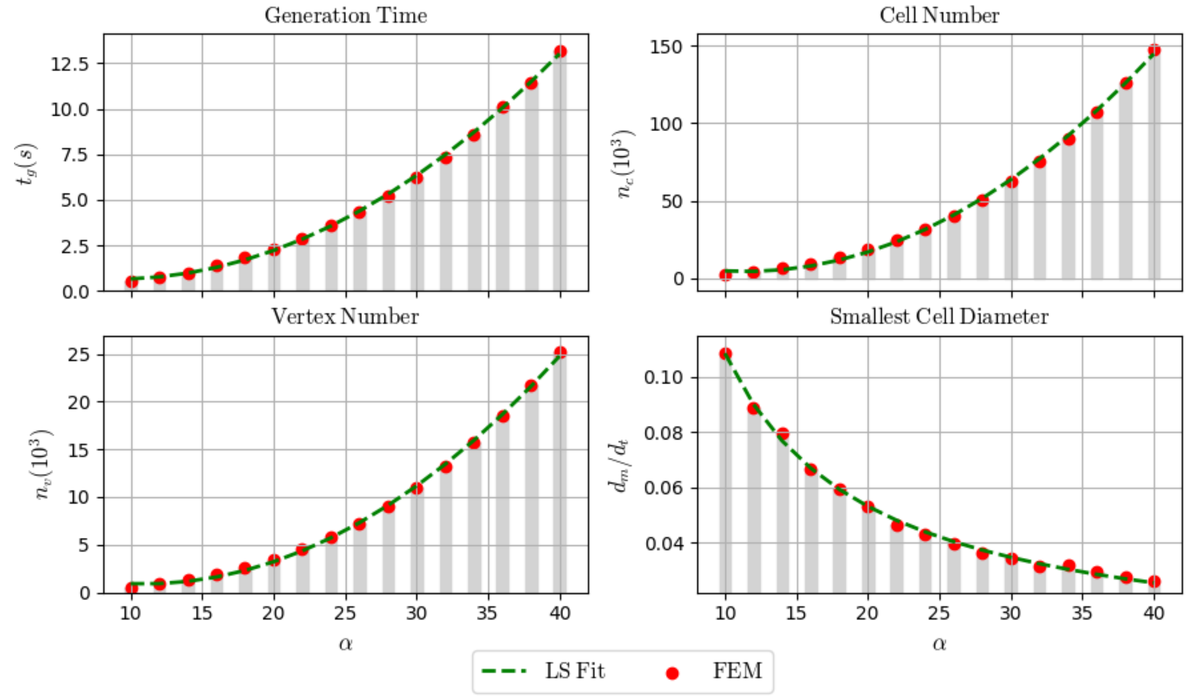

Figure 8.3.1: The dependence of the fundamental mesh quantities on the mesh resolution α. The red dots on the light gray bars represent the data points and are the measured values. The dotted green lines give the non-linear Least Squares approximation to each of the quantities. For generation time, cell and vertex number, there is a quadratic growth. On the other hand, the smallest cell diameter decays inversely with the resolution.

radius with increasing domain size, $\alpha \propto r$. Moreover, changes in the volume of a sphere scale quadratically in r according to:

$$\delta V = V + \delta V - V \approx (r + \delta r)^3 - r^3 \approx r^2 + O\left(\delta r^2\right). \tag{8.3.4}$$

Finally, cell and vertex number both scale with the volume, and the generation time depends directly on the number of cells.

On the other hand, the smallest diameter decreases almost exactly as $1/\alpha$. This is also expected, given that the size of each element is inversely proportional to the mesh resolution for a fixed domain size. One important feature of the increase in mesh resolution can hence be defined.

Both the mesh generation time, which accounts for a non-negligible fraction of the total computation, and the cell number, which determines the memory necessary to store the mesh, grow quadratically in the resolution. On the other hand, the smallest cell diameter, which is related to the error in the solution, only decays as $O\left(1/\alpha\right)$. Therefore, both computation time and memory required grow faster than the quality of the solution with $O\left(\alpha\right)$.

The other component which contributes to the total computation time is given by the PDE solver. The execution time of the solver with respect to the resolution α is shown below in fig. 8.3.2. The LS fit for the solving time is given as:

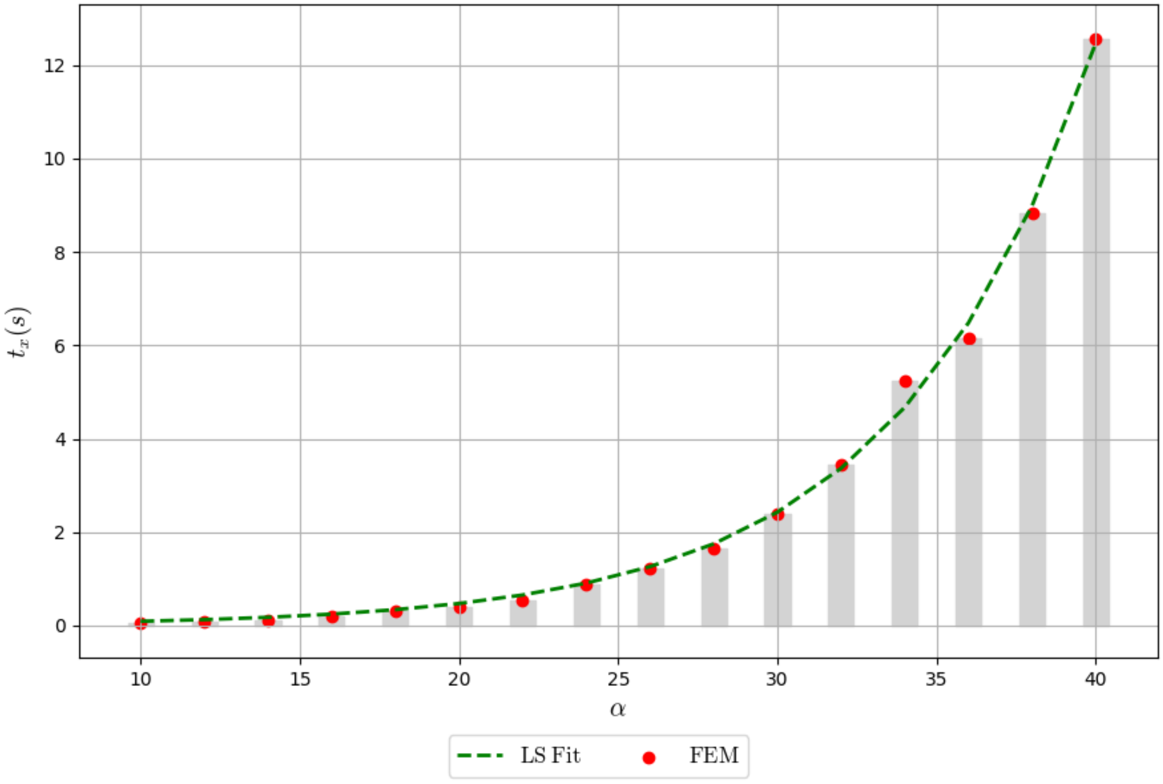

Figure 8.3.2: Solver time against mesh resolution. The time t_s taken by the PDE solver has a quadratic dependence on the mesh resolution α. The red points represent the exact data points, whereas the green dotted line is the best LS fit.

- Solver Time: $t_s \approx 1.791 \cdot 10^{-2}\alpha + 1.636 \cdot 10^{-1}\alpha^2$.

As with the mesh generation time t_g, the solver time t_s scales as $O\left(\alpha^2\right)$, implying the following:

> The total computation time, including mesh generation and PDE solver, scales linearly faster than the solution quality with respect to the mesh resolution α.

8.4. Error Measures

So far, the size of the smallest cell has been utilised as a measure of the quality of the solution. However, in order to make quantitative predictions and allow for comparisons between different meshes, we need to introduce more precise definitions to quantify the error.

8.4.1. Radial Error

One aspect which needs to be considered is the behaviour of the error throughout the domain of computation. This can give a good indication of how to improve the mesh to reduce the error. One of the cases for which the analytical solution is known is that of a point source. Therefore, one can compare the FEM solution to the analytical solution for a sphere of varying size, simulating a spherically symmetric source.

It is of interest to investigate the behaviour of the error as the sphere becomes smaller, better approximating a point source resembling a galaxy in a galaxy cluster. For the simulation of galaxy clusters, it is sufficient to examine the case of a spherical source with a radius 200 times smaller than that of the whole domain. The radial error is a dimensionless quantity defined as the scaled difference between the FEM solution ϕ_{FEM} and the exact analytical solution ϕ, as $|(\phi - \phi_{FEM})/\phi|$. The radial behaviour of the error for sources of decreasing size is shown in fig. 8.4.1. However, a few features of fig. 8.4.1 need to be explained before commenting on the behaviour of the error.

> As the boundary conditions for the PDE are the same for each case, the error tends to a fixed value on the boundary, $E_B \approx 10^{-5}$.

This represents the smallest error which can be achieved as the numerical solution converges to the analytical solution. Conversely, close to the origin at $r \approx 0$ one can see a sudden jump in the value of the error. This is mostly visible in the curves for $\alpha = 10, 16, 22$. This effect can be attributed to the fact that both the solution and the error are evaluated at the vertices of the mesh. Because meshes with a lower resolution have a lower density close to the origin, the radial density of the vertices decreases as $r \to 0$. The same observation can be made close to the boundary of the domain, for $r \to r_t$.

> Close to the boundary, there is a shell of thickness proportional to the domain size divided by the resolution, r_t/α, in which no vertices are present.

Ideally, the solution would converge to the boundary condition as $r \to r_t$. However, the large difference between the values of the error before the aforementioned shell and on the boundary itself is an indication that the solution is not correctly converging to the exact analytical value. For the case of $r_s = r_t/25$ the solution shows good behaviour for mesh resolutions $\alpha \geq 34$. Nonetheless, when $r_s = r_t/200$, which is the case of interest for galaxy clusters, the error close to the source has the same order of magnitude as the solution,

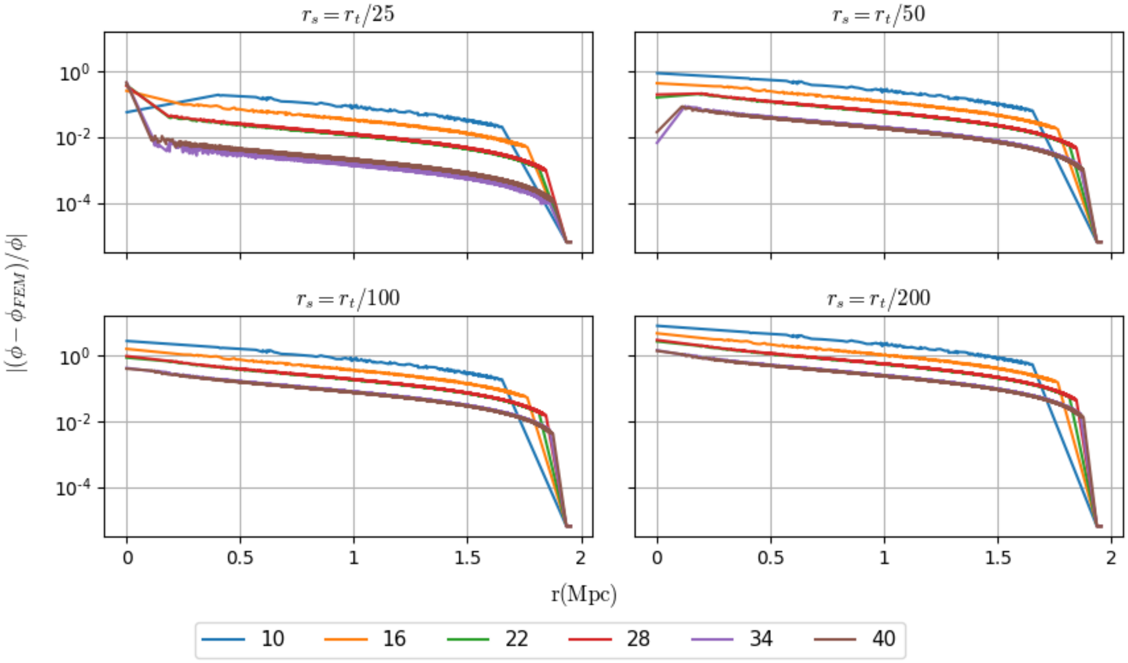

Figure 8.4.1: Relative error in the potential against the radius for various mesh resolutions α, represented by curves of the different colours listed in the legend. Each subplot corresponds to a source with radius r_s given with respect to the domain size r_t. The y-axes of each subplot have a logarithmic scale. Close to the edge of the domain, there is a shell containing no vertices, with a thickness which corresponds to the cell diameter close to the boundary. This is the reason for the step-like behaviour for large values of r. It represents the sudden decrease in the error from the numerical error in the domain to the error on the boundary. The numerical solution on the boundary coincides with the analytical solution, as the BC is given as the analytical solution. Hence the error on the boundary represents the smallest achievable numerical error.

indicating that the solution is not correct. A more quantitative description of the error will be given next.

> Both in the neighbourhood of the source and on the boundary, the FEM solution does not converge to the correct analytical form for the relevant source size of $r_s = r_t/200$.

The low number of vertices close to the origin, and the shell containing no vertices close to the boundary, can be seen more clearly in fig. 8.4.2.

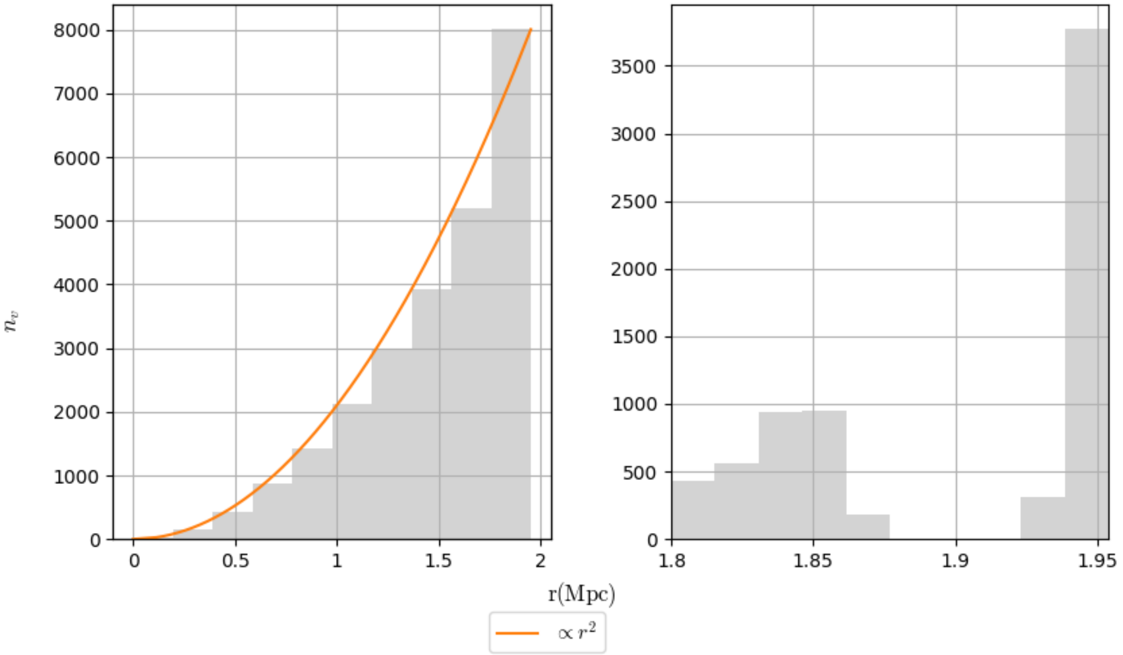

Figure 8.4.2: Histograms for the radial vertex distribution on the mesh. The grey bars represent the number of vertices per bin, while the orange solid line gives a quadratic scaling. In both plots, there are 10 bins. The left histogram covers the whole domain, while the right histogram only covers the region close to the boundary. The quadratic scaling of the vertex number for the domain implies that, close to the origin, very few vertices are present. Conversely, it is clear from the right histogram that close to the boundary there is a shell within which no vertices are present, which causes the step behaviour near the boundary in the radial error plots of fig. 8.4.1.

8.4.2. Error Functional

In order to carry out a quantitative comparison between two meshes, with their quality defined as the error of the solution w.r.t. the exact analytical form, the error itself must be defined more precisely.

A common measure of the error of a numerical solution is the **Error Functional**, which is obtained by integrating the square of the difference between numerical and analytic solutions over the whole domain, and then taking the square root. This ensures that the same weight is given to overestimation and underestimation errors. This error E_F can be expressed as:

$$E_F = \sqrt{\int_\Omega \left(\phi_E - \phi_{FEM}\right)^2 \, \mathrm{d}\Omega}. \tag{8.4.5}$$

However, we should note that E_F does not necessarily reflect the local quality of a solution around a given point, such as the origin. This is due to the fact that errors at all points of the domain are given the same weight.

8.4.3. Maximum Error and Weighted Error

In the case of a point source in the origin, as both the solution and its derivatives decrease with the distance from the source, it is desirable to determine what the error is close to the source. It can be seen from fig. 8.4.1 that for $r_s = r_t/200$, the case of interest for galaxy clusters, the error is a decreasing function of the radius. To quantify the error around the source, we can then look at the maximum value of the error, which can indicate how well the numerical solution matches the analytical counterpart in the region of most interest. The maximum error E_{Max} is defined as:

$$\boxed{E_{Max} = \text{Max}\left[\left|\left(\phi - \phi_{FEM}\right)/\phi\right|\right].} \tag{8.4.6}$$

As in the case of the relative radial error, E_{Max} is a dimensionless quantity. However, the maximum error can also not be used as the sole quantity to define the error, as we want to ensure that the solution is consistent throughout the whole domain. It is thus useful to introduce a **weighted error** E_W as the product between the Error Functional E_F and the maximum error E_{Max}:

$$E_W = E_F \cdot E_{Max}. \tag{8.4.7}$$

As E_{Max} is dimensionless, E_W has the same units as the error functional E_F. Ultimately, this error accounts for the quality of the solution throughout the whole domain, but also reflects how well the solution behaves close to the source itself. All these quantities are shown in fig. 8.4.3 for the case $r_s = r_t/200$. As can be seen in fig. 8.4.3, none of the error measures are exactly a decreasing function of the resolution. All error measures show little decrease for $20 < \alpha < 30$ and $32 < \alpha < 40$. This is particularly true for E_{Max}, which is given in fig. 8.4.3 with a linear y-scale, whereas both E_F and E_W have a logarithmic y-scale. It is important to point out these plateaus, as they will not be present in the case of local mesh refinement, introduced in the next subsection. It is interesting to note that $E_W > E_F$ for all resolutions α. This reflects the following:

For all resolutions $10 \leq \alpha \leq 40$, $E_{Max} > 1$. This indicates that, for all values of α, there are points close to the source where the error is of the same order of magnitude as the solution.

It is hence clear that, even with large values for the resolution α, the numerical solution ϕ_{FEM} does not suitably reproduce the exact analytical solution ϕ. It should also be noted that the minimum value for both E_{Max} and E_W is achieved by the largest resolution, $\alpha = 40$. Qualitatively, this indicates that further increasing the mesh would reduce the error. However, it should be mentioned that for $34 < \alpha < 40$ neither E_F nor E_{Max} show the sharp exponential decrease that was found for $\alpha < 20$.

The value $\alpha = 20$ is a local optimum in the quality of the solution w.r.t. to the generation time. For this reason, it represents a good starting point if we want to selectively refine the mesh. This observation will be used throughout the next subsections.

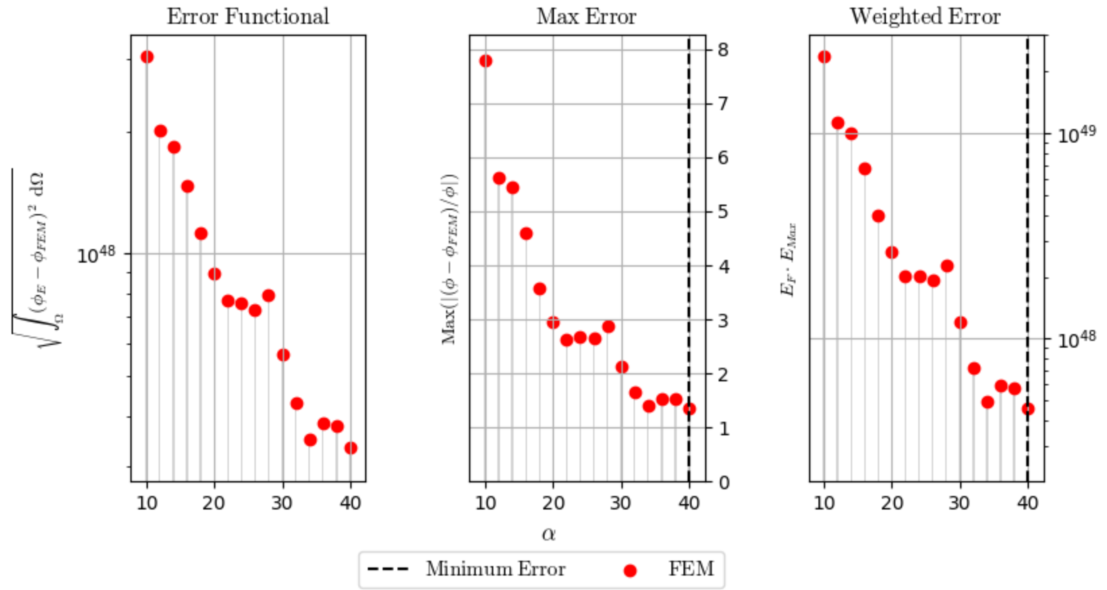

Figure 8.4.3: Error measures against the mesh resolution α for source radius $r_s = r_t/200$. The graphs for the error functional and weighted error have the same units and have y-axes in a logarithmic scale. The max error is dimensionless and has a linear y-scale. The red points represent the results from FEM, while the black vertical dashed lines indicate the minimum value for the Max Error and the Weighted Error. Although the behaviours of each error seem comparable, it must be noted that the max error has a linear scale.

Fig. 8.4.3 then shows that, after a critical value $\alpha_c = 20$, increasing the mesh resolution has diminishing returns. Moreover, as shown in fig. 8.3.2, the mesh generation time t_g increases quadratically in the resolution.

Memory is also an issue: for a large scale computation in which the sources are orders of magnitude smaller than the domain, for example the case of galaxy clusters, there is the need for a mesh of very high resolution. However, as shown below in fig. 8.4.4, the memory required for a mesh rapidly increases with its size. The Least Squares fit for the mesh size in bytes M_α w.r.t. the resolution is given by:

- $M_\alpha \approx 284.594 \cdot 10^4 - 40.709 \cdot 10^4 \alpha + 1.716 \cdot 10^4 \alpha^2$.

One should hence highlight the fact that:

Generation and solver times, as well as cell/vertex numbers and memory required for a mesh, all scale quadratically with the resolution α.

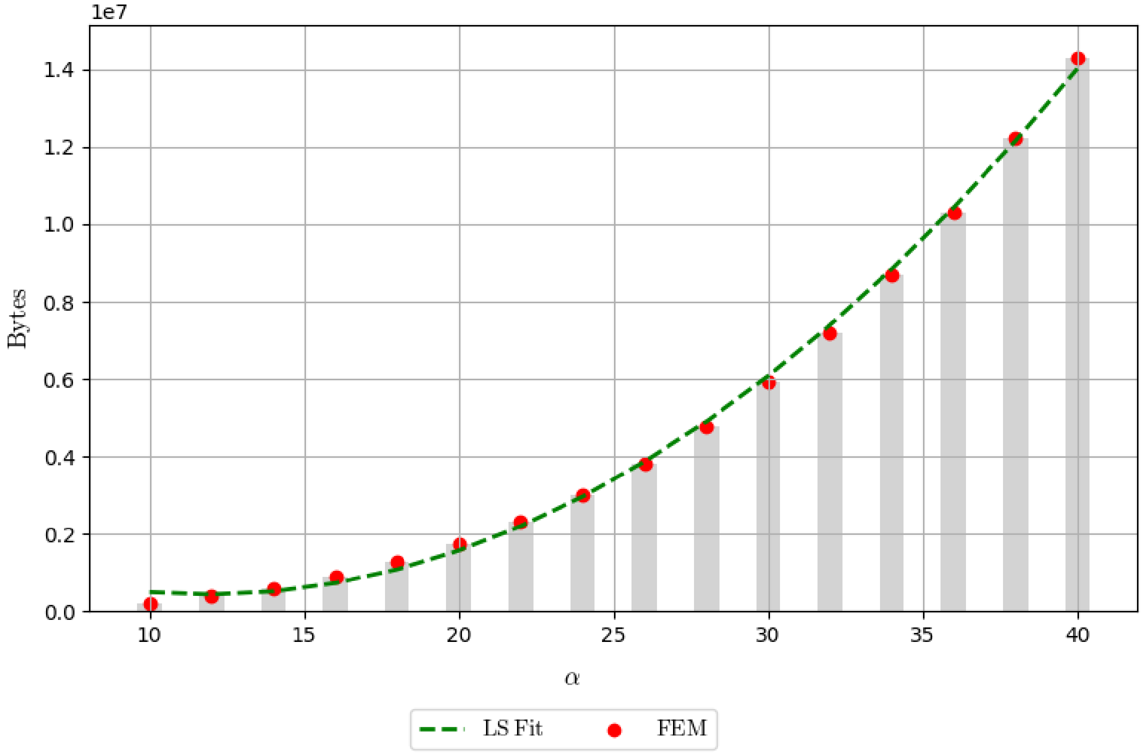

Figure 8.4.4: Memory in bytes required for a mesh of resolution α. The red dot represents the FEM data, whereas the green dotted line gives the Least Squares fit to the data. It can be seen that the scaling has a dominant quadratic component. The size is measured in bytes.

8.5. Uniform Mesh Refinement

A widely used technique to improve the quality of a FEM solution is to increase the number of cells of a pre-existing mesh, through the process of **Mesh Refinement** (see for example [73]). This section will introduce the algorithm used for mesh refinement in FEniCS, and the impact of uniform mesh refinement on computation time and error in the solution.

8.5.1. The Plaza algorithm

Many mesh refinement algorithms exist, but the following discussion will be limited to the algorithm used in FEniCS, called the **8-Tetrahedra Longest-Edge Partition**, introduced by Plaza and Rivara in [74]. This algorithm is based on splitting each tetrahedron element into 8 smaller tetrahedra while maximising the quality of the newly produced elements, as shown schematically in fig. 8.5.1.

As the Plaza algorithm itself ensures a good quality of the generated sub-elements,

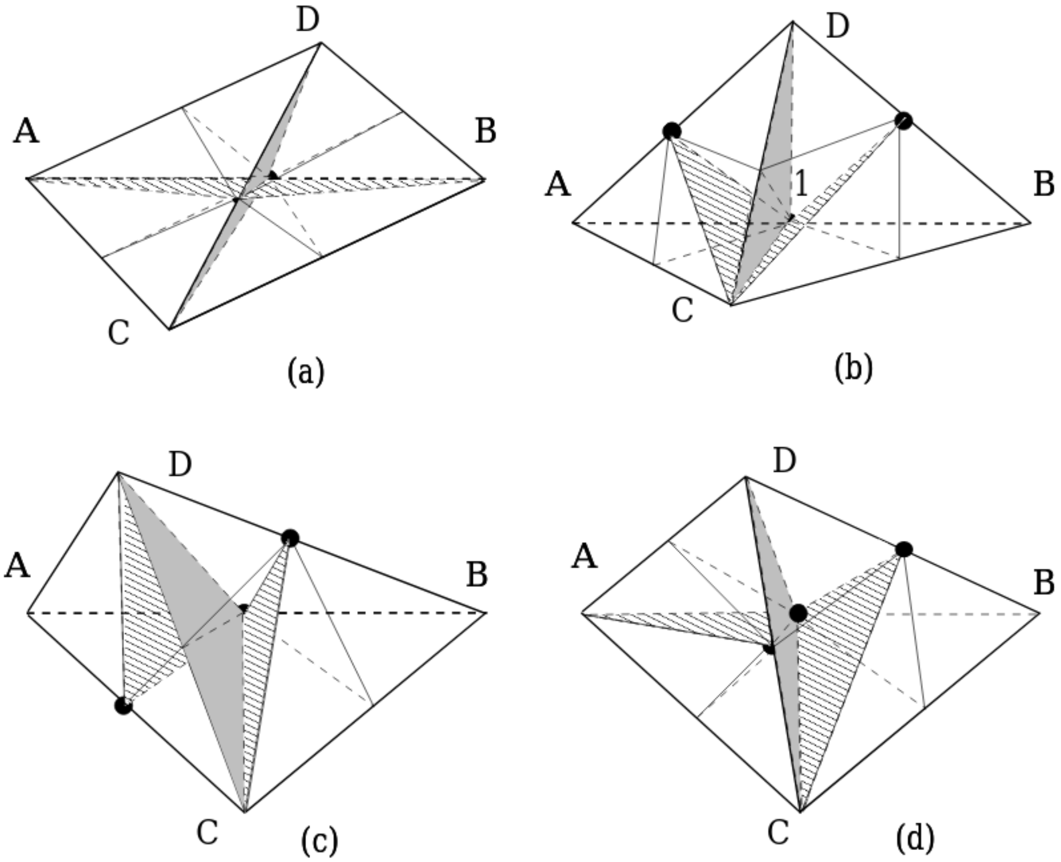

Figure 8.5.1: Four different ways of splitting a tetrahedron element into 8 sub-elements while maintaining good element quality by using the 8-Tetrahedra Longest-Edge Partition algorithm designed by Plaza. Figure from [74].

it is not necessary to analyse the quality of all the elements after each mesh refinement. Nonetheless, it is useful to define what is meant exactly by element quality. Although different measures exist, as explained in [57], probably the most intuitive is given by the ratio of the radii of the circumscribing and inscribing spheres for each element.[6] This ratio gives a measure of how close the element is to a regular tetrahedron or, in other words, whether it is particularly flat along a given axis (bad element) or whether all of the sides have lengths comparable to the height w.r.t. any of the planes (good element).[7] A visual example of what is usually considered a bad element is given in fig. 8.5.2.

Generally, if the quality of the element is kept constant during mesh refinement, a solution will have a lower error for a more refined mesh. Nonetheless, generation time and

[6] The circumscribing sphere is the sphere that contains all the element vertices on its surface. The inscribed sphere is the sphere that has tangent points to all element facets on its surface.

[7] Numerically, the ratio is a number between 0 and 1/3, with 1/3 corresponding to a regular tetrahedron, a cell of the highest quality.

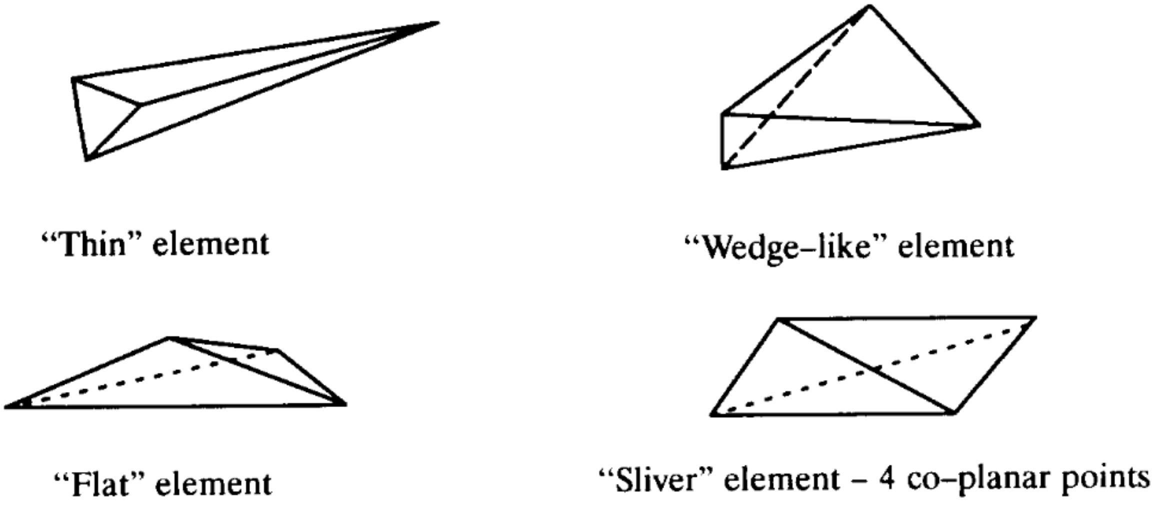

Figure 8.5.2: Four different examples of what is generally considered to be a bad element in a refined mesh. Thin, wedge and flat elements provide poor approximations to the solutions when using linear elements because the points at which the solution is evaluated are not equally distributed in the 3 spatial directions. Sliver elements are the limit of 4 points belonging to the same plane, and cannot be present in the mesh. Figure from [74].

memory required also scale with the refinement times β. Since the refinement algorithm splits each tetrahedron into 8 new elements, the cells increase exponentially with β, the number of refinements. This behaviour should also be expected for the number of vertices n_v and the generation time t_g.

The scaling of all these quantities, to be compared with a mesh generated with a given resolution without refinement, are given in fig. 8.5.3. As expected, all quantities shown in fig. 8.5.3 have an exponential behaviour. More precisely, the Least Squares fits are found to be:

- Generation time t_g: $6.989 \cdot 10^{-4} e^{2.373\beta}$
- Cell Number n_c: $2.284 \cdot e^{2.079\beta}$
- Vertex Number n_v: $1.071 + 0.403 e^{2.07\beta}$
- Minimum Cell Diameter r_c/r_t: $0.127 e^{-0.838\beta}$

It must be mentioned that the mesh refinement was conducted on an initial mesh with resolution $\alpha = 10$. Through the notation introduced at the start of the chapter, this is a $(10, 0, 0)$ mesh. A direct comparison can now be made between a mesh generated with a given resolution α and a mesh of resolution $\alpha = 10$ refined β times, expressed as $(10, \beta, 0)$. It is interesting to compare a non-refined $(40, 0, 0)$ mesh to a $(10, 2, 0)$ uniformly refined mesh. This is a good comparison because one has a similar number of cells:

$$n_c\,(40, 0, 0) = 146339, \quad n_c\,(10, 2, 0) = 144200 \implies n_c\,(40, 0, 0) \approx n_c\,(10, 2, 0)\,. \qquad (8.5.8)$$

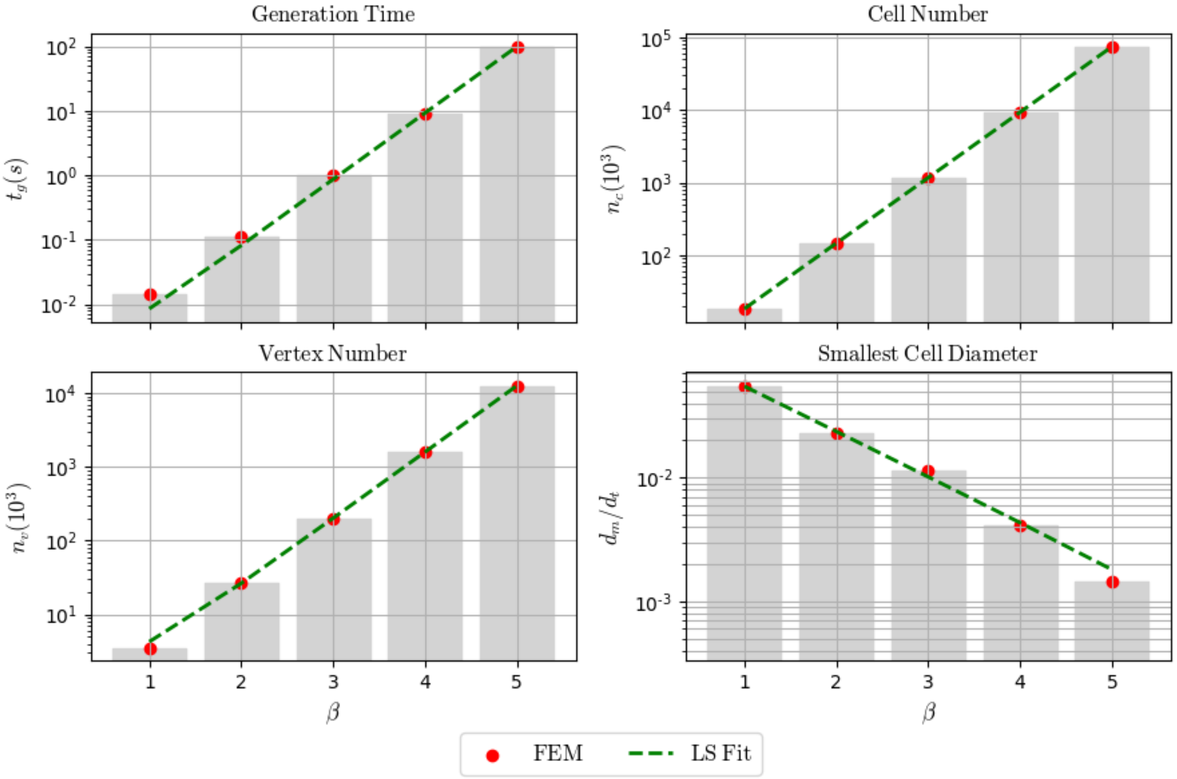

Figure 8.5.3: Mesh quantities w.r.t. times the mesh is uniformly refined, β. The red dots represent the FEM data, whereas the green dashed lines give the best fit. All the y-axes have a logarithmic scale, showing that all quantities have exponential scaling.

However, when comparing the overall generation time one needs to be aware of the following: for the non refined mesh, one can directly read from fig. 8.3.1 that $t_g\left(40,0,0\right) = 13.338s$. On the other hand, to calculate the overall time taken to obtain the twice refined mesh, one needs to account for the generation time of the initial mesh, and add the times of each successive refinement, which generate a mesh with a value β increased by 1. Generally, the time necessary to produce an $(\alpha, \beta, 0)$ uniformly refined mesh can be obtained as:

$$\boxed{t_g\left(\alpha,\beta,0\right) = t_g\left(\alpha,0,0\right) + \sum_{\tilde{\beta}=1}^{\beta} t_g\left(\alpha,\tilde{\beta},0\right).} \tag{8.5.9}$$

In the above, α indicates the resolution of the initial refined mesh, and $\tilde{\beta}$ the number of each successive uniform refinement. Using (8.5.9) for a $(10,2,0)$ mesh gives $t_g\left(10,2,0\right) =$

$0.766s$. This implies that the speedup s_g in the overall mesh generation is given by:

$$\boxed{s_g = \frac{t_g\,(40,0,0)}{t_g\,(10,2,0)} = \frac{13.338}{0.766} \approx 17.4.} \tag{8.5.10}$$

With a substantial speedup such as the one obtained above, it is important to check that the smallest cell has approximately the same size, as a first order check of solution quality. The minimum cell diameters for the two cases are:

$$(r_c/r_t)\,(40,0,0) = 0.0257 \approx (r_c/r_t)\,(10,2,0) = 0.0238. \tag{8.5.11}$$

The following is then clear:

> For approximately equal cell number n_c and minimum cell diameter r_c/r_t, it can be advantageous to generate a mesh with lower resolution and consequently refine this mesh, rather than generating a larger mesh initially.

This behaviour might seem surprising, as the generation time for refined meshes grows exponentially. However, eq. 8.5.10 shows that scaling behaviour alone is not sufficient to determine computational efficiency, as one needs to take into consideration the scale of the problem at hand. Nevertheless, despite the speedup obtained by utilising mesh refinement, one must be cautious in using such an approach: for higher mesh sizes, scaling behaviour becomes important and one needs to consider that the memory required for a refined mesh will scale exponentially. A design goal could hence be formulated as follows:

> In order to minimise both error and computation time, one should make use of mesh refinement, but devise a method which avoids the exponential growth that this implies for both t_g and n_c.

8.5.2. Local Mesh Refinement

The solution to the above-mentioned problem is to use **local mesh refinement**. As the name suggests, local mesh refinement consists of only refining the mesh where a certain criterion is met. This criterion could be any combination of the following (see, for example, [75]):

- Distance from a given point in the domain;
- Value of the solution or its derivatives;
- Difference between the numerical and analytical solutions.

As was shown in fig. 8.4.1, the error is highest close to the center of the domain. More generally, when the size of the source is considerably smaller than the size of the cell, the error is maximal close to the source. Consequently, the following can be observed:

> Due to nonlinearity, the analytic solution for the MOND PDE is not known for multiple point sources. Therefore, local mesh refinement for problems with multiple sources should be carried out in the neighbourhood of each discrete source. This is because, in the case of a single source, where the analytic solution is known, the neighbourhood of the source is where the error is highest. Moreover, it is here that both the solution and its derivatives vary the fastest.

It is clear from the above that all criteria for local mesh refinement are met close to the sources. The way of implementing any function depending on the mesh in FEniCS is given by the `Mesh Function`, as explained in [76]. The `Mesh Function` is a specific type of function which can be defined and evaluated on a set of `Mesh Entities`. `Mesh Entities` are objects that represent the various components of a FEniCS mesh, depending on their geometrical and topological size. The `Mesh Entity` that was used to build a local refinement function is the `Cell`. This matches exactly the concept of a cell in a FEM domain, and it is made up of internal nodes and a connectivity list indicating the connection between different nodes by means of edges.[8]

`Mesh Functions` are extremely useful as they make it possible to set markers on each object corresponding to the chosen `Mesh Entity`. The markers can then be used by built-in functions in FEniCS to carry out tasks such as mesh refinement through the `refine` function. In particular, for a 3D mesh, the `refine` function accepts as an optional parameter a boolean `Mesh Function` indicating which cells should be refined and which should not, hence acting as a boolean mask.

8.5.3. Refinement by Cell Distance

As previously mentioned, the goal is to refine the mesh exclusively around each source. An obvious way to approach this is described in [77] as follows: define a `Mesh Function` marking each cell whose midpoint lies within a given radius, and refine the mesh for each cell that has been marked. Although the approach is an initial improvement over refining the entire mesh, it presents the following issues:

1. The refinement of a sub-volume of the mesh will have the same exponential scaling behaviour as the refinement of the entire mesh;

2. For initially coarse meshes, the method might fail, as it is possible that no cell midpoint falls within the required refinement radius;

[8] It is useful to note that `Vertex` and `Edge` are also a `Mesh Entity`, of geometrical dimension 1 and 2 respectively.

3. The determination of the midpoint of each cell requires a loop over every cell present in the mesh, as cells cannot be ordered according to their distance from an arbitrary point without looping over them with another `Mesh Function`;

4. When refining the mesh for multiple sources, the distance must be calculated from each cell to each source. As the cell number increases exponentially in the refinements, so will the computation time.

8.5.4. Refinement by Containing Cell

Given the issues outlined above with regard to refining the mesh based on the distance of the source from a point in the domain, a better approach is the following: start with a sufficiently coarse mesh and exclusively refine the cells that contain a source, rather than computing their relative distance. The inputs needed for such a function are:

- The mesh to be refined: `mesh`;

- A list of **source locations**: `location`. In order to be correctly evaluated by the `Mesh Function`, the locations need to belong to the `Point` class of FEniCS;

- The number of times the mesh should be locally refined: `gamma`.

The output of the function is the refined mesh. The **MWE** (**Minimal Working Example**) of this initial implementation is the following:

```
def local_refinement (mesh, location, gamma):
    '''Function to refine mesh locally, based on the cells containing each source
    '''

    #Looping over refinement an arbitrary amount of times
    for i in range(gamma):

        #Looping over each source location
        for j, source in enumerate(location):

            #Declaring the boolean Mesh Function for a cell, entity of dimension 3
            cell_to_refine = MeshFunction("bool", mesh, 3)

            #Initialising all markers to false
            cell_to_refine.set_all(False)

            #Iterating over all cells in the mesh
            for cell in cells(mesh):

                #Checking if cell contains the source
                if cell.contains(source):

                    cell_to_refine[cell] = True

            #Refining the mesh only where the markers are True
            mesh = refine(mesh, cell_to_refine)

    #returning the refined mesh
    return mesh
```

Although clearly not optimal, this initial implementation solves the first two problems present in the approach described in [77]:

- The scaling does not depend on the volume, but rather on the amount of sources. Although in theory the growth will still be exponential in the number of local refinements γ, since the number of sources n_s is orders of magnitude smaller than the number of cells n_c, the function should grow polynomially both w.r.t. to the refinement times γ and the source number n_s.
- No midpoint calculation is required, and the function will correctly refine meshes of any initial resolution, including particularly coarse meshes.

Nevertheless, in the above approach it is still necessary to loop over each cell for each source. A clear initial path to improvement is to first store the indices of the cells containing a source. The modified approach differs in the following aspects:

- Only one `Mesh Function` has to be created, valid for all sources, rather than each source having its individual `Mesh Function`.
- The loop to set the `Mesh Function`, which was previously over each cell, is now carried out only for the cells that contain a source.
- The `refine` function is only called γ times, rather than for each source as previously, $\gamma \cdot n_s$ times. Therefore, the speedup w.r.t. the initial implementation increases with n_s.

An MWE of this approach is as follows:

```
def modified_refinement (mesh, location, gamma):
    '''Modified Mesh Refinement Function
    '''

    for i in range(gamma):

        #Declaring Boolean Mesh Function
        contain_function = MeshFunction("bool", mesh, 3)

        #Initialising function to False everywhere
        contain_function.set_all(False)

        #Initialising empty array for the indices of cells containing a source
        cell_index = np.zeros((source_number, 1))

        for j, source in enumerate(location):

            #List comprehension, True for cells containing a source
            contain_list = [cell.contains(source) for cell in cells(mesh)]

            #Converting list to a numpy array
            contain_numpy = np.fromiter(contain_list, float, mesh.num_cells())

            #Then setting the MeshFunction True at those cells
            cell_index[j] = np.nonzero(contain_numpy)[0]

        for cell_containing in cell_index:

            contain_function[cell_containing] = True

        mesh = refine(mesh, contain_function)

    #Returning the refined mesh
    return mesh
```

The problem with the above code is that it is still necessary to loop over each cell in the mesh for every source because, as previously explained, the cell indices cannot be organised without making use of a separate `Mesh Function`.

However, FEniCS provides the `intersect` function to directly calculate the intersection between a given point in the domain and `Mesh Entity` of any type, such as a cell. Although this is a costly operation, it makes it possible to completely avoid the explicit loop over each cell in the mesh, as it only requires a single loop over each source. One can thus loop exclusively over the cells containing a source, as described above. The MWE for this improved implementation is the following:

```
def improved_refinement (mesh, location, beta):
    '''Improved Mesh Refinement Function
    '''

    for i in range(beta):

        #Declaring Boolean Mesh Function
        contain_function = MeshFunction("bool", mesh, 3)

        #Setting function to False everywhere
        contain_function.set_all(False)

        #List comprehension for the cells containing a source
        intersect_list = [intersect(mesh, source).intersected_cells() for
        source in location]

        #Setting the cell function contain_function to true for each cell
        #containing a source
        for cell_index in intersect_list:

            contain_function[cell_index[0]] = True

        #Refining the mesh only for cells that contain a source
        mesh = refine(mesh, contain_function)

    #returning the refined mesh
    return mesh
```

Although the largest speedups are expected to appear when refining multiple sources, fig. 8.5.4 shows that that the computation time also decreases noticeably for a single source for a mesh with configuration $(20, 0, \gamma)$. The computation times for each of the functions can be fitted by a second order polynomial. More specifically the generation times are:

- Initial implementation: ${t_g}^i \approx 0.002 + 0.0692\gamma + 0.003\gamma^2$
- First modification: ${t_g}^m \approx -0.027 + 0.06\gamma + 0.003\gamma^2$
- Final modification: ${t_g}^f \approx -0.038 + 0.052\gamma + 0.002\gamma^2$

Although the scaling is $O(\gamma^2)$ for all implementations, the second order coefficient is lower in the final implementation by $\approx 30\%$. For $\gamma = 10$ this results in a reduction in computation time of 13% and 32% respectively for the modified and improved functions w.r.t. the initial function. That said, one must remember that all three functions implement exactly the same algorithm. The growth in the cell number n_c and the reduction of the smallest cell diameter d_m are hence common to all the implementations described, and are shown in fig. 8.5.5. The fitted behaviours are found to be:

- Cell number: $n_c = 20.493 + 0.513\gamma + 0.04\gamma^2$

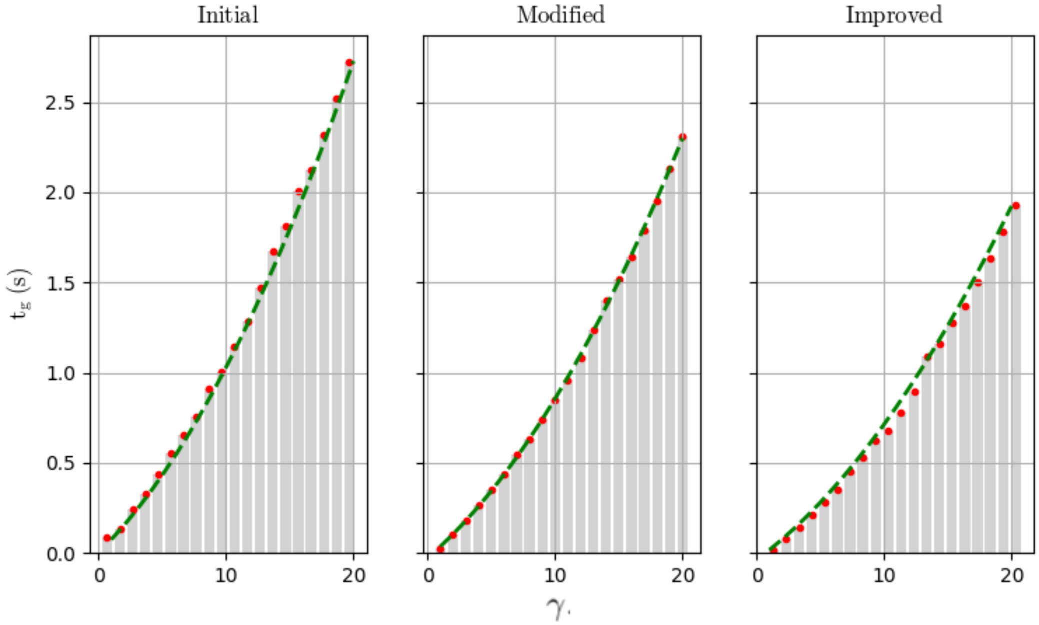

Figure 8.5.4: Time taken for different refinement functions to refine a single source, against the local refinement times γ. The red dots represent the FEM data, whereas the green dashed line shows the Least Squares fit. All the refinement times have a quadratic scaling. However, w.r.t. the initial implementation, for the median value $\gamma = 10$ the modified refinement provides a speedup of $\sigma \approx 12\%$ whereas the improved refinement function provides a speedup of $\approx 32\%$.

- Smallest cell diameter: $d_m/d_t = 0.142e^{-0.693\gamma}$.

It can be seen that the local mesh refinement function manages to reproduce two important features, which could not both be simultaneously obtained through standard mesh generation or uniform mesh refinement: polynomially increasing cell number n_c and exponentially decreasing cell diameter d_m.

As d_m/d_t is a good indicator of the error, one can infer that both the computation time and overall mesh size now grow slower than the mesh quality. However, d_m/d_t is not sufficient to make quantitative observations on the error across the whole domain. Therefore, the relative radial error was again calculated as done for fig. 8.4.1, and is shown in fig. 8.5.6. When comparing fig. 8.5.6 to fig. 8.4.1, which represents the same quantity for a non refined mesh, several qualitative statements can be made for the case of local refinement::

- The curve corresponding to $\gamma = 0$ has an error which in the case of $r_s/r_t = 100, 200$ is of the order of magnitude of the solution close to the source. This is never the case for $\gamma \geq 3$. This implies that the even for low values of γ, the numerical solution converges to the exact analytic solution.

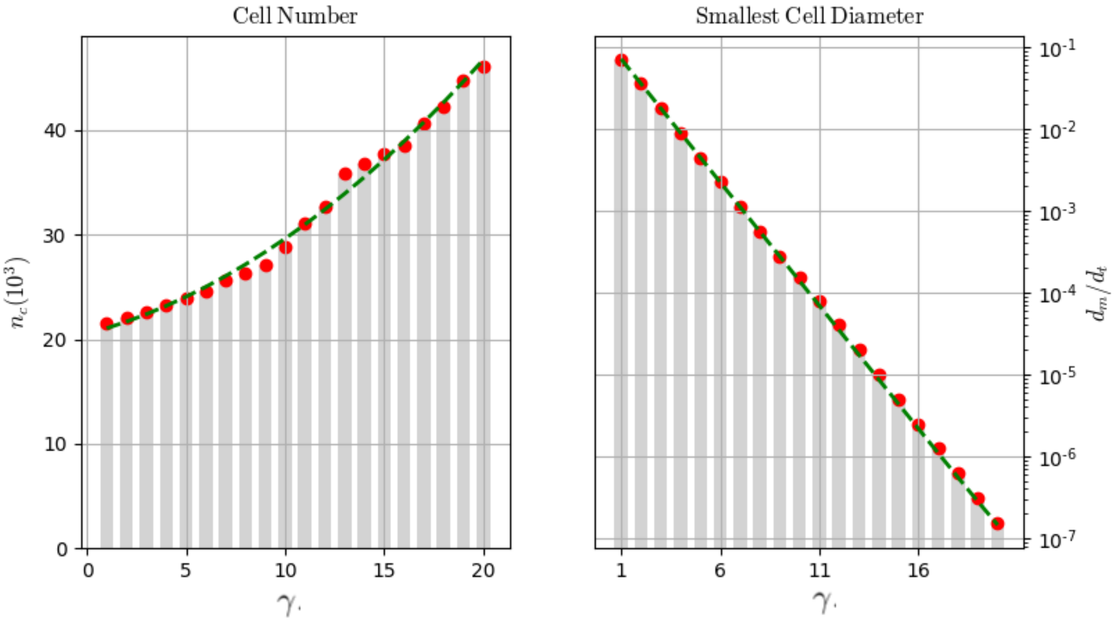

Figure 8.5.5: Mesh quantities as a function of the local refinement times γ for the local refinement algorithm. The red points represent the FEM data, whereas the green dashed lines give the Least Squares fit. It can be seen that the number of cells n_c increases quadratically. This is an improvement over the case of uniform refinement, where the growth was exponential in β. On the other hand, the smallest cell diameter still decreases exponentially, as desirable to reduce the error.

- The shell containing no vertices close to the domain boundary remains the same regardless of the value of γ since the mesh is only refined close to the source. Nonetheless, for $\gamma \geq 6$ and for all source radii, the difference in value between the numerical solution at the edge of the shell and the exact solution on the boundary is at most $O(10)$, compared to $O(10^4)$ for the non refined meshes in fig. 8.4.1 for $r_s/r_t = 200$. This indicates that the mesh refinement close to the source improves the quality of the solution throughout the domain, not only in the neighbourhood of the source.

- For all source sizes the error does not seem to decrease for $\gamma > 6$. Unlike the case of no refinement, where no local minimum was present, it appears that $\gamma = 6$ provides an optimal value for local refinement.

- The error becomes smaller with decreasing source size and increasing γ. Moreover, for the case of most interest i.e. $r_s/r_t = 200$, the radial error is lower than 10^{-2} throughout the whole domain. This indicates that the solution converges faster for a smaller source, which is the desired result when implementing a point source.

Although the aforementioned qualitative observations are encouraging, it is again necessary to quantitatively define the error in order to make a comparison between refined and

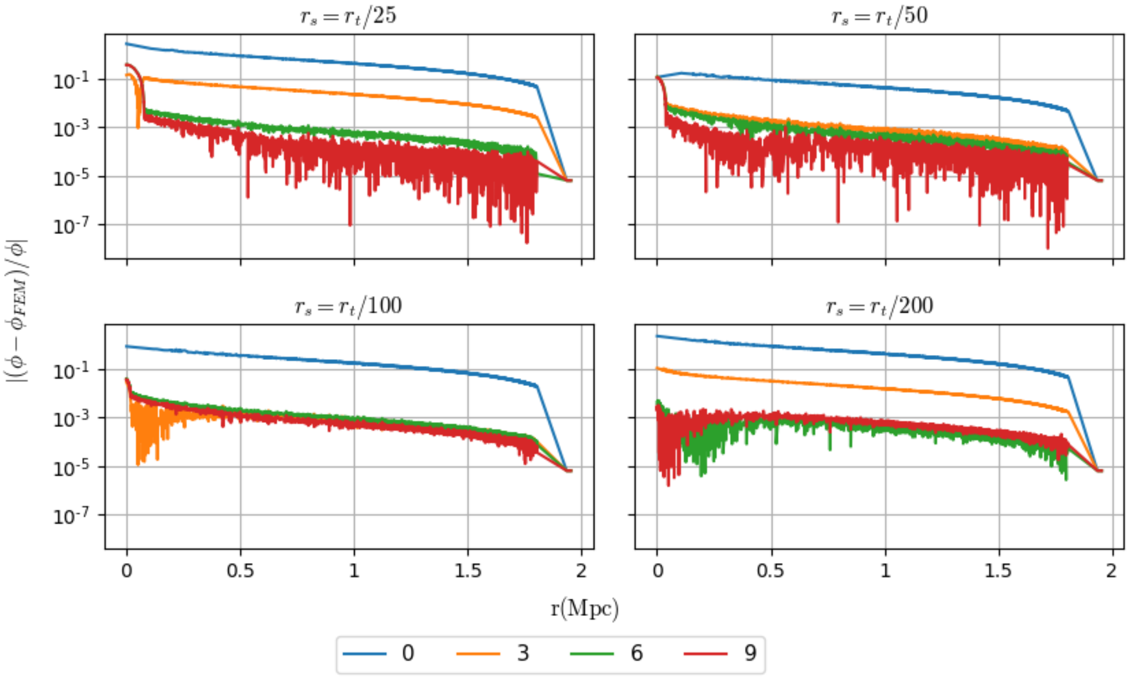

Figure 8.5.6: Relative radial error for a single source placed at the origin against the local refinement γ. The y-axes are logarithmic in all plots. Different colours represent different values of γ. The initial mesh before refinement, corresponding to the curve $\gamma = 0$, was generated with $\alpha = 20$. It can be seen that, for all source radii r_s, the error does not decrease for $\gamma > 6$. Moreover, the large error between the solution in the domain and on the boundary is reduced from $O(10^4)$ to $O(10)$ when compared to the case of non-refined meshes.

non refined meshes. To do so, the error functional, max error and weighted error were calculated for the refined meshes, as was previously done for the non refined meshes in fig. 8.4.1. The results for the locally refined meshes are shown in fig. 8.5.7.

As expected from the analysis of fig. 8.5.6, the error functional reaches its minimum for $\gamma = 6$. On the other hand, the maximum error is lowest for $\gamma = 9$, after which it appears to stabilise around $3 \cdot 10^{-3}$. As a result, the weighted error has a minimum at $\gamma = 6$, after which it also remains approximately constant. The following can then be observed:

- Compared to the case of no refinement, the minimum value achieved for the error functional is $\approx 10^3$ lower.

- For $\gamma > 2$, the maximum error is smaller than the value of the solution. Moreover, for $\gamma \geq 6$ it is smaller by at least two orders of magnitude.

- Overall, the maximum error can always be made smaller than unity, implying that

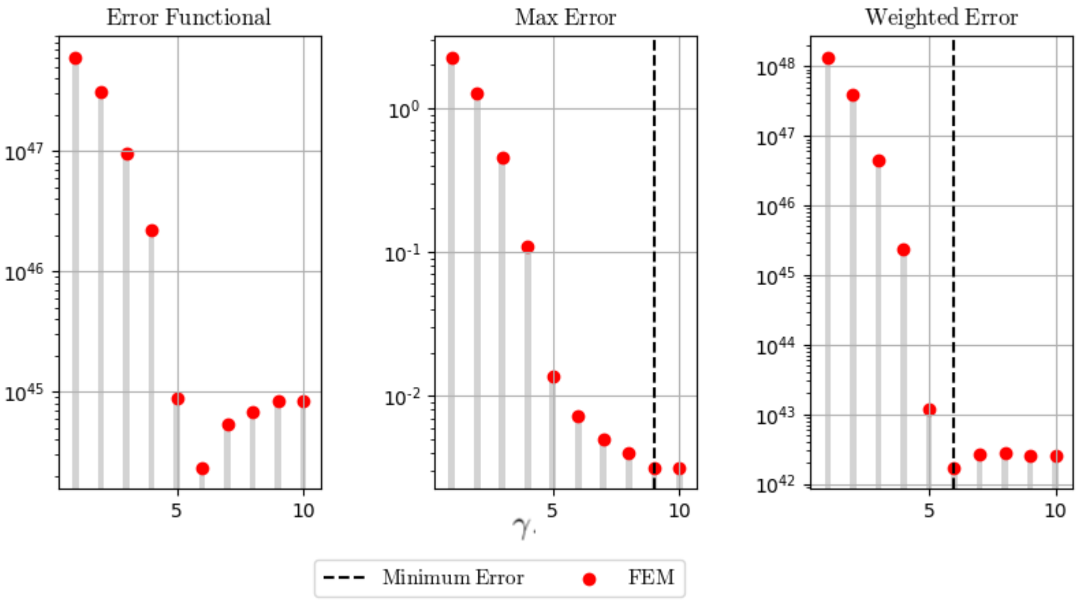

Figure 8.5.7: Error measures for a mesh with local refinement γ. The red points show the FEM data whereas the vertical dashed black lines give the minimum value for the error. For all subplots the y-axes are logarithmic. Unlike the case of non refined meshes, the minimum values of E_{Max} and E_W do not coincide. This is because the error functional E_F has a local minimum at $\gamma = 6$. Therefore, increasing γ past a critical value increases the error.

the weighted error will be smaller than the error functional. This is in contrast with the non refined mesh, where the maximum error was of order unity even for the highest resolution $\alpha = 40$.

- As expected on the basis of the previous observations, the weighted error is $O(10^5)$ smaller for a refined mesh with $\gamma = 6$ than for a non refined mesh of $\alpha = 40$.

8.5.5. Speedup from Local Refinement

After these quantitative observations, a first comparison can be made for meshes that result in the same weighted error E_W. Taking $E_{W1}\,(40, 0, 0)$ for the non refined mesh and $E_{W2}\,(20, 0, 2)$ for the locally refined mesh, we have approximately equal weighted errors:

$$E_{W1} \approx E_{W2} \approx 5 \cdot 10^{47}. \tag{8.5.12}$$

The speedup in generation time s_g (which, as previously, is given by the sum of each refinement step for the locally refined mesh) is then found as:

$$\boxed{s_g = t_{g1}/t_{g2} = 13.338/\left(2.298 + 0.074\right) \approx 4.39.} \tag{8.5.13}$$

This shows that the speedup gained by the local mesh refinement is lower than what was obtained by uniformly refining the mesh as in eq. 8.5.10. Nonetheless, in that case the comparison was carried out for configurations with the same total number of cells n_c, related to the minimum cell diameter, rather than the error itself. It must be noted that:

> For an arbitrarily small source, it is not possible to reach the weighted error obtained by local mesh refinement using uniform refinement alone, without running into memory issues.

The is due to the fact that, although a large computation time is not optimal but in theory viable, a mesh exceeding a given size will cause an error in the solver, preventing it from obtaining the solution regardless of the computation time available.

For this reason, it is harder to give a realistic comparison, on the basis of direct error measures alone, between a locally refined mesh and a mesh that is either not refined or uniformly refined across the domain. However, there is a good reason why the use of the minimum cell size is an acceptable measure of solution quality, as will be explained in the next subsection.

8.5.6. Relationship Between Cell Size and Error

The minimum cell size for the local refinement gives the scale at which the mesh can reliably express the solution and its changes. It is of course necessary that the behaviour of the source is properly reflected in the numerical solution, and it can be verified that this happens when multiple cells can fit in the volume occupied by the source itself.

Intuitively, this behaviour is to be expected: as shown in fig. 8.4.1 and fig. 8.5.6, larger sources benefit less from a higher number of cells. Empirically, the following observation holds:

> For the numerical solution to converge to the analytic solution to a sufficient degree, the cells containing the source should have radii smaller than, but of the same order of magnitude of, the source itself.

This can be directly verified from fig. 8.5.6 by noting the following:

- For $r_s = r_t/100 = 0.01 \cdot r_t$, the error does not noticeably decrease for $\beta > 3$. In fact, for $\beta = 3$ one has a minimum cell radius[9] $r_m/r_t \approx 0.01$. Hence, the minimum cell radius approximately coincides with the source radius. Recalling that by cell radius one really means its circumradius, this indicates that the element can completely fit inside the source volume. Consequently, the volume of the source will be intersected by multiple elements.

- For $r_s = r_t/200 = 0.005 \cdot r_t$ and $\beta = 3$, one has that $r_s < r_m$. Therefore, it will not be possible to fit an element inside the source volume, and it could occur that the

[9] Technically, one has the ratio between the diameters, but the ratio between the radii will clearly have the same value.

source is exclusively contained in a single element. On the other hand, for $\beta = 6$ one has $r_m = 0.002 < r_s$. In this case, the situation is again that multiple elements intersect the volume of the source. As can be seen in fig. 8.5.6, for $\beta > 6$ the error stops decreasing.

From both aforementioned cases, it can be seen that the value for which the error is lowest can be inferred from the minimum cell size.

Therefore, one can take the minimum cell size of a locally refined mesh for which the related weighted error reaches a minimum, and compare different mesh configurations which result in the same minimum cell size. This is advantageous due to the fact that good LS fits of r_m were obtained for both non refined and uniformly refined meshes.

8.5.7. Comparison for Equal Minimum Cell Radius

As the optimal value for the local refinement times is given by $\beta = 6$, it is of interest to compare meshes with the minimum cell radius corresponding to this mesh, $r_m \approx 0.002$. For a non refined mesh, one can obtain this radius for a configuration with $(277, 0, 0)$. On the other hand, for a uniformly refined mesh with configuration $(10, \beta, 0)$, the refinement times necessary would be $\beta \approx 4.95 \approx 5$. With these two configurations, one can calculate the required generation times:

- Non refined mesh: $t_{g1}(277, 0, 0) \approx 619.78s$;
- Uniformly refined mesh: $t_{g2}(10, 5, 0) \approx 0.7 + 0.0075 + 0.08 + 0.86 + 9.26 + 99.38 \approx 110.29s$.

It is now straightforward to calculate the speedups for each situation w.r.t. to the locally refined mesh with configuration $(20, 0, 6)$, with $t_g(20, 0, 6) \approx 2.3s$:

$$\boxed{s_{g1} \approx 620/2.3 \approx 270, \quad s_{g2} \approx 110.3/2.3 \approx 48.} \tag{8.5.14}$$

This result shows that, in order to obtain a solution with an arbitrarily low error, local mesh refinement can provide significant speedups over both non refined and uniformly refined meshes. Nevertheless, it must be noted that the generation time for non refined and uniformly refined meshes is constant w.r.t. the amount of sources n_s, since for all sources the error is expected to fall equally below a given threshold. On the other hand, the generation time for a locally refined mesh will grow with the number of sources n_s, implying that for large n_s uniform refinement will become the better option. It is hence important to analyse the scaling behaviour of the local mesh refinement functions, both the initial and the final version, w.r.t. n_s.

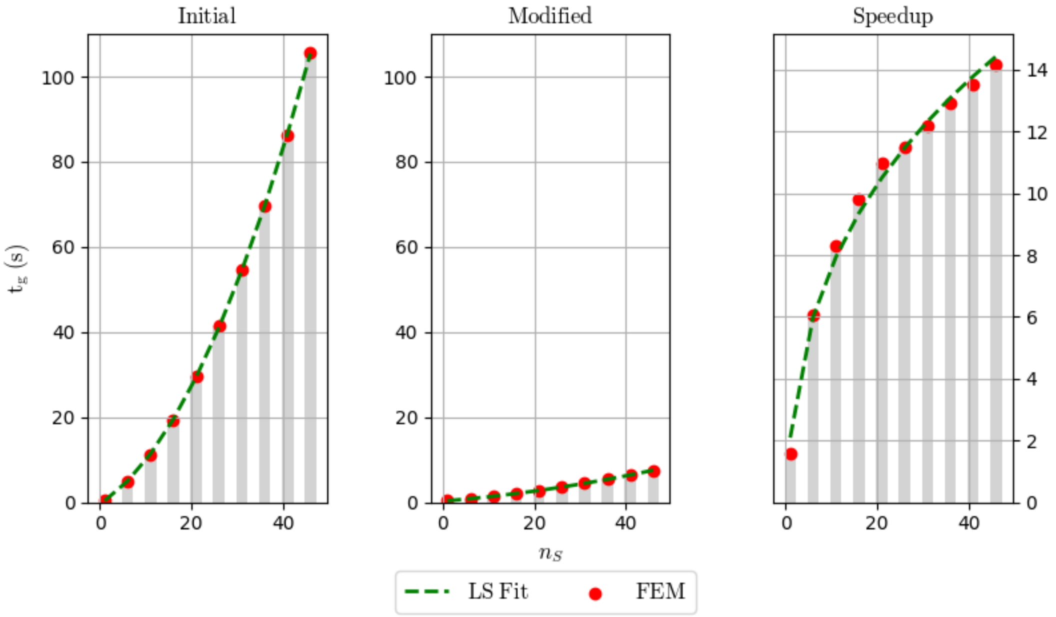

Figure 8.5.8: Time taken for the initial local mesh refinement function and for its final modified implementation against the number of sources. The refinement times are fixed to $\beta = 6$. The rightmost subplot depicts the speedup between the initial and final implementations. Red points represent the FEM data, and green dashed lines give the best LS fit.

8.5.8. Local Refinement and Source Number

The result of this analysis is given in fig. 8.5.8, where the local refinements are fixed to $\gamma = 6$, the optimal value for the source size of interest $r_s = r_t/200$. The LS fits for the two implementations, as well as the speedup between the two, are the following:

- Initial function refinement time: $t_g{}^i \approx -0.382 + 0.674 n_s + 0.035 {n_s}^2$
- Final function refinement time: $t_g{}^f \approx 0.236 + 0.086 n_s + 0.0015 {n_s}^2$
- Speedup: $s_g = -2.691 + 4.78 {n_s}^{1/3}$.

Although both functions scale with $O({n_s}^2)$, the speedup s_g provided by the improved function also increases with the source number, as would be highly desirable. With these results, we can now make a fairer comparison between the methods of local and uniform mesh refinement.[10] In order to make the comparison, it is necessary to find the number of sources n_s for which the refinement time $t_g{}^l\,(20, 0, 6)$ of the local mesh refinement algorithm matches that $t_g{}^u\,(10, 5, 0)$ of the uniform mesh refinement.

[10] It is unnecessary to consider non refined meshes. This is because it has been shown that, in order to reach the error required, the uniform refinement of an initially coarse mesh is more efficient than the direct generation of a mesh with high resolution for the same final number of cells.

This threshold indicates the value of n_s above which local refinement is no longer advantageous. This is calculated by equating the initial local refinement function to the uniform refinement:

$$\boxed{{t_g}^i = {t_g}^u \implies -0.382 + 0.674{n_s}^i + 0.035\left({n_s}^i\right)^2 = 110.3 \implies {n_s}^i \approx 48.} \tag{8.5.15}$$

Carrying out the same calculation for the final refinement function:

$$\boxed{{t_g}^f = {t_g}^u \implies 0.236 + 0.086{n_s}^f + 0.0015\left({n_s}^f\right)^2 = 110.3 \implies {n_s}^f \approx 244} \tag{8.5.16}$$

The amount of sources for which utilising local refinement is advantageous w.r.t. uniform refinement is thus considerably larger for the final local refinement function. More precisely, it is larger by a factor s_s:

$$\boxed{s_s = {n_s}^f/{n_s}^i = 244/48 \approx 5.} \tag{8.5.17}$$

That said, it should be stated that the largest galaxy clusters can contain around 1000 separate galaxies. One should then ask whether in such cases there is any advantage in utilising the proposed local mesh refinement algorithm.

To answer this question, one has to take into consideration the computation time of the full PDE solver, which is contributed to by three main sections:

- Mesh Generation;
- Mesh Refinement (uniform or local);
- PDE Solver.

When each of the above contributions for the final local refinement function are included, we arrive at the results in fig. 8.5.9. The total time shown in fig. 8.5.9 includes the constant generation time for an initial mesh with $\alpha^i = 20$, local refinement with the final refinement function for a different amount of sources, and the time taken by the PDE solver.

Perhaps surprisingly, the total time does not show the quadratic behaviour seen for the local refinement. Instead growing linearly with the number of sources n_s. The best LS fit for the total time t_t is given by:

- $t_t \approx 4.71 + 0.16n_s$.

The linearity of t_t in the number of sources can be partially understood from the weak quadratic dependence of the final local refinement function. Moreover, it implies that the solver time grows at most linearly in n_s. As the linear growth of the solver time t_s is

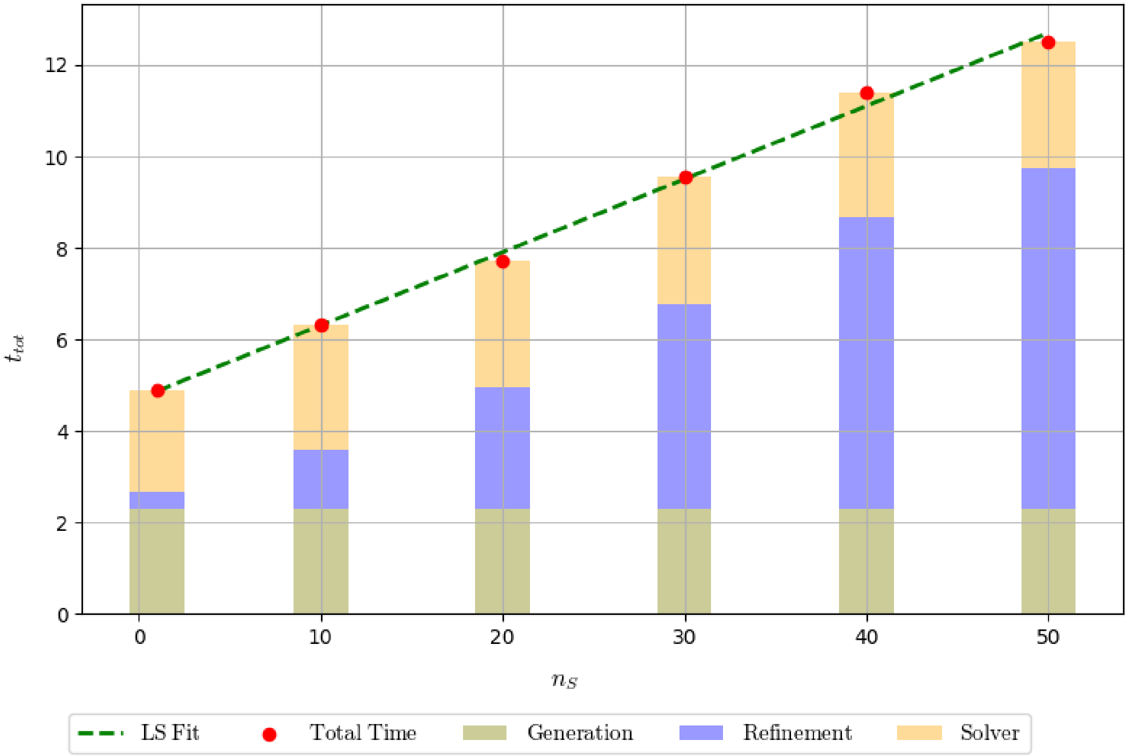

Figure 8.5.9: Total time taken for mesh generation, refinement and PDE solution when using local mesh refinement, against the number of sources. The refinement times were fixed to the optimal value $\beta = 6$. The times taken by the various sections are given by different coloured bars, whereas the red points represent the total time. The green dashed line gives the best LS fit. The values of n_s are 1 and all multiples of 10 up to 50.

negligible w.r.t. the total time t_s in the interval $10 \leq 50$, we can take its average value for $n_s \geq 10$, $t_s \approx 2.75s$.

It might be surprising that the solver time only shows a weak linear growth. However, this could be explained by noting that the solver times are determined by two competing contributions:

- The solver time t_s generally decreases when the initial guess is closer to the numerical solution, as fewer iterations are necessary to reduce the error below a given tolerance. A finer mesh close to where the solution changes most rapidly increases the accuracy of the initial guess. This reduces the steps necessary to arrive at an arbitrarily low error threshold (see, for example, [78]).

- The effect described above can be countered by the slight increase in the cell number n_c when refining the mesh locally. As previously described, the cell number n_c grows linearly with the source number n_s. Since one has $n_s \ll n_c$, it is likely that the increase in solver time caused by this increase in n_s is almost completely balanced out by the decrease in t_s caused by having a more accurate initial guess.

It is now possible to make a new, more realistic comparison, between the total computation time for an equal amount of sources when using local and uniform mesh refinement. First, it is interesting to compare the case where it was found that the two mesh generation times coincide, that is for $n_s = 244$.

To obtain the solver time for the uniform refined mesh with $\beta = 5$ the following must be noted: it has been previously shown that for approximately equal n_c, a non refined and a uniformly refined mesh will approximately have the same minimum cell size. This indicates that, to good approximation, the two meshes will have elements of comparable qualities. Thus, it is natural to infer the solver time t_s of a uniformly refined mesh from that of a non refined mesh with the same amount of cells.

It was previously shown that a uniformly refined mesh with configuration $(10, 5, 0)$ is approximately equivalent to a non refined mesh of configuration $(277, 0, 0)$. For the latter value of α, the LS fit for the solver time gives:

$$t_s\,(277, 0, 0) = 1.791 \cdot 10^{-2} \cdot 277 + 1.636 \cdot 10^{-1} 277^2 \approx 12557s \approx 3.5 \,\text{hours}. \tag{8.5.18}$$

In this case, the solver clearly accounts for almost the entirety of the computation time, and the mesh generation and uniform refinement times can be neglected. On the other hand, for $n_s = 244$ the total time taken when utilising local refinement could be obtained by either utilising the linear relation from fig. 8.5.9, or by adding each contribution individually.

As for $n_s = 244$ the quadratic contribution to the local refinement time is not negligible, one should adopt the latter approach, which yields for the total time:

$$t_t{}^f \approx 0.236 + 2.29 + 2.75 + 0.086 \cdot 244 + 0.0015 \cdot 244^2 \approx 115s. \tag{8.5.19}$$

If one had used the initial local refinement algorithm, the result would have been:

$$t_t{}^i \approx -0.382 + 0.674 * 244 + 0.035 * 244^2 + 2.29 + 2.75 \approx 2253s. \tag{8.5.20}$$

In both cases there is a total speedup for local refinement s_t, when compared to uniform refinement. For the initial local refinement function this speedup is of order unity:

$$s_t{}^i \approx 12557/2253 \approx 5.6. \tag{8.5.21}$$

On the other hand, when utilising the improved local refinement function the speedup is of two orders of magnitude:

$$\boxed{s_t{}^f \approx 12557/115 \approx 109.} \tag{8.5.22}$$

8.5.9. Linear and Nonlinear Solvers

So far, the solver time t_s has only been defined as a function of the mesh parameters α, β, γ. This has been the case because the solver used was the default serial solver in FEniCS. This approach can be fully justified by taking into account the total computation time t_{tot}. For the cases of interest, one has a number of sources of the order $n_s = 50$. From fig. 8.5.9, we can see that, for $n_s = 50$m the solver time t_s does not account for the majority of the computation time. However, we will see in the next chapter that, when utilising elements of degree 3, the total computation time is dominated by t_s. As the MOND PDE is non-linear, one needs to utilise a non-linear solver in addition to the linear solver used for the matrix-vector multiplications. By default, FEniCS uses the following combination:

- The linear solver is based on sparse LU (Lower Upper) decomposition through Gaussian elimination;
- The non-linear solver is based on the Newton-Raphson method.

As explained in [76], the use of a sparse LU linear solver can be advantageous when working with meshes with an overall number of vertices $n_v \ll 10^6$. LU solvers are also called direct solvers, as opposed to iterative solvers such as Krylov solvers. When using Krylov solvers, a preconditioner can be used to reformulate the problem in a form that is easier to solve numerically. As previously explained, for elements of degree 1, the solution is only computed at the vertices of each tetrahedron defining the element. On the other hand, for degree 3 elements the solution is computed at 20 points per cell, including 2 extra points on each edge, and an additional point on each of the four facets of each element's tetrahedron. As was shown in fig. 8.5.5, for a single source and a mesh with values $(20, 0, 6)$, we have a number of nodes equal to the vertices $n_v = O(10^4)$. On the other hand, locally refined meshes with $n_s = O(100)$ will have a number of nodes $n_v = O(10^5)$. When using a degree 3 element then, one can see that the number of nodes can reach $n_v = O(10^6)$. In this case, the use of a default LU solver cannot offer good performance. One can adopt one of two strategies:

- Utilise an LU solver that is optimised for parallelisation;
- Utilise an iterative Krylov solver, and pair it with a preconditioner optimised for parallel operation.

FEniCS offers a wide selection of linear solvers, both direct and iterative. One can also choose between two non-linear solvers, a Newton-Raphson solver or a SNES solver. The advantage of using a SNES solver is the possibility of defining a line search method to determine the local minimum of a function. In order to find the optimal solver configuration, it is necessary to test different combinations of linear and non-linear solvers. The direct and iterative solvers were analysed independently, and paired to both the Newton-Raphson and SNES non-linear solver to find the best combination. The results of this analysis will now be shown. It should be specified that only the linear solvers which could provide convergence for the MOND PDE in the case of $n_s = 50$ were examined further. Moreover, the results were analysed for degree 1 elements. This is due to the fact that for $n_s = 50$ one already has $n_v = O(10^5)$, hence making this analysis also applicable to elements of degree 3. The results for the direct solvers are shown in fig. 8.5.10. It can be seen that the performance gain stemming from the use of a better linear solver is much higher than the one stemming from the use of a different nonlinear solver. Nonetheless, for each solver the Newton non-linear solver offers a speedup of up to ≈ 1.04 compared to the SNES solver with its fastest line search algorithm. On the other hand, the difference in performance between the various linear solvers is more pronounced: the MUMPS solver offers a speedup of ≈ 1.3 w.r.t. to UMFPACK and up to ≈ 5.8 w.r.t. SUPERLU. Both values are taken for the case of most interest, namely $n_s = 50$. When compared to the default FEniCS LU solver previously examined in the chapter, MUMPS offers a speedup of up to ≈ 1.8. In addition, it should be noted that the default FEniCS LU solver is serial, whereas MUMPS is optimised for parallel computation. A similar comparison was carried out for the performance of the iterative Krylov solvers. In this case, the two solvers which

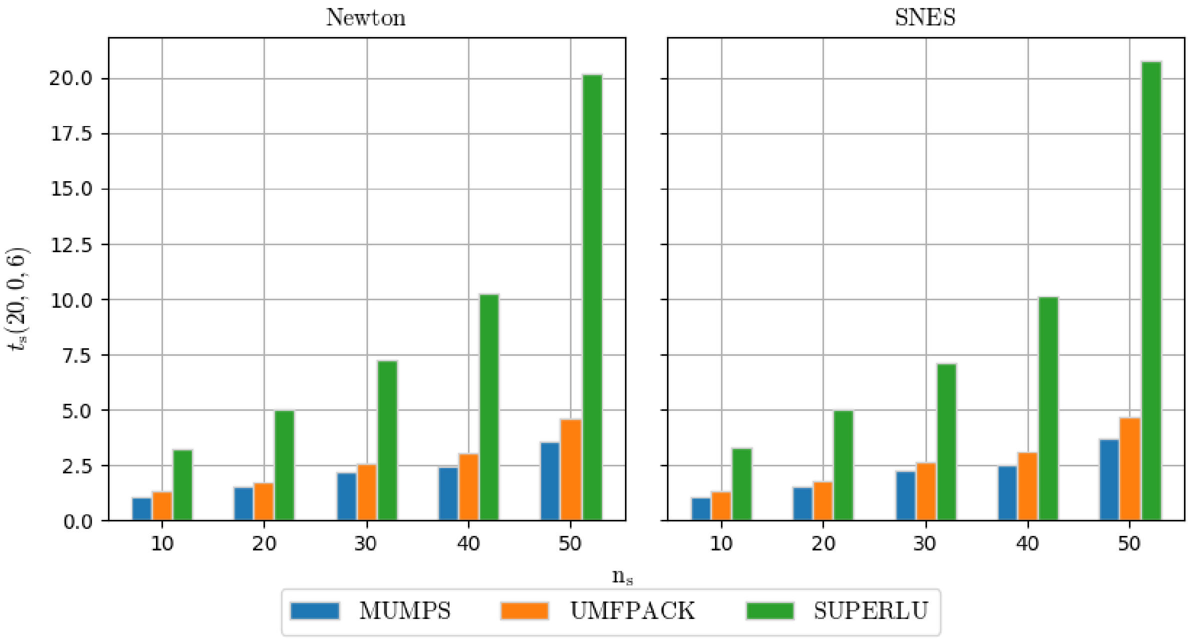

Figure 8.5.10: Solver times for different direct linear solvers, MUMPS, UMFPACK and SUPERLU. All results are for a mesh configuration of $(20, 0, 6)$ against a variable amount of sources $10 \leq n_s \leq 50$. The two graphs are for different non-linear solvers, Newton-Raphson and SNES respectively. It can be seen that the MUMPS solver offers the lowest solver time t_s for all values of n_s, when paired to both the Newton-Raphson and SNES solvers. Moreover, MUMPS is the only solver with a predominantly linear growth, whereas UMFPACK and SUPERLU show quadratic scaling. Overall, the use of the Newton-Raphson non-linear offers better performance for each solver and all values of n_s.

were able to achieve convergence for the MOND PDE, namely bicgstab and gmres, were both paired to two different preconditioners. Both preconditioners, ilu and hypre-euclid, are based on the method of incomplete LU factorisation. However, ilu is only configured for sequential operation and cannot be used alongside MPI, whereas hypre-euclid is optimised for parallel computation. The results are shown in fig. 8.5.11. As can be seen from fig. 8.5.11, the choice of a nonlinear solver does not have a strong impact on the performance of the solver, as in the case of the LU solvers in fig. 8.5.10. Nonetheless, for $n_s = 50$, the use of a Newton solver can provide a speedup of ≈ 1.05. This speedup is virtually identical to the one achieved by the use of the Newton nonlinear solver for the case of a LU linear solver. Moreover, the performance of the bicgstab and gmres krylov solvers is similar for all values of n_s, although the gmres solver diplays a speedup of up to 1.04 w.r.t. bicgstab when paired to the Newton nonlinear solver. It is, however, clear that the use of the ilu preconditioner is advantageous for all values of n_s for both the Newton and SNES nonlinear solvers. For the Newton case and $n_s = 50$, utilising ilu results in a speedup of ≈ 2 for bicgstab and ≈ 1.9 for gmres when compared to hypre-euclid. Overall,

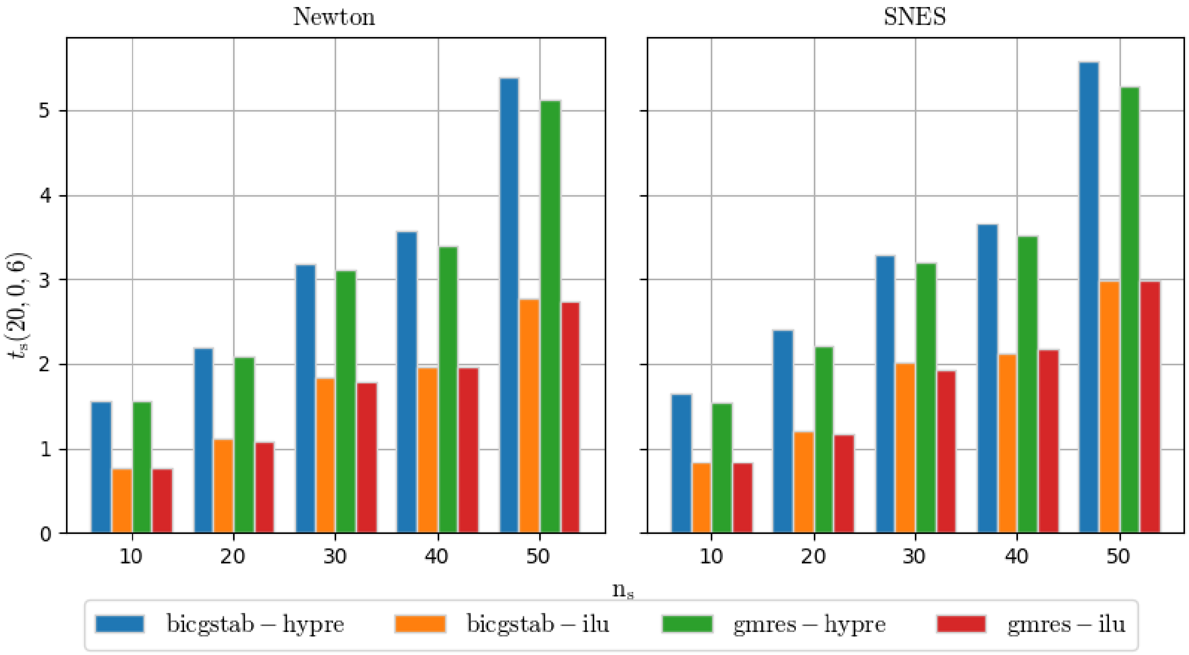

Figure 8.5.11: Solver times for different krylov iterative solvers, bicgstab and gmres. Two solvers were paired to two different preconditioners. The ilu preconditioner is optimised for serial computation and cannot be used with MPI, whereas hypre-euclid preconditioner is optimised for parallel computation. It can be seen that the difference in performance between the two krylov solvers, as well as between the use of a Newton or SNES nonlinear solver, is small in comparison to the advantage of using an ilu preconditioner. Nonetheless, one should take into account that the ilu preconditioner cannot be used for parallel computation.

the best configuration for the Krylov solver for serial execution is given by the gmres solver paired to the ilu preconditioner. When compared to the fastest LU solver MUMPS, gmres paired to ilu results in a speedup of up to ≈ 1.3, and is hence up to ≈ 2.3 faster than the default FEniCS LU solver.

However, for the case of degree 1 elements, the choice of the fastest solver provides a speedup for the total computation time which can be considered negligible even in the best case scenario: for $n_s = 50$, one has a solver time $t_s \approx 2.7s$. When considering a total time $t_t \approx 12.5s$, it can be seen that the solver only takes up $\approx 21\%$ of the total computation time. For this case, gmres provides a speedup of ≈ 1.2 w.r.t the default FEniCS LU solver, resulting in a total speedup of $\approx 4\%$. Hence, for $n_s = 50$, utilising the fastest nonlinear-linear solver combination can grant at best a speedup of order unity, $O(1)$. On the other hand, we have shown that the use of local mesh refinement could provide a speedup of $O(100)$ w.r.t. the use of uniform mesh refinement. It can therefore be concluded that, with elements of degree 1, the use of an optimal solver has only a marginal impact on the overall computation time, whereas the largest speedup can be gained by utilising an optimised local refinement algorithm, which also directly affects the solver time, regardless of the solvers in use. However, for degree 3 elements this is no longer the case, as the solver time makes up for most of the computation time, as will be described in chapter 9.

Chapter 9

Code Parallelisation

This chapter describes the optimisation of the serial code presented in Chapter 10, achieved using the MPI standard for distributed memory parallel computation. First, an overview will be given on the two main formalisms for computational speedups which can be achieved through parallelisation. These will be illustrated by Amdahl's law and Gustafson's law, which describe scaling with fixed total problem size and fixed problem size per computational unit respectively. Subsequently, an explanation will be given for why parallelism is necessary for the calculation of an apparent mass density in galaxy clusters when dealing with multiple discrete sources. Finally, the performance of the total FEM program will be analysed w.r.t. both strong and weak scaling when run on a multi-core architecture, namely, a single-threaded 8-core Intel i7-9700 with maximum frequency $f_{max} = 4.7GHz$, $8GB$ of RAM and $64KB$, $256KB$ and $12MB$ of L1, L2 and L3 cache respectively.

9.1. Scaling Behaviours

When dealing with simulations of physical systems, or modelling of real world phenomena more generally, the scale of the problem at hand must be taken into consideration. For time dependent problems, the amount of computation required increases for a smaller time step. Similarly, for time independent problems, one can generally obtain more accurate results by increasing the resolution of the mesh on which the computation is carried out. The PDE for MOND falls into the second category.

As explained in Chapter 10, in order to obtain accurate results for the case of galaxy clusters, one needs to ensure that the volume of the cells close to each of the sources is smaller than the volume of the source itself. By using local mesh refinement, we showed in the previous chapter that s considerable speedup of $O(100)$ can be obtained w.r.t. uniform mesh refinement for the sequential computation. However, the computation time for local mesh refinement grows at most quadratically with the number of sources. However, the computation time required to solve the PDE for the gravitational potential in a galaxy cluster for the desired hence increases approximately linearly with the number of sources.

In principle, then, one can expect that distributing the computation time amongst multiple computation units will result in a net speedup. However, before quantitative considerations are made on the speedup that can be obtained by parallelisation, it is necessary to introduce two models which quantify the maximum achievable speedup for a given problem, based on the fraction of the code which is inherently sequential.

9.1.1. Strong Scaling: Amdahl's Law

The first paradigm to evaluate the possible speedup for a program with a sequential fraction s and a parallel fraction p is the following: given a number of cores N, the maximum speedup S_S that one can achieve is given by the relation:

$$S_S = \frac{1}{(1-p) + \frac{p}{N}}. \tag{9.1.1}$$

This type of speedup relates to so-called **strong scaling**. In short, strong scaling describes the speedup that can be achieved when one has a fixed problem size. This relation was first put forward in 1967 by Amdahl in [79], when parallel computing was by no means as established as it is in the modern day. Eq. 9.1.1 is hence named **Amdahl's Law**.

A few observations can be made about this equation:

- A key simplification of the model is that the code is divided exactly into two components. One of these can be parallelised with no overhead and is named p, whereas the other cannot be parallelised by any means and is named s;

- As all code must fall into one of the two categories just defined, it is useful to normalise the total computation time for the non parallelised computation: $p+s=1$;

- By observing that all code which is not parallel must be sequential, one can rewrite eq. 9.1.1 as:

  $$S_A = \frac{1}{s + \frac{p}{N}}. \tag{9.1.2}$$

  From the above form, one can analyse the limit in which unlimited cores are available to be utilised in parallel:

  $$\lim_{N\to\infty} \frac{1}{s + \frac{p}{N}} = \frac{1}{s}. \tag{9.1.3}$$

- The other limit of interest is a completely parallel program. This can be expressed by the vanishing of the sequential fraction s:

  $$\lim_{s\to 0} \frac{1}{s + \frac{p}{N}} = \frac{1}{\frac{1}{N}} = N. \tag{9.1.4}$$

  Therefore, in the case of a vanishing sequential component, the only bound to the speedup is given by the available number of cores N.

In real world applications, it is not possible to achieve programs with $s = 0$. Therefore, from the last two observations it is clear that, regardless of the value of N, the maximum speedup of a program will ultimately be determined by its sequential fraction s. At the same time, if one can arbitrarily reduce the sequential component, the speedup is exclusively bounded by N. Both asymptotic behaviours can be seen in fig. 9.1.1 from [80].

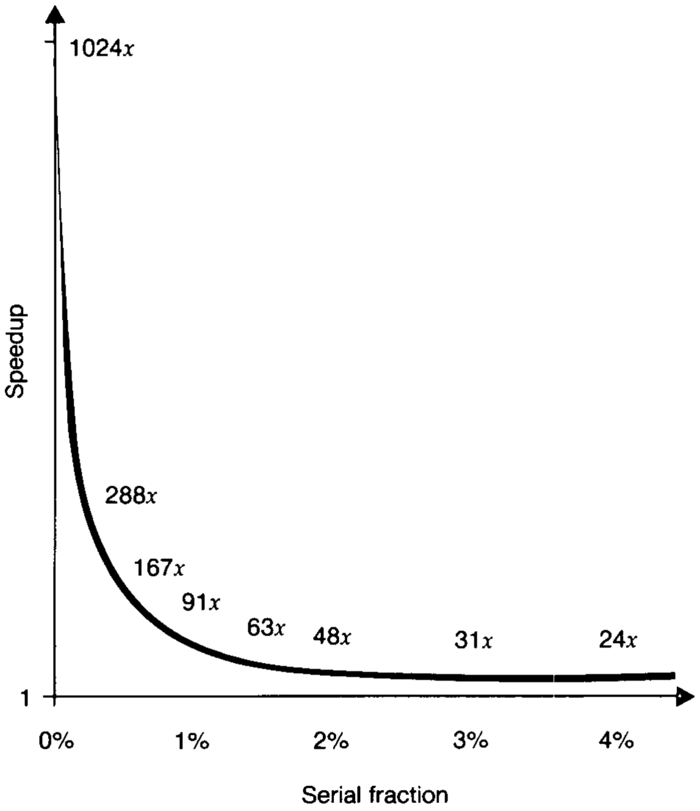

Figure 9.1.1: The relationship described by Amdahl's Law for different values of the sequential fraction. The x-axis gives the value of s as a percentage of the whole program, and the y-axis indicates the speedup S_A. Here, a fixed value of $N = 1024$ is taken. One can see that two clear asymptotes are present: for small values of s, the speedup reaches its maximum possible value of $S_A = N = 1024$, whereas for increasing serial fractions S_A approaches 0, corresponding to no parallelisation present, $s = 1$.

When Amdahl's Law first emerged, the main conclusion of its scaling behaviour was that the use of a single core with increased performance would provide a better speedup when compared to multiple processors, for comparable cost and power consumption. However, in the modern day two aspects have changed profoundly:

- Many programs, especially for scientific research, are designed with High Performance Computing in mind, resulting in a much lower value of s which allows for greater speedups when increasing N;

- Greater resolution in time steps or meshes generally yields better results. For example, it is common that errors below a certain threshold can only be achieved when utilising a higher mesh resolution. In other words, it is advantageous to be able to compute larger problems in comparable amounts of time.

Consequently, the following observation can be made: a fixed problem size with a variable amount of cores is not a realistic model in today's HPC landscape: one should instead include the speedup gained by solving larger problems while using a larger N. A paradigm to treat speedup with these characteristics is given by Gustafson's Law.

9.1.2. Weak Scaling: Gustafson's Law

The results presented in Chapter 10 made the following clear: for the case of galaxy clusters, meaningful results cannot be achieved unless the cell size close to the sources is below a given threshold. As computation time grows with the amount of cells n_s, one can identify the problem size with the amount of cells.

The main statement of Gustafson's Law is the following: rather than calculating the speedup of a fixed program when increasing N, one should analyse a situation in which the problem size increases with N. This approach essentially states that smaller problems, for which the sequential component is non-negligible, will not greatly benefit from a large N, in agreement with Amdahl's Law. However, making the assumption that the parallel component scales linearly with N and that no extra overhead is introduced, one gets a speedup S_G of the form:

$$\frac{s+p\cdot N}{s+p}. \tag{9.1.5}$$

Eq. 9.1.5 is the form that was initially given in 1988 by Gustafson himself in [80]. It can be seen that this form can be obtained directly from Amdahl's Law when one abandons the notion that the parallel fraction p is fixed.

As mentioned above, one can increase p linearly with N as:

$$p \to p\cdot N \tag{9.1.6}$$

Plugging this into eq. 9.1.1 yields a modified form of Amdahl's Law:

$$\frac{s+p}{s+\frac{p}{N}} \to \frac{s+p\cdot N}{s+\frac{p\cdot N}{N}} = \frac{s+p\cdot N}{s+p}. \tag{9.1.7}$$

The following observations can be made about eq. 9.1.5:

- When the program is almost exclusively parallel, one obtains the same limit as for Amdahl's Law:

  $$\lim_{s\to 0}\frac{s+p\cdot N}{s+p} = \frac{p\cdot N}{p} = N. \tag{9.1.8}$$

  In this case, the speedup is once again bounded by N;

- Unlike the case of Amdahl's Law, taking the limit of large N yields a speedup that becomes independent of s:

  $$\lim_{N\to\infty} \frac{s + p \cdot N}{s + p} = \frac{p \cdot N}{p} = N. \tag{9.1.9}$$

  One can thus see that, even for a program which has comparable values of p and s, increasing N will result in a linear speedup S_G. On the other hand, it should be noted that, in the general case, increasing the problem size might not lead to obtaining better results or a larger amount of useful information. Furthermore, it is possible that the computation time that can be parallelised grows faster than linearly, in which case the speedup will no longer be linearly increasing with N.

Although the assumptions that the parallel part p grows exactly linearly with N, and that no overhead is associated with increasing values of p, cannot always be satisfied for arbitrary values, Gustafson's Law still provides a useful guideline for the parallelisation of large problems. Ultimately, the speedup will depend on how precise a solution must be, or how big a domain of computation is required. Gustafson's paradigm is particularly suited to calculations for galaxy clusters, where increasing both the domain size and the accuracy of the mesh generally leads to more accurate and realistic results.

9.2. Parallelisation Results

This section describes the process of parallelising the code described in chapter 8 through the use of the MPI standard. The section is divided into two different subsections, one for elements of d_3 and the other for elements of d_1 . For convenience, degree 3 is indicated by d_3 , and degree 1 by d_1 . These two cases have to be treated independently for two reasons:

- The use of d_3 elements results in a higher number of vertices n_v. This is because each element has two nodes on each edge and a node on each facet, in addition to the nodes already present on each vertex of the tetrahedron for elements. Overall, a mesh with d_3 elements will have up to $O(10)$ more vertices than an equivalent mesh with d_1 elements. Approximately, as the solver time t_s grows quadratically in the number of vertices for large n_s, one expects the computation time to grow by $O(100)$.

- The use of d_3 elements is only required when it is necessary to calculate the second derivative of the solution to great accuracy. This is the case when one wants to analyse an apparent mass distribution in galaxy clusters;

- d_1 elements should, instead, be used when the highest derivative required is the first derivative, such as in the calculation of the gravitational acceleration. In many cases, such as the study of the potential created by a particular mass configuration, d_3 elements are therefore not necessary.

The first case introduced next will be that of d_3 elements, where the solver time t_s occupies the vast majority of the computation time t_t. This case is somewhat simpler than the one of d_1 , where different components of the program, such as mesh generation, refinement and solver all give non-negligible contributions to the total time t_{tot}.

9.2.1. d_3 Elements

When utilising elements with d_3 , that is to say elements on which the solution is approximated by a polynomial of order 3, one obtains the results for the total run time shown in fig. 9.2.1.

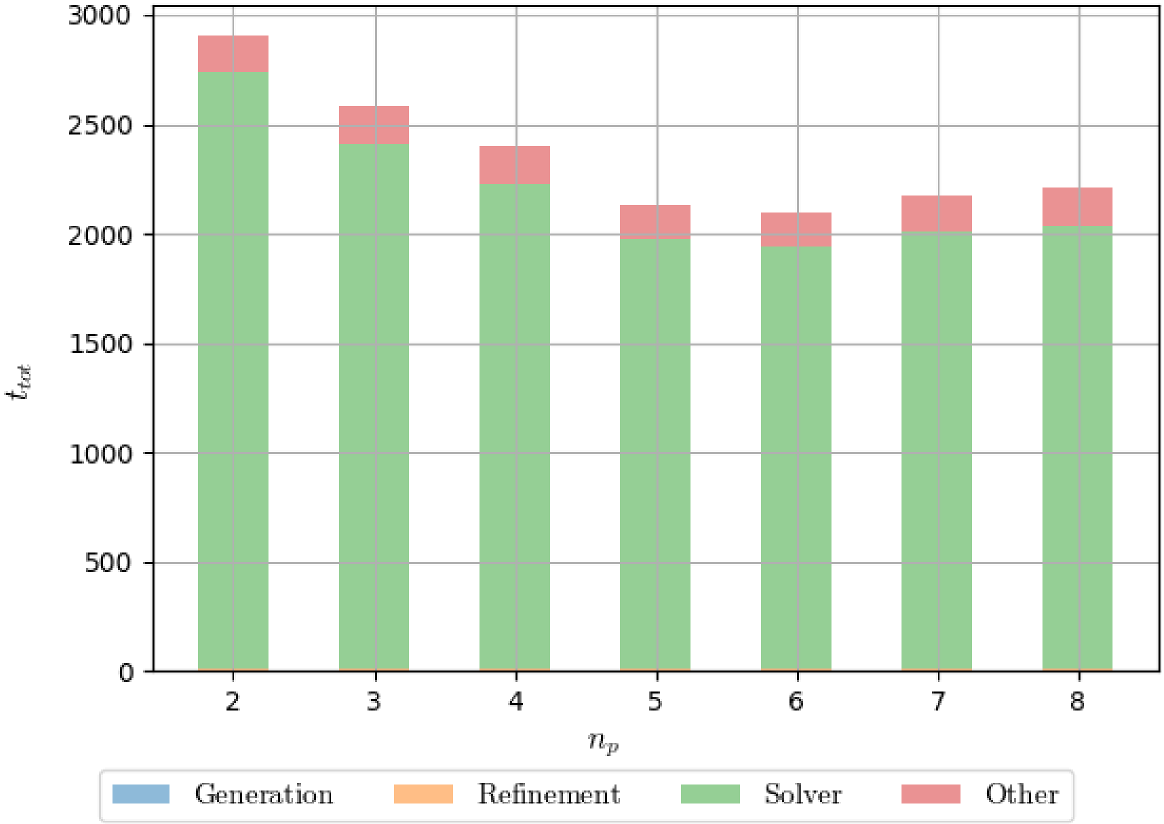

Figure 9.2.1: Total time for a fixed amount of sources $n_s = 80$, using d_3 elements. The mesh configuration is given by $(20, 0, 6)$. Each colour represents a section of the program. It can be seen that the solver time t_s dominates the computation time, and both the generation time t_g and refinement time t_r are negligible. The 'Others' label indicates processes such as plotting and explicit communication between processors for MPI. The x-axis gives the number of processors n_p used to carry out the computation.

It is clear from fig. 9.2.1 that, when using d_3 elements, the vast majority of the computation time is taken by the solver.

In fact, for all numbers of processors used n_p, the solver time t_s takes up over $\approx 92\%$ of the total computation time t_t. On the other hand, generation time t_g and refinement time t_r jointly take less than $\approx 1\%$ of the computation time for all values of n_p. Other processes, such as explicit communication between processors through MPI, plotting and post-processing of data never take up more than $\approx 7\%$ of the computation time.

In chapter 8 it was shown that, for d_1 elements, utilising local mesh refinement could considerably reduce the total computation time. This is still true for d_3 elements. However, unlike in the case of d_1 elements, the total computation time is not dominated by the mesh refinement, but by the solver. The times shown in fig. 9.2.1 pertain to the default sequential solver used in FEniCS. A modest speedup of $\approx 31\%$ can still be observed from $n_p = 2$ to $n_p = 6$, due to the fact that the domain of computation is split into more subdomains on which the PDE is independently solved by each processor. However, as the default FEniCS solver is not optimised for parallel computation, the overheads caused by the communication between processors for cells at the interface between two subdomains impede a substantial speedup.

It can be seen that the solver time t_s stops decreasing for $n_p > 6$, after which the communication overheads outweigh the parallel speedup when increasing n_p. This implies that one can calculate the effective serial component of the solver s_s by comparing the solver times t_s for $n_p = 6$ and $n_p = 1$. However, when using d_3 elements and the default sequential solver, one cannot obtain a solution using a single processor due to memory limitations.[1] As the computation time on a single processor cannot be obtained exactly, one can estimate this quantity from observing that for $n_p \leq 5$ the solver time t_s decreases approximately linearly. Although it cannot be guaranteed that this behaviour extends to the single processor behaviour, an approximation can be made, giving a single processor solver time of $t_s \approx 3000s$. Now the serial and parallel fractions can be estimated by taking value of the lowest solver time, which is $t_s \approx 1950$ for $n_p = 6$. For a serial section S and a parallel section P, this gives:

$$P + S \approx 3000, \quad P/6 + S \approx 1950 \implies \frac{5}{6}P \approx 1050 \implies P \approx 1260. \tag{9.2.10}$$

From the above, one can then find the normalised parallel and serial fractions p and s, so that $p + s = 1$:

$$\boxed{p = 1260/3000 = 0.41, \implies s = 1 - p = 0.59.} \tag{9.2.11}$$

It is useful to introduce a new variable to quantify speedup, σ. However, according to Amdahl's law, the large serial fraction s limits the achievable speedup to $\sigma_A = 1/0.59 \approx$

[1] The total memory that can be used by a single process with the default sequential solver in FEniCS is limited to 4GB. This size is exceeded for mesh configurations of $(20, 0, 6)$ with even a single source. This limitation is not imposed when using MPI and multiple processors.

1.7. One could then define the Amdahl speedup efficiency η as the ratio of the largest speedup η_A obtained with a configuration with a bound on the number of compute units $n_p \leq 8$ as:

$$\boxed{\eta_A = \frac{\text{Max}\,(\sigma)}{\sigma_A}} \tag{9.2.12}$$

It can be seen that, although the default FEniCS solver is sequential, the subdivision of the domain into smaller subdomains can result in a speedup of $\sigma_t \approx 3000/1950 \approx 1.5$. This corresponds to a speedup efficiency of:

$$\eta_A \approx 1.5/1.7 \approx 0.88 \tag{9.2.13}$$

Although the efficiency is close to 1, the speedup is likely to be limited by the communication needed between the processors for elements at the boundary between two subdomains, which is not optimised for the MPI standard in the default solver. This results in the large serial fraction $s = 0.59$ which ultimately limits the speedup to $\sigma_A = 1.7$, a clearly suboptimal value when utilising configurations with up to $n_p = 8$ compute units. Another useful speedup efficiency that can be introduced is the parallel efficiency η_p, providing the ratio of the largest speedup achieved w.r.t. a completely parallel program with $p = 1$, with the same single-core computation time. This is equivalent to the ratio of the largest speedup to the number of available cores:

$$\boxed{\eta_p = \frac{\text{Max}\,(\sigma)}{\text{Max}\,(n_p)}.} \tag{9.2.14}$$

When using the default FEniCS solver, one has:

$$\eta_p = \frac{1.5}{8} \approx 0.19. \tag{9.2.15}$$

From the above it is clear that, although the Amdahl efficiency η_A is close to unity, the large serial fraction of the default FEnICS solver results in a low value for the parallel efficiency η_p. Therefore, it is necessary to investigate how the performance of other solvers available in FEniCS can reduce the solver time t_s. An analysis on the performance of the different solvers available in FEniCS was carried out in chapter 8. The main findings were the following:

- The use of the Newton nonlinear solver can provide a speedup of $\approx 5\%$ w.r.t. the SNES nonlinear solver;

- Amongst the direct LU solvers, the `MUMPS` solver provides the best performance. Nonetheless, the `UMFPACK` solver provides comparable performance in the range $10 \leq n_s \leq 50$. `MUMPS` is optimised for parallel computation, whereas `UMFPACK` is optimised for serial computation.

- For iterative Krylov solvers, the choice between the solvers `bicgstab` and `gmres` has a negligible impact on performance. On the other hand, the choice of the `ilu` preconditioner results in speedups of $\sigma \approx 2$ for both solvers when compared to the `hypre-euclid` preconditioner. It must, however, be taken into account that `ilu` cannot be used alongside MPI, so for parallel computation one needs to use `hypre-euclid`. For the serial case, the LU solver `MUMPS` gives a speedup of $\sigma \approx 1.5$ w.r.t. `gmres` with `hypre-euclid`. As both `MUMPS` and `hypre-euclid` are optimised for MPI, it is interesting to test if this relation also holds when utilising multiple processors.

It has been shown in chapter 8 that the solver times for `MUMPS` and `gmres` increase approximately linearly with the number of processors used n_p, although a quadratic component becomes non-negligible for large values of n_s. For the range $a0 \leq n_s \leq 80$ this implies the following:

For the analysis of weak scaling and the speedup because of Gustafson's law, it is necessary to choose a parameter for which the computation time grows approximately linearly. It is suitable to choose the number of sources n_s as the parameter defining the linearly increasing problem size.

To verify that the growth of the solver time t_s w.r.t. the number of sources n_s can indeed be approximated by a linear relation in the range of interest, one must test the weak scaling behaviour with the source number n_s increasing with the number of cores used n_p. However, one must first infer the parallel and serial fractions of the computation, p and s respectively, by analysing the behaviour under strong scaling. This is achieved by keeping the problem size constant and increasing the number of cores utilised.

By defining the problem size as the number of sources n_s, one can use the following method for weak scaling: for each added core, 10 sources are added to the problem.

As the parallelisation is targeted at a multi-core system with 8 cores, the number of sources for weak scaling varies in the range $10 \leq n_s \leq 80$. For the evaluation of strong scaling it is hence suitable to choose n_s to be close to the average of this range. A suitable choice is given by $n_s = 40$. The evaluation of strong and weak scaling must then follow these steps:

1. Compute the strong scaling for the different solvers for a set number of sources $n_s = 40$;

2. From the results of strong scaling, choose the solver with the largest parallel component p;

3. For the chosen solver, determine the serial and parallel fractions s and p;

4. Utilise the serial and parallel fractions to compute strong scaling by increasing the problem size by 10 sources for each added core.

9.2.2. Strong Scaling for Different Linear Solvers

The first step is to fix the problem size to $n_s = 40$ and evaluate the performance of three solvers:

- The fastest serial LU solver `UMFPACK`;
- The fastest parallel LU solver `MUMPS`;
- The fastest combination of Krylov solver and parallel preconditioner, `gmres` with `hypre-euclid`.

The results for the strong scaling are shown in fig. 9.2.2. From fig. 9.2.2, one can clearly see that `gmres` has the best parallel performance for all numbers of cores $2 \leq n_p \leq 8$. On

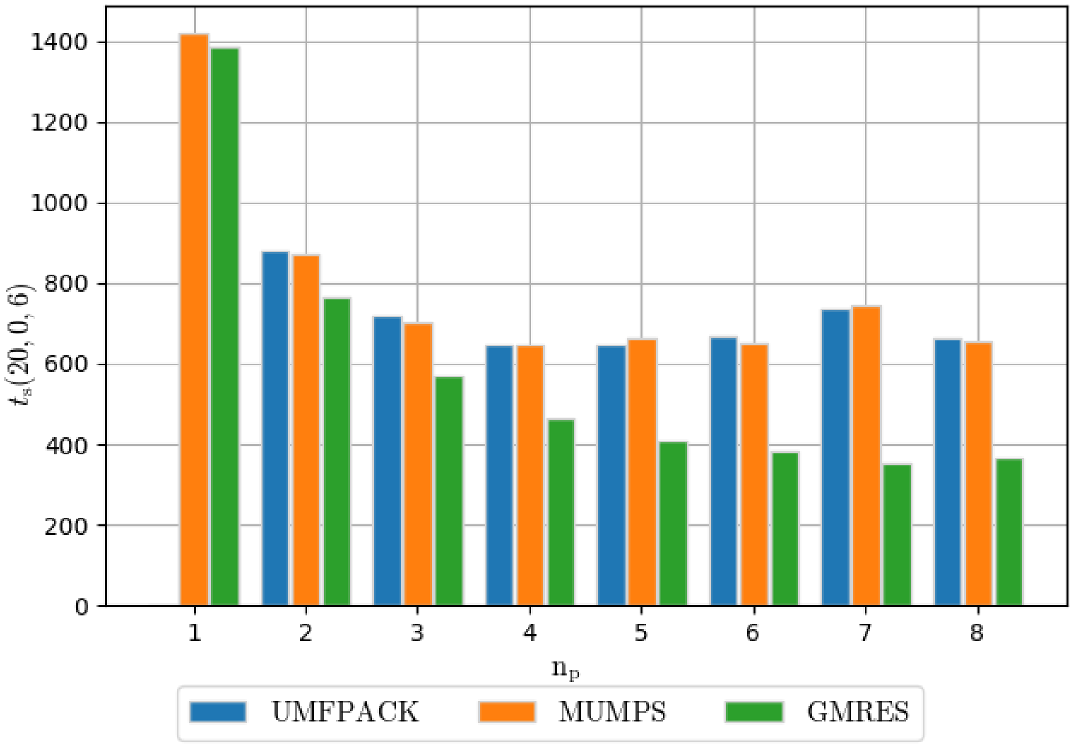

Figure 9.2.2: Total time for a fixed amount of sources $n_s = 80$, using d_3 elements, for different choices of the linear solver. The three solvers shown are the fastest sequential LU solver `UMFPACK`, the fastest parallel LU solver `MUMPS`, and the fastest iterative solver with parallel precodintioner, `gmres` with `hypre-euclid`. It can be seen that, although `MUMPS` is optimised for parallel computation, its performance is similar to that of `UMFPACK` for all values of n_p. On the other hand, `gmres` with the `hypre-euclid` preconditioner has a comparable solver time on one processor, but provides a speedup of up to $\sigma \approx 2.1$, achieved for $n_p = 7$ when compared to `MUMPS` and `UMFPACK`.

the other hand, the sequential times for MUMPS and gmres are comparable. The value for $n_p = 1$ for UMFPACK is not present in fig. 9.2.2 because UMFPACK cannot produce the solution for the given mesh configuration $(20, 0, 6)$ due to memory constraints. One can see that gmres achieves the lowest value of solver time t_s for $n_p = 7$. For this value, it provides a speedup of ≈ 2.1 w.r.t. to MUMPS and of $\sigma \approx 2$ w.r.t. UMFPACK. It should also be noted that UMFPACK, which is optimised for sequential operation, outperforms MUMPS, optimised for parallel computation, for $n_p = 5, 7$. Nonetheless, in the general case MUMPS is marginally faster. It should be mentioned that the performance of gmres should be attributed to the use of the parallel hypre-euclid preconditioner.

In contrast to the parallel case, in the sequential computation shown in chapter 8, MUMPS provided a speedup of $\sigma \approx 1.5$ w.r.t. gmres with hypre-euclid. This is an indication that the optimisation on the hypre-euclid preconditioner grants a better performance than the parallel optimisation of the direct LU solver MUMPS. It is hence clear that:

> For the parallel case, the best performance for the linear solver is achieved by the combination of gmres and the hypre-euclid preconditioner.

From here on, the gmres and hypre-euclid combination will be referred to as gmres for simplicity. Unlike in the case of the default FEniCS LU solver shown in fig. 9.2.1, it was possible to obtain results for the solver time of gmres on a single processor, allowing the precise calculation of its serial and sequential components, P and S. As the solver time does not decrease for $n_p > 7$, the value for $n_p = 7$ was used as follows:

$$P + S \approx 1385s, \quad P/7 + S \approx 350s \implies \frac{6}{7}P \approx 1035s \implies P \approx 1207s. \tag{9.2.16}$$

Normalising to obtain the serial and parallel fractions p and s, we then have:

$$\boxed{p = \frac{P}{P+S} \approx 1207/1385 \approx 0.87 \implies s = 1 - p = 0.13.} \tag{9.2.17}$$

When comparing the value of p to that of the default FEniCS LU solver, one finds that for gmres it is larger by a factor of ≈ 2.12. More importantly, the serial component that determines the maximum strong speedup is smaller by a factor of ≈ 4.5. Following Amdahl's law, this results in a maximum speedup for strong scaling when using gmres of:

$$\sigma_A = \frac{1}{s} \approx 7.7. \tag{9.2.18}$$

It can be seen that for an 8-core configuration this value almost coincides with the maximum achievable parallel speedup $\sigma_p = 8$. When comparing this value to the maximum achievable speedup for the default FEniCS LU solver, $1/0.59 \approx 1.7$, we can see that the speedup for gmres can be larger by a factor of ≈ 4.5. The shortest solver time t_s is achieved for $n_p = 7$, and provides a speedup w.r.t. the serial performance of:

$$\sigma_s \approx 1385/352 \approx 3.92. \tag{9.2.19}$$

For the speedup efficiency, the Amdahl efficiency η_A and parallel efficiency η_p are approximately equal:

$$\eta_A \approx 3.92/7.7 \approx 0.51, \quad \eta_p \approx 3.92/8 \approx 0.49. \tag{9.2.20}$$

It can then be seen that, although the Amdahl efficiency η_A is higher for the default solver, the following applies:

> `gmres` achieves a parallel efficiency η_p greater than the default FEniCS solver by a factor of ≈ 2.6.

In order to determine whether the extrapolation of the values for s and p for `gmres` is reliable, one can directly plot the strong speedup as shown in fig. 9.2.3. As can be seen from fig. 9.2.3, there is excellent agreement between the best LS fit for the speedup and

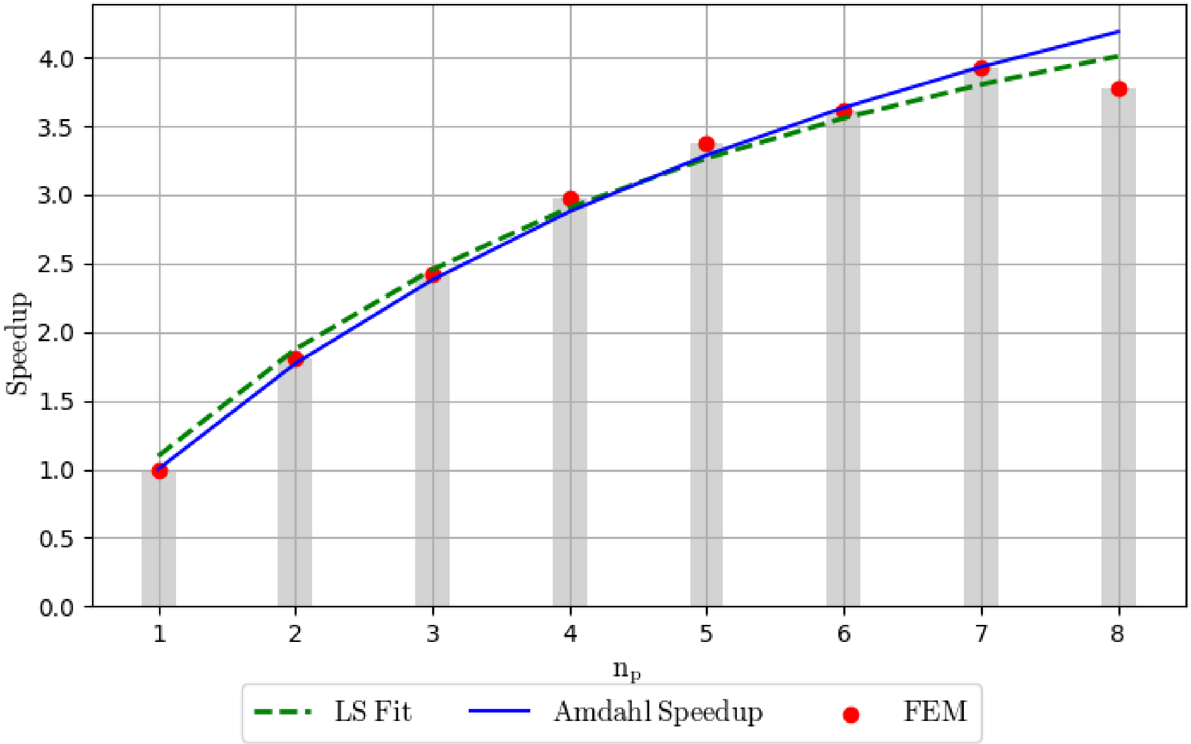

Figure 9.2.3: Speedup for the strong scaling of the `gmres` solver with the `hypre-euclid` preconditioner. The red points represent the FEM data, whereas the green dotted line gives the best LS fit. The solid blue line represents the theoretical value for Amdahl's law for the inferred value for the serial fraction $s = 1.3$. The x-axis gives the number of processors used n_p. It can be seen that there is excellent agreement between the LS fit and the value for Amdahl's law based on the inferred value of s. The values for the Amdahl and parallel speedup efficiencies are approximately equivalent at $\eta_A \approx \eta_p \approx 0.5$.

the theoretical value provided by Amdahl's law. The LS fit has values:

$$s_{LS} \approx 0.15, \quad p_{LS} = 0.76. \tag{9.2.21}$$

Compared to the inferred value $s = 0.13$, there is only a difference of $\approx 15\%$ in the LS fit. It should, however, be noted, that from the LS fit one has $s_{LS} + p_{LS} \approx 0.91 \neq 1$. This can be interpreted as the existence of a portion of the code, approximately 0.09 of the total, which does not decrease linearly with an increase in the number of cores n_p but is not completely sequential. This could be due to communication between processes in MPI, or to the fact that for $n_p \gg 50$ the second order term of the solver time w.r.t. the number of sources n_s becomes non-negligible.

After confirming the validity of the values for s and p through the use of Amdahl's law, it is possible to compute the weak scaling behaviour. As previously shown in fig. 8.5.11 of chapter 8, `gmres` scales approximately linearly with n_s. In the range $10 \leq n_s \leq 50$ the quadratic contribution is negligible. The linear behaviour should hence still give a good approximation up to $n_s = 80$. After identifying the problem size with the amount of sources, the weak scaling is obtained by increasing the number of cores n_p alongside n_s. By varying n_s by 10 for each data point, one obtains the results shown in fig. 9.2.4. As can be seen from fig. 9.2.4, the solver time t_s with a set number of sources per processor $n_s = 10 \cdot n_p$ has a quadratic behaviour with a minimum at $n_p = 4$ corresponding to $n_s = 40$. In the ideal case, for a parallel component that grows exactly linearly, the solver time for a fixed problem size per core would be constant. However, it was previously stated that the scaling of the solver time t_s w.r.t. n_s also has a quadratic component. Although the quadratic behaviour is evident, the deviation from a constant value is always modest, in the range $10 \leq n_s \leq 80$: the solver time t_s is at most 22% larger, for $n_p = 4$, and 19% smaller, for $n_p = 8$. This implies that for $n_s \leq 4$ the parallel effective fraction p is larger than the fraction computed with Amdahl's law, whereas for $n_p \geq 5$ this fraction is smaller.

The behaviour for $n_p \geq 5$ can be explained by taking into account the communication overhead between processors. If one makes the assumption that the sphere is perfectly partitioned into n subdomains, each subdomain is a regular spherical wedge. If the partitions are equal, the volume of each wedge, and hence the number of vertices n_v and cells n_p inside each wedge, is approximately equal. Without communication between processors, one would hence expect a constant time for the weak scaling. Similarly, if the communication time between cores was fixed, there should be no noticeable increase in the solver time t_s w.r.t. n_p. Indeed, if each core communicated with only two other cores, it would follow that the communication time would not appreciably increase w.r.t. n_s.

However, it should be noted that, for the case of $n_p = 3$, each processor has to communicate with at most two neighbours. This is the same value as the ideal case in which each wedge is identical and each core only communicates with two other cores. Therefore, for $n_p \leq 3$ one does not expect the communication to add a substantial overhead. On the other hand, as the lists for both cells and vertices need to be updated after each refinement, the mesh also has to be re-partitioned every time the refinement function is utilised. In all tests, the sources were uniformly scattered across the domain. As the mesh partitioning cannot be set manually and is fully handled by the `SCOTCH` library,

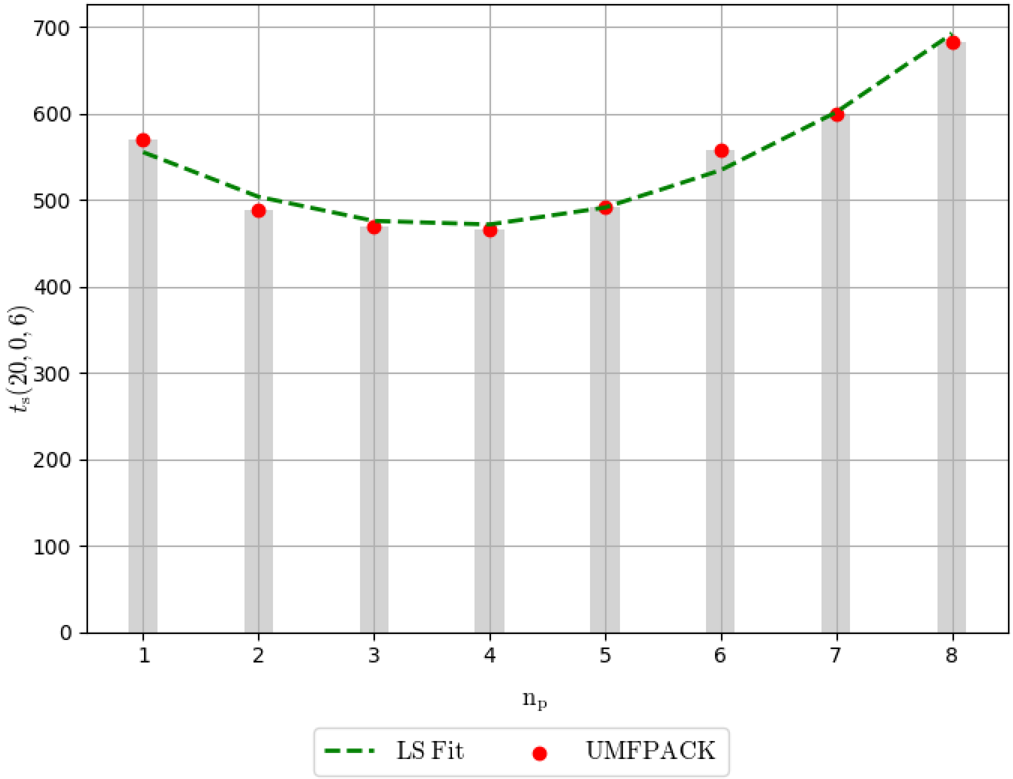

Figure 9.2.4: Solver time for the combination of the `gmres` Krylov solver and the `hypre-euclid` preconditioner for a set amount of sources per processors: $n_s = 10n_p$. It can be seen that the solver time behaves as a parbola with a minimum at $n_p = 4$. The red points give the FEM results, whereas the green dashed line gives the best LS fit. For all values of $n_p \geq 2$, the solver time t_s stays approximately within ±20% of the serial time. This indicates that the approximation of linear scaling with problem size is valid in the range $10 \leq n_s \leq 80$.

the more sources are added, the higher the chance is that the domain will not be split into equal regular spherical wedges. This, in turn, implies that the likelihood that each processor will have to communicate with more than two neighbours grows with the source number n_s. One can then expect a growth in the solver time t_s after the critical value $n_p = 3$, when each processor has to communicate with more than two neighbours. The communications between cores are blocking, meaning that computation cannot be carried out by the core sending a message before confirmation is received. Moreover, the solver time is bounded below by the time taken by the slowest core. It is hence sufficient that one core has a number of neighbours larger than 3 for the solver time to increase. By making the reasonable assumption that the probability of a core having more than one neighbour scales linearly with n_p, we can explain the linear increase in t_s for the range $n_p \geq 4$.

Instead, the decrease in t_s for the range $1 \leq n_p \leq 3$ can be explained by noting from fig.

8.5.11 that in the serial case, the linear coefficient of the scaling of t_s w.r.t. n_s is smaller than one. This implies that, provided that communication overheads are small compared to the computation time, increasing n_p for a fixed n_s per compute unit will in fact decrease the solver time t_s. Given the aforementioned observations, an exact linear speedup is not expected. Even so, because the solver time t_s only changes by approximately ±20% in the range $1 \leq n_p \leq 8$, the weak speedup should be of the same order of magnitude as the one described by Gustafson's law. The weak scaling behaviour is shown in fig. 9.2.5. As can be seen from fig. 9.2.5, the LS fit for the weak speedup coincides almost exactly with the formula for the Gustafson speedup. It should be noted that in order to obtain the LS fit, the serial fraction was bounded above by the value inferred by the use of Amdahl's law, $s \leq 0.13$. On the other hand, the parallel fraction was left as a free parameter, and the

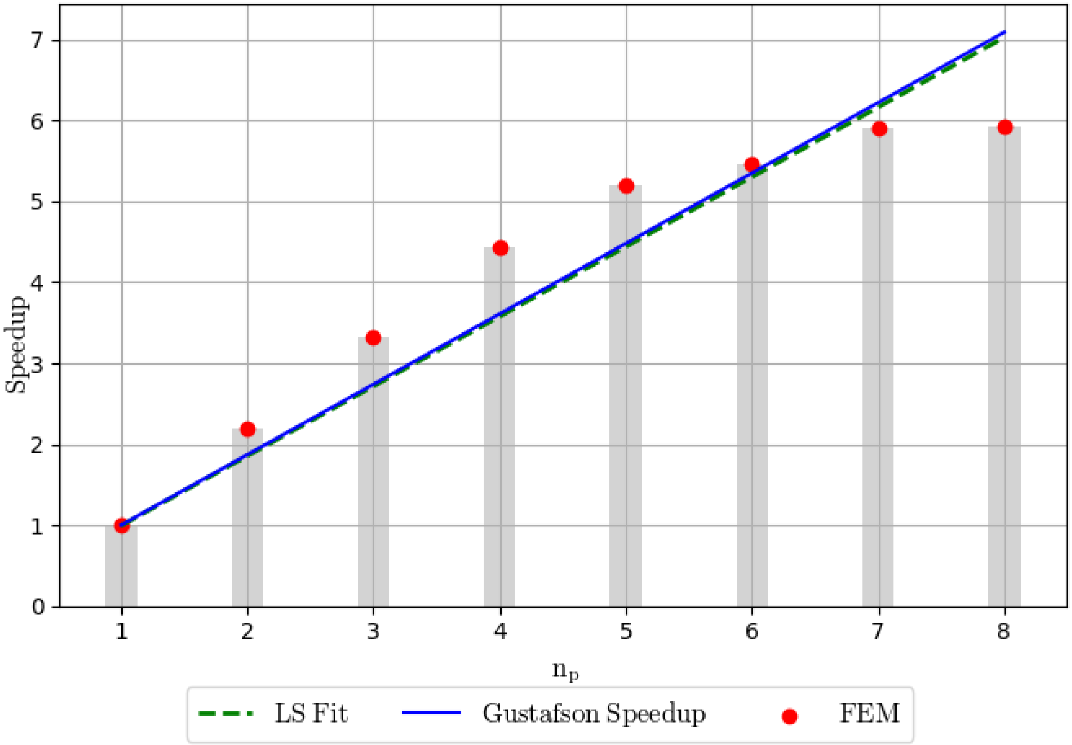

Figure 9.2.5: Weak speedup for the combination of the `gmres` Krylov solver and the `hypre-euclid` preconditioner for a set amount of sources per processor: $n_s = 10n_p$. The red points give the FEM results, whereas the green dashed line gives the best LS fit. The solid blue line gives the theoretical Gustafson speedup for the inferred value $s = 0.13$. For the LS fit, the serial value was bounded to $s \leq 0.13$, whereas the parallel value was left as a free parameter. It can be seen that there is excellent agreement between the theoretical Gustafson speedup value and the linear LS fit.

LS fit for the weak speedup was found to be:

$$\sigma_W = 0.13 + 0.86 n_p. \tag{9.2.22}$$

Thus, the value found for p for the LS fit was approximately equal to the theoretical prediction, $p_{LS} = 0.86 \approx 0.87$. Nevertheless, it is clear from fig.9.2.5 that the scaling is not exactly linear. This was expected from the result presented in fig. 9.2.4, where, for $n_p \leq 6$, it was shown that that solver time t_s was lower than for the serial case, indicating a speedup larger than the Gustafson speedup for the given value of s. On the other hand, for $n_p \geq 7$, the solver time t_s increased, indicating a speedup lower than the theoretical maximum. Lastly, it is important to observe that the speedup increases at a lower rate for $n_p \geq 5$. This indicates that for this range the communication overheads have to be added to the serial fraction s in order to obtain a reliable speedup estimate. Nonetheless, it should be noted that:

> On a system with 8 cores, the weak speedup is close to the theoretical maximum, which allows us to compute a problem size larger by a factor of 8, hence almost an order of magnitude, with only a 22% increase in computation time.

9.2.3. d_1 Elements

The same analysis carried out for d_3 elements was also carried out for d_1 elements, over which the solution is approximated by linear function, a polynomial of degree 1. As was done previously, the first step is to analyse how the computation time is distributed among the various sections of the program. Again, this was carried out for the highest problem size of interest, $n_s = 80$. The results are shown in fig. 9.2.6.

It is immediately clear that the situation for d_1 elements is very different from the case of d_3 examined in the previous section. For all cases $n_p \geq 2$, the solver time t_s accounts for at most $\approx 20\%$ of the total computation time t_t. In chapter 8 it was shown that the choice of a faster iterative solver can at most increase the serial solver time by a factor of ≈ 2.3. This implies that the maximum speedup that can be achieved by utilising a faster solver is of $\sigma \approx 0.2/2.3 \approx 8.5\%$. One can then see that, compared to the case of d_3 , the choice of a faster linear solver is not an efficient approach to significantly increase performance. Moreover, the generation time t_g is non-negligible for all cases, and takes up to 25% of the total time. This approximately constant time includes the mesh generation and mesh partitioning amongst all processors. Both processes must be carried out sequentially as FEniCS does not support any parallel mesh generation schemes, so the mesh generation represents an immutable contribution to the total serial fraction s_t. It should be noted that the mesh partitioning process results in a slight increase of the overall mesh generation time for a larger n_p. Nonetheless, the generation time t_g only varies from $\approx 2.64s$ for $n_p = 2$ to $2.71s$ for $n_p = 8$, hence the variation of $0.07s$ is negligible for all values of n_p.

Other operations such as explicit communication between processors and plotting never contribute more than 7% to the total time and can hence be neglected. Instead, most of

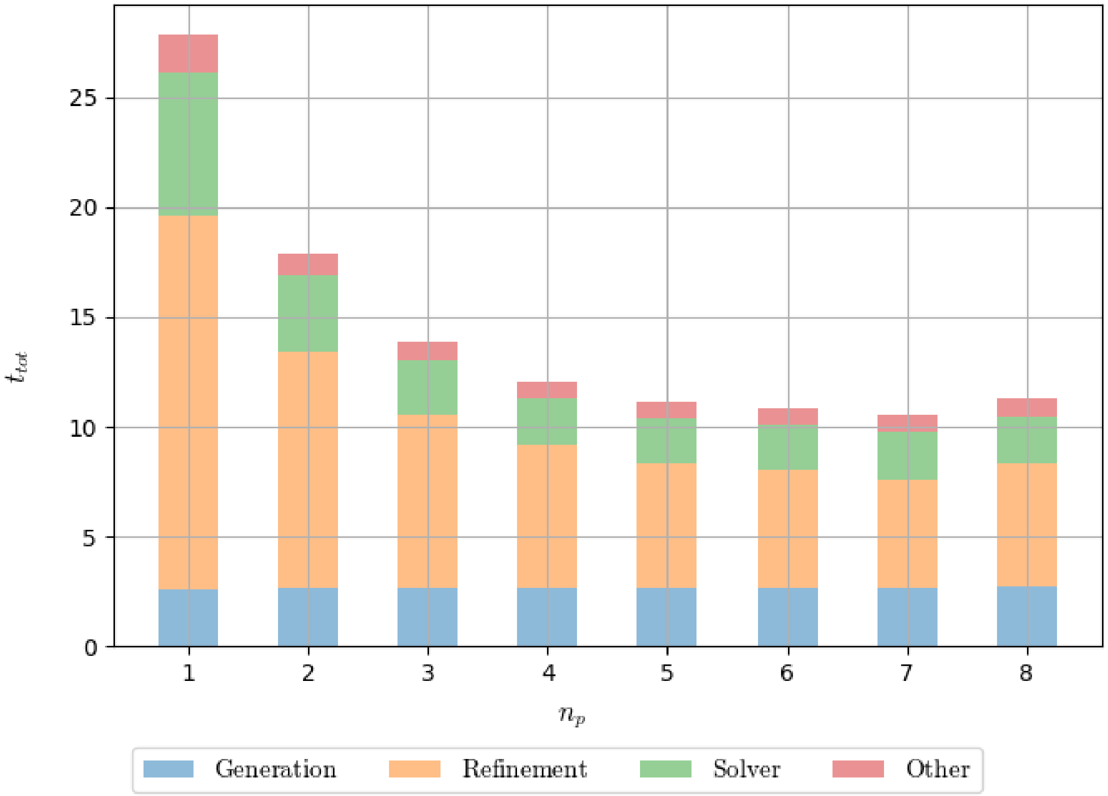

Figure 9.2.6: Total computation time for a fixed problem size $n_s = 80$, for a variable amount of cores n_p. The solver used is the default FEniCS LU solver. Bars of different colours represent the computation time of separate sections. Unlike in the case of d_3 elements, the solver does not take up the majority of the computation time for any value of n_p. The Other section includes the sum of computations such as explicit communications between processors, plotting and other data manipulation.

the computation time is spent on local mesh refinement. It should again be noted that, although the local refinement time t_l is the largest contribution to t_t, it was shown in chapter 8 that, for serial computation, utilising local mesh refinement provides a speedup to the total computation of ≈ 109. This is due to the reduction in the number of cells required to obtain an arbitrarily low error, and the relative decrease in solver time t_s. However, it should be mentioned that for the improved local refinement function introduced in chapter 8 to be used during parallel computation, the following observations had to be made:

- After mesh partitioning, each processor still has access to a global list of all cells and vertices in the mesh;

- It is not possible for a process to refine a cell that does not belong to its subdomain without communicating with another processor;

- To avoid needless communication and allow for parallelisation, every processor should only refine the cells in its own subdomain.

To comply with all of the above, we need to refer to the implementation of the improved local refinement function in chapter 8. To avoid the loop over each cell in the mesh, it was observed that we could use the FEniCS `intersect` function, as follows:

```
        intersect_list = [intersect(mesh, source).intersected_cells() for
        source in location]
```

The issue of this approach is that, when using more than one processor, the mesh object will refer to the submesh on the subdomain assigned to each processor, whereas the list of cells available to each processor will contain all cells in the complete mesh. In the serial computation, once the `intersect_list` has been obtained, we can loop over its elements to set the `meshfunction` to `True` for each cell containing a source. This is expressed as:

```
        for cell_index in intersect_list:
            contain_function[cell_index[0]] = True

            #Refining the mesh only for cells that contain a source
            mesh = refine(mesh, contain_function)
```

The problem is that this approach does not cover the case in which no cell contains the source. This is not a problem if a single processor contains the full mesh, but when using multiple processors the sources will be distributed among the different subdomains. To fix this problem, we could use various approaches, for example:

- Sharing a single `meshfunction` among all processors;
- Generating a list of cells on each processor which only contains the mesh entities contained in the submesh.

However, the easiest solution is to simply check that the list is not empty before setting the value of the `meshfunction`. This avoids communication between processors and the rearrangement of the cell list that is attached to each subdomain, and it only requires a simple conditional statement. The code then becomes:

```
            for cell_index in intersect_list:

                #If the list is empty, the source belongs to a different
                #process' subdomain. Do not update the meshfunction
                if not len(cell_index) == 0:

                    contain_function[cell_index[0]] = True

            #Refining the mesh only for cells that contain a source
            mesh = refine(mesh, contain_function)
```

This small modification allows each processor to independently refine the mesh of the subdomain it is assigned to, avoiding communication overheads. It is now possible to estimate the serial and parallel components of the total time t_t. In order to do so, it can be seen that both the generation time t_g and Other time t_o can be taken as completely serial, as mesh generation and plotting are exclusively carried out by one core. On the other hand, the values for the solver and refinement times, t_s and t_l, can be obtained by using Amdahl's law as we did for d_3 elements. Starting with the solver, we consider the cases of $n_p = 1$ and $n_p = 7$, as the total time is lowest for the latter value. We then get, for the serial and parallel components of the solver S_s and T_s:

$$S_s + P_s \approx 6.6s, \quad S_s + P_s/7 = 2.1s \implies \frac{6}{7}P_s = 4.5s \implies P_s \approx 5.25s. \tag{9.2.23}$$

After normalisation, we then have serial and parallel fractions for the solver, s_s and s_p:

$$\boxed{p_s = P_s/(P_S + S_s) \approx 0.79 \implies s_s = 1 - p_s \approx 0.21.} \tag{9.2.24}$$

Following the same procedure for the local refinement function, the serial and parallel portions P_l and S_l give:

$$S_l + P_l \approx 17s, \quad S_l + P_l/7 \approx 5s \implies \frac{6}{7}P_l \approx 12s \implies P_l \approx 14s. \tag{9.2.25}$$

The resulting serial and parallel fractions s_l and p_l read:

$$\boxed{p_l = P_l/(S_l + P_l) \approx 0.82 \implies s_l = 1 - p_l = 0.18.} \tag{9.2.26}$$

It is now possible to compute the serial and parallel fractions for the total computation time as the sum of the fractions from each component. With t_i being the fraction of the total time taken by each component for $n_p = 1$, summing over each i^{th} component we obtain:[2]

$$s_t = \frac{1}{t_t}\sum_{i=1}^{3} t_i s_i = \frac{s_g \cdot t_g + s_l \cdot t_l + s_s \cdot t_s}{t_t}. \tag{9.2.27}$$

[2] In the calculation of the overall serial fraction s_t, the Other times are discarded, as the value is highest for $n_p = 1$ and hence does not reflect the explicit MPI communication between processors, which is the only quantity of interest within Other. The Other time is mostly taken by plotting, which can be neglected when we are only interested in the time taken to obtain the solution itself.

Plugging in the values found above, we have:

$$\boxed{s_t \approx \frac{1 \cdot 2.6 + 0.18 \cdot 17 + 0.21 \cdot 6.6}{26.2} \approx 0.27 \implies p_t = 1 - s_t \approx 0.73.} \tag{9.2.28}$$

It is now possible to calculate the speedup over the range $n_p \leq 8$ and compare it to Amdahl's law. The result is shown in fig. 9.2.7. As can be seen from fig. 9.2.7, there is excellent agreement between the inferred value of s_t and the related speedup obtained from Amdahl's law. For the LS fit, both values p_t and s_t were only bounded between 0 and 1, and the best fit is found to be:

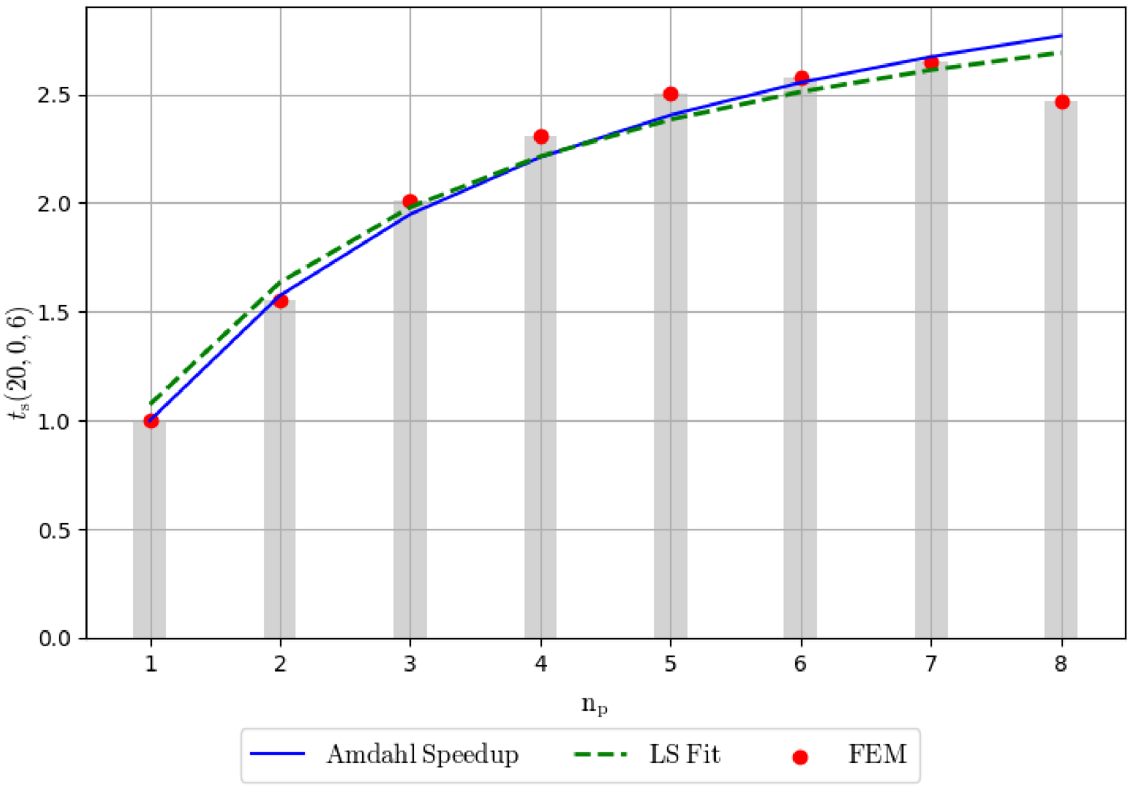

Figure 9.2.7: Strong speedup for a fixed problem size $n_s = 80$, and a variable number of cores n_p. The red points represent the FEM data, the green dashed line is the best LS fit and the solid blue line gives the theoretical speedup from Amdahl's law based on the total serial component s_t. It can be seen that there is once again excellent agreement between Amdahl's law and the inferred values for the total serial and parallel fractions s_t and p_t.

$$s_{LS} \approx 0.29, \quad s_{LS} \approx 0.64. \tag{9.2.29}$$

Once again, the values obtained from the LS fit do not add up to 1, instead leaving a value of 0.07 unaccounted for. As for the case of d_3 , this represents the portion of the code which is neither completely serial nor parallel. This includes the communication time between processors, which grows when increasing n_p but is not present in the computation on one core. It should also be noted that the largest speedup is reached for the case of $n_p = 7$. As for the case of degree = 2, this implies that for $n_p > 7$, the communication between processors balances out the speedup gained from dividing the domain of computation.

With the values for the total serial and parallel fractions s_t and p_t, we can analyse the behaviour under weak scaling. As previously done for the d_3 case, the problem size per compute unit is fixed at $n_s = 10 \cdot n_p$. The results are shown in fig. 9.2.8. Two separate regions can be identified in fig. 9.2.8:

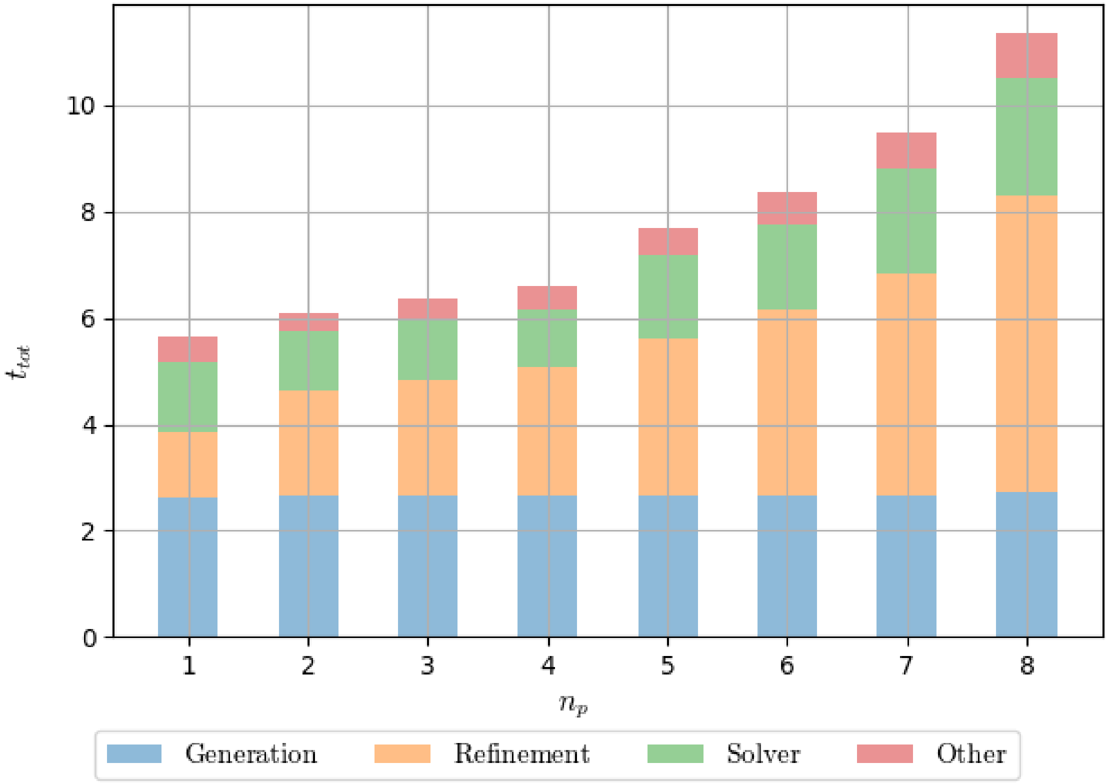

Figure 9.2.8: Weak scaling behaviour for d_1 elements with a fixed problem size per core, $n_s = 10n_p$. The different colours give the computation time for different sections. It can be seen that for $n_p \leq 4$ we have behaviour consistent with the theoretical maximum for weak scaling. On the other hand, for $n_p \geq 5$ there is a linear increase in t_t. This is due to the fact that for $n_s > 4$ the quadratic component of the refinement function can no longer be neglected, so the ratio of the quadratic growth to the linear speedup gives an overall linear increase in the time t_t. This is because for $n_s \geq 50$ the refinement time t_l is the largest fraction of the total time t_t.

- For $n_p \leq 4$, we have an approximately constant computation time t_t, in line with what is expected from the theoretical maximum for weak scaling.
- For $n_p \geq 5$ there is a linear increase in the computation time t_t.

To find the cause of this behaviour, we must consider the LS fit of the scaling for the local refinement function w.r.t. the source number n_s given in chapter 8. This is given as:

$$t_g = 0.236 + 0.086 n_s + 0.0015 n_s^2. \tag{9.2.30}$$

When analysing the case of weak scaling, the assumption is made that the execution time t_t grows linearly w.r.t. the problem size n_s. From eq. 9.2.30 it can be seen that for $n_s \approx 58$, the linear and quadratic contribution have the same value. This means that for $n_p \geq 60$ we can no longer expect a constant computation time for a linear relation between the number of compute units n_p and source number n_s. More specifically, for $n_s = 30$ the linear contribution is $\approx 92\%$ larger than the quadratic contribution, so the difference between them is of order $O(1)$ and the linear term is dominant. On the other hand, for $n_p = 50$ the difference is of $\approx 11\%$, or order $O(0.1)$. It is hence clear that for $n_p \geq 5$ the approximation of a linear growth in computation time with the problem size breaks down, and the two different regimes can hence be clearly explained. In particular, assuming the speedup obtained by increasing n_p is indeed linear, the increase in t_{tot} is in fact expected to be linear, as it stems from dividing a quadratic increase by a linear speedup. Nonetheless, although for $n_p = 8$ the value of t_t is larger than for $n_p = 1$, the two are related by a factor of ≈ 2. This implies that, although weak scaling does not exactly apply to the situation, there is still a speedup of ≈ 4 for $n_s = 80$ w.r.t. serial execution.

Chapter 10

Discussion, Conclusions and Future Work

Throughout this work we have treated three main aspects concerning the MOND theory:

1. Relativistic expansions;

2. Behaviour of the baryonic, apparent and PDM mass distributions in galaxy clusters;

3. Performance and accuracy of FEM solutions of the AQUAL formulation.

Regarding relativistic expansions of MOND, we first studied Hossenfelder's theory of Covariant Emergent Gravity described in [26]. We then examined its connection to Verlinde's theory of Emergent Gravity from [21], the retrieval of a MOND-like potential in the non-relativistic limit, and its agreement with experimental data. We showed that, although CEG introduces a spacetime generalisation for the spatial quantities described in EG, the timelike normalisation chosen for the imposter 4-vector, on which the theory is built, is fundamentally incompatible with Verlinde's EG. Accordingly, we proposed that the vector field of CEG is lightlike, analysed the spacetime resulting from the lightlike field perturbing a Minkowski potential, and presented the geodesic equations. In addition, we showed that the full deep MOND equation can be retrieved from the EOM of CEG in Cartesian coordinates, without needing to assume spherical symmetry. Furthermore, we studied the two possibilities for the behaviour of photons in the spacetime of CEG, which is perturbed by the imposter field. For the case of the photons travelling on the geodesic of the background metric, we referred to the recent detection of concurrent gravitational and EM signals and concluded that, if CEG predicts that the photons travel along the geodesics of the effective metric, the theory is experimentally falsified. This is because any modified theory of gravity postulating the existence of two non-conformally related metrics would predict a time delay between the arrival of the gravitational and EM signals from a neutron star merger such as GW170817, which was experimentally ruled out in [33] and [35]. In addition, we showed that, if instead the photons propagate on the geodesics of the background metric, this implies the possibility of the existence of two distinct event horizons. In this case, the photons could, in principle, go beyond the horizon defined for massive particles and come out unscathed, thus capable of delivering quantum information about the physics past the horizon to a massive particle through an interaction. Furthermore, the propagation of photons along geodesics of a different spacetime from the massive particles would be in direct contradiction with the Equivalence principle, which states that gravity universally couples to all forms of energy.

Due to the numerous inconsistencies listed above, we then followed the work in [40] and [46], and proposed that the relativistic expansion of EG could, in fact, be a Generalised Aether Theory. We showed that, by perturbing a Minkowski background with a spacetime generalisation of the displacement field introduced by Verlinde, it is possible to recover MOND with an arbitrary interpolation function in the non-relativistic limit. Furthermore, we demonstrated that the lightlike normalisation we adopted is consistent with both EG and GEA, and that it is possible that the energy-momentum tensor of the field can reduce, or completely dispense of, the missing mass on the scale of galaxy clusters. In addition, the normalisation we chose results in a covariant theory which is fully compatible with the observations from the GW170817 event.

There are still open questions regarding the ability of GEA theories to describe MOND phenomenology on cosmological backgrounds. With the goal of developing a full-fledged gravitational lensing framework, more work is needed in studying the form of the EOMs for the GEA theory with the EG dressed potential on FLRW backgrounds. In addition, it should be verified whether the energy momentum tensor of the dressed potential can provide a non-negligible contribution to the total energy present in galaxy clusters, as a path to explaining the missing mass that cannot be accounted for by non-relativistic MOND or EG. Ultimately, more work is also needed in identifying predictions, if any exist, that would differ between the non-relativistic limit of GEA and the AQUAL formulation of MOND, in order to have a more concrete direction towards which to focus the more theoretical research.

Regarding the mass distribution in galaxy clusters, we investigated the behaviour of baryonic, PDM, and apparent mass distributions in galaxy clusters in a non-relativistic setting. In particular, we utilised the fully non-linear AQUAL formulation of MOND to analyse the apparent and PDM mass distribution stemming from a baryonic mass distribution including both a gas and galactic component. Following the catalog of Reiprich in [51], and the work done by Moffat and Brownstein in [52], we adopted a King β model for the ICM, under the assumption that the hot gas populating the cluster is close to thermal equilibrium. Furthermore, we utilised data on the galaxy population from the Abell catalog, and selected only the clusters of population groups 0 and 1 which were also contained in the Reiprich catalog. Overall, we simulated 15 galaxy clusters, with baryonic fractions divided more or less equally between gas dominated and galaxy dominated.

From a computational standpoint, we solved the fully non-linear AQUAL MOND PDE through the use of the FEM in the FEniCS software package in Python. For the serial case we showed that, by utilising local mesh refinement around each of the galaxies in the cluster, we could achieve speedups of $O(100)$ w.r.t. a uniformly refined mesh, for a number of galaxies of ≈ 250. In addition, we utilised MPI to allow for the computation of the potential with degree 3 elements, and achieved near-optimal speedups for weak scaling, which hence enabled us to compute a problem size 8 times larger than for the sequential case, with an increase of only 22% in the computation time. This was achieved by utilising the `gmres` Krylov solver paired to the `hypre-euclid` parallel pre-conditioner. All the simulations were run on a single-threaded 8-core Intel i7-9700 CPU with maximum frequency $f_{max} = 4.7GHz$, $8GB$ of RAM and $64KB$, $256KB$ and $12MB$ of L1, L2 and L3 cache respectively. Our implementation is the first to have treated the fully non-linear MOND PDE with interpolation function for the case of galaxy clusters through FEM.

With regards to the results for the mass distributions, we analysed the integrated mass for two different situations: a set of concentric spheres around the center of the cluster, and the sum of the set of spheres enclosing each galaxy. In the first case, we verified that the integrated mass has a behaviour that deviates both from the case of spherical symmetry and from the deep MOND regime, since the central regions of the clusters are neither spherically symmetric nor in the deep MOND regime. We saw that, in all clusters, up to a radius which varies from 1/10 to 1/2 of the radius of the cluster, the baryons accounted for a larger fraction of the total apparent mass when compared to the PDM. This demonstrates that the PDM mass distribution is not peaked in the center of the cluster, and that it does not exclusively follow the distribution of the ICM, even in clusters where the ICM is the dominant baryonic component. For the second case, we verified that within the critical radius around each galaxy, the apparent mass is virtually zero. Moreover, we saw that, regardless of the fraction of total baryonic matter contained in each radius, the PDM is always close to zero in proximity of the galaxies. On the other hand, the PDM mass grows approximately linearly in the radius for all clusters, showing that, regardless of the baryonic fraction contained around the galaxies, the PDM tends to clump around the galaxies, which will cause an abundance of smaller gravitational lenses farther from the central region of the cluster. Lastly, we verified the theoretical prediction that both the PDM and apparent mass distributions can take on negative values, which come in the form of tori around the galaxies. We showed that the transition between the positive and negative distributions is smooth, so that the galaxies are surrounded by tori of negative mass density, through which a halo of positive mass PDM encapsulates each galaxy. Ultimately, our simulations confirmed a number of phenomena, among which some have not yet been attributed to MOND. First, the concentration of the PDM is predominantly around the galaxies even when the ICM is the dominant component. Second, it is possible for both the PDM and apparent mass to have a negative mass density which, if observed, could in no way be explained by dark matter models. Third, there are no peaks of dark matter in the central regions of the clusters and, within the critical radius of every galaxy, no PDM is present at all. All of these results were made possible by the use of a numerical method, FEM, capable of treating discrete and continuous solution in the same PDE but on a different footing, without smoothing the discrete distribution.

In this work, we carried out our treatment of galaxy clusters exclusively through the AQUAL formulation of MOND with the standard interpolation function. As was previously mentioned, if the boundary of a domain is not in the deep MOND regime, the mass contained in that volume is determined by the value of the interpolation function on the boundary. Therefore, it would be of interest to study the apparent mass distributions generated other MOND interpolation functions, such as the simple and exponential interpolation functions. Moreover, for this work we limited our sample to 15 clusters, in order to ensure that we could describe the ICM and the galaxy component of the baryonic mass in a realistic scenario. However, given that most galaxy mass measurements already include the contribution of the apparent dark matter halo, it is challenging to find raw data obtained, for example, by unbiased luminosity measurements. An interesting alternative would be to simulate clusters in which each galaxy has unique mass and radii, in order to understand how the PDM distribution behaves under a variation in the scale of the critical radius w.r.t. the galactic radius. Finally, the gas distribution that was used to model the ICM is known to be divergent, and to make the rather stringent assumption that the gas is close to thermal equilibrium. In the ideal case, it would be best to simulate a gas distribution derived entirely by principles stemming from MOND, rather than Newtonian gravity.

From the computational side, the possibility of having a large distributed memory computer available would allow for the simulation of much larger systems, and to greater accuracy. However, in that case, it would be paramount to have access to libraries capable of mesh generation and partitioning in parallel, as well as the ability to change the mesh partitioning criteria, especially after local mesh refinement. In fact, with the libraries used for this work, the mesh has to be re-partitioned every time that local mesh refinement is carried out, which would definitely represent a bottleneck if working with a larger distributed memory system. With the access to more computational power, a useful application would be to achieve a resolution comparable to the simulations carried out in this work, for each of the galaxies in a cluster, in order to more precisely search for effects, such as negative mass distributions, which could be observationally verifiable. Once again, for such a purpose a parallel mesh partitioning library would be necessary, which is capable of taking into account mesh refinement and split the domain not only according to the number of cells, but also user input, with the goal of having a separate processor allocated to each of the $O(1000)$ galaxies contained in the cluster.

Appendix A

CEG in Schwarzschild spacetime

A.1. The spherically symmetric covariant case: Schwarzschild geometry

After determining that the EOM in a flat background are plagued by numerous issues, the next step is to attempt to find an explicit solution in a curved background that can provide valid results. It must be noted that in Hossenfelder's approach the effect of the imposter field on the metric is completely neglected, so this amounts to a *test field* treatment[1] . All assumptions made previously for the spherically symmetric, flat spacetime case still hold, with the only obvious difference being the use of the Schwarzschild metric, replacing Minkowski.

Throughout the flat spacetime analysis, the assumption of a static field greatly simplified calculations due to vanishing partial derivatives with respect to time, and the lack of any non-zero Christoffel symbols including t. In Schwarzschild spacetime, one has to instead consider the two non vanishing Christoffels symbols containing t:

$$\Gamma^{r}_{tt} = \frac{GM\left(r-2GM\right)}{r^{3}}, \quad \Gamma^{t}_{tr} = \Gamma^{t}_{rt} = \frac{GM}{r\left(r-2GM\right)}. \tag{A.1.1}$$

The symmetry of the Christoffel symbols in the two lower indices was noted[2]:

$$\Gamma^{\alpha}_{\mu\nu} = \Gamma^{\alpha}_{\nu\mu}. \tag{A.1.2}$$

It is useful at this point to introduce another important property, that of **metric compatibility**:

[1] The notation is analogous to that of a test particle, whose equations of motion do not account for the effect of its own energy on the background curvature)

[2] The symmetry of the Christoffel symbols in the lower indices is the reason the definition of the Riemann tensor can be simplified to that of the commutator of covariant derivatives. The extra term present from commuting covariant derivatives is the torsion tensor, which is equal to the commutator of Christoffel symbols in the two lower indices. This commutator always vanishes in GR.

Since Christoffels "adjust" partial derivatives to make them independent of the used coordinate system, they ensure that the covariant derivative of the metric vanishes, so that the effect of curvature on differentiation is fully absorbed in the Christoffel.

Mathematically, this is expressed as:

$$\boxed{\nabla_\mu g_{\alpha\beta} = 0.} \tag{A.1.3}$$

Armed with these two notions, one can now find an explicit formula for the EOM in a curved background. Starting from the kinetic term χ, one obtains:

$$\chi = \frac{4}{3}\overbrace{(\nabla_\nu u^\nu)(\nabla_\mu u^\mu)}^{A} - \frac{1}{2}\overbrace{(\nabla_\mu u_\nu)(\nabla^\mu u^\nu)}^{B} - \frac{1}{2}\overbrace{(\nabla_\mu u_\nu)(\nabla^\nu u^\mu)}^{C} \tag{A.1.4}$$

$$A = (\nabla_\alpha u^\alpha)\left(\nabla_\beta u^\beta\right) \to \nabla_\alpha u^\alpha = \nabla_\beta u^\beta = \tag{A.1.5}$$

$$\partial_\alpha u^\alpha + \Gamma^\alpha_{\alpha\lambda} u^\lambda = \Gamma^\alpha_{\alpha 0} u^0 = 0 \,\forall\, \alpha \tag{A.1.6}$$

$$C = (\nabla_\alpha u_\beta)\left(\nabla^\beta u^\alpha\right) = \left(\nabla_\alpha u^\beta\right)(\nabla_\beta u^\alpha) = \tag{A.1.7}$$

$$\left(\nabla_i u^0\right)\left(\nabla_0 u^i\right) + \left(\nabla_0 u^i\right)\left(\nabla_i u^0\right) = 0. \tag{A.1.8}$$

For C, $u^i = 0$ was used. The B term gives the only non-vanishing contribution, as in the case of flat spacetime:

$$B = (\nabla_\alpha u_\beta)\left(\nabla^\alpha u^\beta\right) = (\nabla_\alpha\, g_{\beta\nu} u^\nu)\left(g^{\rho\alpha}\nabla_\rho u^\beta\right) = \tag{A.1.9}$$

$$g_{\beta\nu} g^{\rho\alpha}\left(\nabla_\alpha u^\nu\right)\left(\nabla_\rho u^\beta\right). \tag{A.1.10}$$

Metric compatibility was used. The best way to proceed is to evaluate the expression term by term, by looking at each non-zero index combination of the metric and its inverse, which yields:[3]

$$g_{\beta\nu} g^{\rho\alpha}\left(\nabla_\alpha u^\nu\right)\left(\nabla_\rho u^\beta\right) = \tag{A.1.11}$$

$$g_{\beta\nu} g^{\rho\alpha}\left(\partial_\alpha u^\nu + \Gamma^\nu_{\alpha\sigma} u^\sigma\right)\left(\partial_\rho u^\beta + \Gamma^\beta_{\rho\eta} u^\eta\right) = \tag{A.1.12}$$

$$g_{00} g^{00}\left(\partial_0 u^0 + \Gamma^0_{0\sigma} u^\sigma\right)\left(\partial_0 u^0 + \Gamma^0_{0\eta} u^\eta\right) + g_{ii} g^{jj}\left(\partial_j u^i + \Gamma^i_{j\sigma} u^\sigma\right)\left(\partial_j u^i + \Gamma^i_{j\eta} u^\eta\right) + \tag{A.1.13}$$

$$g_{00} g^{jj}\left(\partial_j u^0 + \Gamma^0_{j\sigma} u^\sigma\right)\left(\partial_j u^0 + \Gamma^0_{j\eta} u^\eta\right) + g_{ii} g^{00}\left(\partial_0 u^i + \Gamma^i_{0\sigma} u^\sigma\right)\left(\partial_0 u^i + \Gamma^i_{0\eta} u^\eta\right). \tag{A.1.14}$$

Working out one term explicitly as an example:

$$g_{ik} g^{lm}\left(\partial_m u^k\right)\left(\partial_l u^i\right) = \begin{cases} g_{ii} g^{ll}\left(\partial_l u^i\right)\left(\partial_l u^i\right) & i = k, \quad l = m \\ 0 & \text{otherwise.} \end{cases} \tag{A.1.15}$$

[3] In all terms apart from the first, an ill-defined Einstein summation is used, with two contravariant and two covariant indices equal to each other. Although formally not allowed, this abuse of notation gives a sum yielding the correct value since $g_{\mu\nu}$ is diagonal.

Simplifying as before, and using $g_{00}g^{00} = 1$, the expression becomes:[4]

$$\overbrace{\left(\Gamma^0_{0\sigma}u^\sigma\right)\left(\Gamma^0_{0\eta}u^\eta\right)}^{F} + \overbrace{g_{ii}g^{jj}\left(\Gamma^i_{j\sigma}u^\sigma\right)\left(\Gamma^i_{j\eta}u^\eta\right)}^{G} + \tag{A.1.16}$$

$$\overbrace{g_{00}g^{jj}\left(\partial_j u^0 + \Gamma^0_{j\sigma}u^\sigma\right)\left(\partial_j u^0 + \Gamma^0_{j\eta}u^\eta\right)}^{H} + \overbrace{g_{ii}g^{00}\left(\Gamma^i_{0\sigma}u^\sigma\right)\left(\Gamma^i_{0\eta}u^\eta\right)}^{I}. \tag{A.1.17}$$

It is most convenient to evaluate each term individually.[5]

$$F = \left(\Gamma^0_{0\sigma}u^\sigma\right)\left(\Gamma^0_{0\eta}u^\eta\right) \xRightarrow{\Gamma^0_{00}=0} 0 \tag{A.1.18}$$

$$G = g_{ii}g^{jj}\left(\Gamma^i_{j\sigma}u^\sigma\right)\left(\Gamma^i_{j\eta}u^\eta\right) \xRightarrow{\Gamma^i_{j0}=0\,\forall\, i,j} 0 \tag{A.1.19}$$

$$H = g_{00}g^{jj}\left(\partial_j u^0 + \Gamma^0_{j\sigma}u^\sigma\right)\left(\partial_j u^0 + \Gamma^0_{j\eta}u^\eta\right) \xRightarrow{\sigma=\eta=0,\, j=r} g_{00}g^{rr}\left(\partial_r u^0 + \Gamma^0_{r0}u^0\right)^2 \tag{A.1.20}$$

$$I = g_{ii}g^{00}\left(\Gamma^i_{0\sigma}u^\sigma\right)\left(\Gamma^i_{0\eta}u^\eta\right) \xRightarrow{\sigma=\eta=0,\, i=r} g_{rr}g^{00}\left(\Gamma^r_{00}u^0\right)^2. \tag{A.1.21}$$

To simplify further, one looks at the explicit form of the Schwarzschild metric, for which the spacetime interval reads:

$$\boxed{\mathrm{d}s^2 = -\gamma(r)\mathrm{d}t^2 + \gamma(r)^{-1}\mathrm{d}r^2 + r^2\mathrm{d}\theta^2 + r^2\sin^2(\theta)\,\mathrm{d}\varphi^2}. \tag{A.1.22}$$

The quantity γ is defined as:

$$\boxed{\gamma = \frac{r - 2GM}{r}.} \tag{A.1.23}$$

Here, M is the mass at the origin of the coordinate system, and r is the radial distance from the origin. An important simplification can now be made, firstly by noting that the Schwarzschild metric used is diagonal, so that the inverse metric is also diagonal, with each component replaced by its reciprocal:

$$g^{\mu\nu} = g^{-1}_{\mu\nu}. \tag{A.1.24}$$

It can be seen that $g_{00} = -g^{rr}$, giving:

$$g_{00}g^{rr} = -g^2_{00} = -\gamma^2. \tag{A.1.25}$$

Another useful simplification can be obtained by expressing the Christoffel symbols in terms of γ:

$$\Gamma^0_{r0} = \frac{GM}{r^2\gamma}, \quad \Gamma^r_{00} = \frac{\gamma GM}{r^2}. \tag{A.1.26}$$

[4] It can be shown that for a metric that can be written in diagonal form, $g_{\mu\nu} = \left(g^{\mu\nu}\right)^{-1}$. Consequently, one has the identities: $g^{aa} = g_{aa} \;\; \forall a$, and $g_{\mu\nu}g^{\mu\nu} = 4$

[5] the arrows give the index values for which each summation is non-zero, to avoid writing out terms that have a zero contribution.

It is now possible to rewrite the kinetic term as:

$$\chi = -\frac{1}{2}B = -\frac{1}{2}(H+I) = \tag{A.1.27}$$

$$-\frac{1}{2}\left(g_{00}g^{rr}\left(\partial_r u^0 + \Gamma^0_{r0}u^0\right)^2 + g_{rr}g^{00}\left(\Gamma^r_{00}u^0\right)^2\right) = \tag{A.1.28}$$

$$\frac{1}{2}\left(g^2_{00}\left(\partial_r u^0 + \Gamma^0_{r0}u^0\right)^2 + \frac{\left(\Gamma^r_{00}u^0\right)^2}{g^2_{00}}\right). \tag{A.1.29}$$

In the last line A.1.25 was used. Making use of A.1.22 and re-writing the Christoffel symbols according to A.1.26 gives:

$$\chi = \frac{1}{2}\left(\gamma^2\left(\partial_r u^0 + \frac{GMu^0}{r^2\gamma}\right)^2 + \left(\frac{\gamma GMu^0}{r^2}\right)^2\frac{1}{\gamma^2}\right) = \tag{A.1.30}$$

$$\boxed{\frac{1}{2r^4}\left(\left(\gamma r^2\partial_r u^0 + GMu^0\right)^2 + \left(GMu^0\right)^2\right).} \tag{A.1.31}$$

The following must be pointed out:

The result for χ does not match what found in [26].

This implies that the EOM will also differ by those found by Hossenfelder. The other terms necessary to calculate the equations of motion are ϵ and $\epsilon_{\mu\nu}$. It is easy to see that the former vanishes:

$$\epsilon = \epsilon^\mu_\mu = \left(\nabla_\mu u^\mu + \nabla_\mu u^\mu\right) = 2\nabla_\mu u^\mu = 2\left(\partial_\mu u^\mu + \Gamma^\mu_{\mu\nu}u^\nu\right) = 0 \quad \forall\ \mu. \tag{A.1.32}$$

For the latter, it is noted that it only appears in the E.O.M.s in the combination:

$$\nabla_\mu\left(\sqrt{\chi}\epsilon^\mu_\sigma\right) = \sqrt{\chi}\nabla_\mu\epsilon^\mu_\sigma + \epsilon^\mu_\sigma\nabla_\mu\sqrt{\chi}. \tag{A.1.33}$$

Since $\nabla_\mu\sqrt{\chi} = \partial_\mu\sqrt{\chi}$ vanishes $\forall\,\mu \neq r$, one only has to compute ϵ^r_σ, which is done as follows:

$$\epsilon^\mu_\sigma = \frac{1}{2}\left(\overbrace{\nabla_\sigma u^\mu}^{J} + \overbrace{\nabla^\mu u_\sigma}^{K}\right). \tag{A.1.34}$$

Evaluating each term separately gives:

$$J = \nabla_\sigma u^\mu \xrightarrow{\mu=r} \partial_\sigma u^r + \Gamma^r_{\sigma\lambda}u^\lambda \xrightarrow{\sigma=0} \Gamma^r_{00}u^0 \tag{A.1.35}$$

$$K = \nabla^\mu u_\sigma = g^{\mu\nu}g_{\rho\sigma}\left(\partial_\nu u^\rho + \Gamma^\rho_{\nu\lambda}u^\lambda\right) \xrightarrow{\mu=r,\,\rho=\lambda=0} g^{rr}g_{00}\left(\partial_r u^0 + \Gamma^0_{r0}u^0\right) = \tag{A.1.36}$$

$$-\gamma^2\left(\partial_r u^0 + \Gamma^0_{r0}u^0\right). \tag{A.1.37}$$

Overall this yields, for $\sigma = 0$:

$$\epsilon^r_0 = \Gamma^r_{00} u^0 - \gamma^2 \left(\partial_r u^0 + \Gamma^0_{r0} u^0 \right). \tag{A.1.38}$$

On the other hand, the divergence $\nabla_\mu \epsilon^\mu_\sigma$ has to be calculated in full:

$$\nabla_\mu \epsilon^\mu_\sigma = \overbrace{\nabla_\mu \left(\nabla_\sigma u^\mu \right)}^{L} + \overbrace{\nabla_\mu \nabla^\mu u_\sigma}^{M}. \tag{A.1.39}$$

The first term needed from A.1.39 is:

$$L = \nabla_\mu \left(\nabla_\sigma u^\mu \right). \tag{A.1.40}$$

As noted previously, covariant derivatives do not commute. To obtain the value of L, it is best to treat the first covariant derivative as a (1,1) tensor, take its covariant derivative, and plug the explicit form back in at the end of the calculation. By setting

$$A^\mu_\sigma \equiv \nabla_\sigma u^\mu, \tag{A.1.41}$$

A.1.40 becomes:

$$L = \nabla_\mu \left(A^\mu_\sigma \right) = \partial_\mu \left(A^\mu_\sigma \right) - \Gamma^\lambda_{\mu\sigma} A^\mu_\lambda + \Gamma^\mu_{\mu\lambda} A^\lambda_\sigma = \tag{A.1.42}$$

$$\partial_\mu \left(\nabla_\sigma u^\mu \right) - \Gamma^\lambda_{\mu\sigma} \nabla_\lambda u^\mu + \Gamma^\mu_{\mu\lambda} \nabla_\sigma u^\lambda. \tag{A.1.43}$$

The definition for the covariant derivative for a higher rank tensor was used (see e.g. [6])[6]. Expanding each term so to calculate them explicitly:

$$L = \overbrace{\partial_\mu \left(\partial_\sigma u^\mu + \Gamma^\mu_{\sigma\lambda} u^\lambda \right)}^{N} - \overbrace{\Gamma^\lambda_{\mu\sigma} \left(\partial_\lambda u^\mu + \Gamma^\mu_{\lambda\alpha} u^\alpha \right)}^{O} + \overbrace{\Gamma^\mu_{\mu\lambda} \left(\partial_\sigma u^\lambda + \Gamma^\lambda_{\sigma\alpha} u^\alpha \right)}^{P} \tag{A.1.44}$$

$$N = \partial_\mu \partial_\sigma u^\mu + \partial_\mu \left(\Gamma^\mu_{\sigma\lambda} u^\lambda \right) \xrightarrow{\lambda=0,\, \sigma=0} \partial_r \left(\Gamma^r_{00} u^0 \right) = \Gamma^r_{00} \partial_r u^0 + u^0 \partial_r \Gamma^r_{00}. \tag{A.1.45}$$

In evaluating N it was noted that partial derivatives commute, and as previously showed $\partial_\mu u^\mu = 0$. The remaining terms give:

$$O = \Gamma^\lambda_{\mu\sigma} \left(\partial_\lambda u^\mu + \Gamma^\mu_{\lambda\alpha} u^\alpha \right) \xrightarrow{\mu=\alpha=\sigma=0} \Gamma^\lambda_{00} \partial_\lambda u^0 + \Gamma^\lambda_{\mu0} \Gamma^\mu_{\lambda0} u^0 = \tag{A.1.46}$$

$$\Gamma^r_{00} \partial_r u^0 + \left(\Gamma^r_{00} \Gamma^0_{r0} + \Gamma^0_{r0} \Gamma^r_{00} \right) u^0 = \Gamma^r_{00} \left(\partial_r u^0 + 2 \Gamma^0_{0r} u^0 \right), \tag{A.1.47}$$

and:

$$P = \Gamma^\mu_{\mu\lambda} \left(\partial_\sigma u^\lambda + \Gamma^\lambda_{\sigma\alpha} u^\alpha \right) \xrightarrow{\mu=\alpha=\sigma=0,\, \lambda=r} \Gamma^0_{0\lambda} \left(\partial_0 u^r + \Gamma^\lambda_{00} u^0 \right) = \Gamma^0_{0r} \Gamma^r_{00} u^0. \tag{A.1.48}$$

[6] One can see that the covariant derivative for a higher rank tensor includes one partial and derivative and a number of Christoffels equal to the rank. In the procedure, each index is treated as if describing a corresponding (co)vector.

On the other hand, M could be most easily evaluated by using the Laplace-Beltrami operator:

$$M = \nabla_\mu \nabla^\mu u_\sigma = \frac{1}{\sqrt{|g|}} \partial_\mu \left(g^{\mu\nu} \sqrt{|g|} \partial_\nu u_\sigma \right). \tag{A.1.49}$$

For the Schwarzschild metric:

$$\sqrt{|g|} = \left| r^2 \sin\theta \right|. \tag{A.1.50}$$

Since spherical symmetry is assumed, this results in:

$$\frac{1}{2} \partial_r \left(g^{rr} r^2 \partial_r g_{\sigma\rho} u^\rho \right) \xrightarrow{\sigma=\rho=0} -\frac{1}{r^2} \partial_r \left[\gamma r^2 \partial_r \left(\gamma u^0 \right) \right] = \tag{A.1.51}$$

$$-\frac{1}{r^2} \partial_r \left(\gamma^2 r^2 \partial_r u^0 + 2GM\gamma u^0 \right). \tag{A.1.52}$$

In the last step the derivative of γ was used,

$$\partial_r \gamma = \partial_r \left(1 - \frac{2GM}{r} \right) = \frac{2GM}{r^2}. \tag{A.1.53}$$

However, an important observation should be made at this point. The explicit evaluation of the imposter field through its E.O.M.s is needed to ultimately calculate a perturbation to a background metric in order to apply the formalism of gravitational lensing.

In this formalism (see e.g. [6]), the perturbation of the background metric is only considered up to first order. This means that the metric used to raise indices is in fact the background metric, and only coefficients linear in the perturbation are kept, with second order terms such as products of Christoffel symbols neglected in the approximation, whereas the first derivatives of the Christoffel symbols are kept.

Although in the calculation that was just carried out the indices were raised with the full Schwarzschild metric (as necessary to make contact with the calculations from [26]), terms quadratic in the Christoffels can be neglected. As χ and ϵ_0^r have already been expressed in terms of Christoffel symbols, it is hence necessary, in order to obtain a coherent approximation, to also evaluate the Laplace Beltrami term with respect to the Christoffel symbols without utilising the formula from A.1.49 directly, instead using the approach outlined for L. M is then calculated setting $B_\nu^\rho \equiv \nabla_\nu u^\rho$, giving:

$$\nabla_\mu \left(\nabla^\mu u_\sigma \right) = g^{\mu\nu} g_{\rho\sigma} \nabla_\mu \left(\nabla_\nu u^\rho \right) = g^{\mu\nu} g_{\rho\sigma} \left(\overbrace{\partial_\mu B_\nu^\rho}^{Q} + \overbrace{\Gamma^\rho_{\mu\lambda} B_\nu^\lambda}^{R} - \overbrace{\Gamma^\lambda_{\mu\nu} B_\lambda^\rho}^{S} \right). \tag{A.1.54}$$

Evaluating each term individually:

$$g^{\mu\nu} g_{\rho\sigma} Q = g^{\mu\nu} g_{\rho\sigma} \left(\partial_\mu B_\nu^\rho \right) = \tag{A.1.55}$$

$$g^{\mu\nu} g_{\rho\sigma} \partial_\mu \partial_\nu u^\rho + g^{\mu\nu} g_{\rho\sigma} \partial_\mu \left(\Gamma^\rho_{\nu\lambda} u^\lambda \right) \xrightarrow{\sigma=\lambda=0,\, \mu=r} g^{rr} g_{00} \left(\partial_r^2 u^0 + \partial_r \left(\Gamma^0_{r0} u^0 \right) \right), \tag{A.1.56}$$

$$g^{\mu\nu} g_{\rho\sigma} R = g^{\mu\nu} g_{\rho\sigma} \left(\Gamma^\rho_{\mu\lambda} B_\nu^\lambda \right) = \tag{A.1.57}$$

$$g^{\mu\nu} g_{\rho\sigma} \Gamma^\rho_{\mu\lambda} \partial_\nu u^\lambda + g^{\mu\nu} g_{\rho\sigma} \Gamma^\rho_{\mu\lambda} \Gamma^\lambda_{\nu\alpha} u^\alpha \xrightarrow{\lambda=0,\nu=\mu=r;\, \lambda=r,\nu=\mu=0} \tag{A.1.58}$$

$$g^{rr} g_{00} \Gamma^0_{r0} \partial_r u^0 + g^{rr} g_{00} \Gamma^0_{r0} \Gamma^0_{r0} u^0 + g^{00} g_{00} \Gamma^0_{0r} \Gamma^r_{00} u^0. \tag{A.1.59}$$

The second arrow describes the two possible sets of nonzero indices and, once again, the only nonzero contributions are given by $\sigma = 0$. Finally, for the last term:

$$g^{\mu\nu} g_{\rho\sigma} S = g^{\mu\nu} g_{\rho\sigma} \left(\Gamma^{\lambda}_{\mu\nu} B^{\rho}_{\lambda}\right) = \tag{A.1.60}$$

$$g^{\mu\nu} g_{\rho\sigma} \Gamma^{\lambda}_{\mu\nu} \left(\nabla_{\lambda} u^{\rho}\right) \xrightarrow{\rho=\mu=\nu=0,\lambda=r}= g^{00} g_{00} \Gamma^{r}_{00} \partial_r u^0 + g^{rr} g_{00} \Gamma^{r}_{00} \Gamma^{0}_{r0} u^0. \tag{A.1.61}$$

It is now possible to approximate all quantities calculated so far to first order in the Christoffel symbols:

$$\chi \approx \frac{1}{2}\gamma^2 \left[\left(\partial_r u^0\right)^2 + 2\partial_r u^0 \Gamma^{0}_{r0} u^0\right], \quad \sqrt{\chi} = \frac{-\gamma}{\sqrt{2}} \left|\partial_r u^0 + \Gamma^{0}_{r0} u^0\right| \tag{A.1.62}$$

$$\epsilon^{r}_{0} \approx \Gamma^{r}_{00} u^0 - \gamma^2 \left(\partial_r u^0 + \Gamma^{0}_{r0} u^0\right) \tag{A.1.63}$$

$$\nabla_{\mu} \epsilon^{\mu}_{0} \approx -\gamma^2 \left(\partial^2_r u^0 + \partial_r \left(\Gamma^{0}_{r0} u^0\right) + \Gamma^{0}_{r0} \partial_r u^0\right) - \Gamma^{r}_{00} \partial_r u^0 + u^0 \partial_r \Gamma^{r}_{00}. \tag{A.1.64}$$

Before going any further, one must note the following problem:

> The value of χ determines the expression resulting in a PDE for the potential ϕ, and χ one found here does not match that obtained by Hossenfelder.

This implies that the potential obtained by using the χ derived here will generally not match the potential obtained by Hossenfelder, which is to first order logarithmic. Furthermore, there is no indication in Verlinde's original work that the displacement field would have the same form on a Schwarzschild background. Finally, the same assumption is made about the rest frame of the imposter field as for the nonrelativistic case, carrying along with it the problems described for that limit. Due to these issues, the Schwarzschild case is not worth pursuing further.

Appendix B

Quantities for the Lightlike Field EOM

If one decides to in fact fix the vector field to have a lightlike normalisation, the time component is uniquely dependent on the chosen metric. A clear candidate is the metric that was obtained in section 3.4. Throughout this appendix, all quantities necessary to compute the EOM for a lightlike field on this background are given. These calculations are not included in the main body as a full solution yielding a logarithmic potential in the correct regime could not be found, and other avenues were followed.

B.1. Quantities for the Lightlike Field EOM

With the Christoffel symbols available, the next step is to calculate the E.O.M.s for u^μ, deviating now completely from the field defined by Hossenfelder in [26] and instead using a light-like normalisation. In section **??**, u^0 was obtained by fixing the light-like normalisation of u^μ in a Minkowski background. On the other hand, for the E.O.M.s it is desirable to obtain the normalisation directly from the perturbed metric, initially without approximating. One then gets, using the general case of Cartesian coordinates for the spatial component of the metric:

$$u_\mu u^\mu = g_{\mu\nu} u^\nu u^\mu = g_{00} u^0 u^0 + g_{ii} u^i u^i = -\left(1+2\phi\right)\left(u^0\right)^2 + \left(1-2\phi\right) u^i u^i = 0 \tag{B.1.1}$$

$$\left(1+2\phi\right)\left(u^0\right)^2 = \left(1-2\phi\right)\phi^2 L^2 \to u^0 = \sqrt{\frac{1-2\phi}{1+2\phi}}\phi L, \tag{B.1.2}$$

The positive root was taken for u^0. The 4-vector u^μ now reads:

$$u^\mu = \phi L \left(\sqrt{\frac{1-2\phi}{1+2\phi}}, n^i\right). \tag{B.1.3}$$

As spherical symmetry is assumed, and one considers the effect of a single mass at the origin of the coordinate system on an individual test particle, the coordinate system can be aligned so that one of the axes corresponds to the vector connecting the source mass to the test particle, which then gives the simplified expression:

$$u^\mu = \phi L \left(\sqrt{\frac{1-2\phi}{1+2\phi}}, 1, 0, 0 \right). \tag{B.1.4}$$

With this definition, it is possible to again evaluate all the terms needed to form the Lagrangian for the free field, with the difference being that the metric used is no longer Minkowski, and that u^μ has a non-vanishing spatial component, exclusively in the r direction. Whereas it is convenient for now to delay the choice of coefficients for the kinetic term, given that the arguments of Hossenfelder no longer apply, the contractions of the derivative terms have to be evaluated all the same, so it is good to start from these.

$$(\nabla_\mu u^\mu) = \overbrace{\partial_\mu u^\mu}^{A} + \overbrace{\Gamma^\mu_{\mu\alpha} u^\alpha}^{B} \tag{B.1.5}$$

$$A = \partial_\mu u^\mu \xrightarrow{\mu=r} \partial_r u^r \tag{B.1.6}$$

$$B = \Gamma^\mu_{\mu\alpha} \xrightarrow{\alpha=r} \Gamma^\mu_{\mu r} u^r = \left(\Gamma^t_{tr} + \Gamma^r_{rr} + 2\Gamma^\theta_{\theta r}\right) u^r, \tag{B.1.7}$$

where it was used that $\Gamma^\theta_{\theta r} = \Gamma^\varphi_{\varphi r}$. Hence, to 1^st order in ϕ (and hence to 1^st order in u^r), one has:

$$(\nabla_\mu u^\mu)^2 \approx (\partial_r u^r)^2 + 2u^r \left(\Gamma^t_{tr} + \Gamma^r_{rr} + 2\Gamma^\theta_{\theta r}\right) \partial_r u^r. \tag{B.1.8}$$

The next term is:

$$(\nabla_\mu u_\nu)(\nabla^\nu u^\mu) = (\nabla_\mu u^\nu)(\nabla_\nu u^\mu) = \left(\partial_\mu u^\nu + \Gamma^\nu_{\mu\alpha} u^\alpha\right)\left(\partial_\nu u^\mu + \Gamma^\mu_{\nu\beta} u^\beta\right) = \tag{B.1.9}$$

$$\overbrace{\partial_\mu u^\nu \partial_\nu u^\mu}^{C} + \overbrace{(\partial_\mu u^\nu)\Gamma^\mu_{\nu\beta} u^\beta}^{D} + \overbrace{\Gamma^\nu_{\mu\alpha} u^\alpha (\partial_\nu u^\mu)}^{E} + \overbrace{\Gamma^\nu_{\mu\alpha} u^\alpha \Gamma^\mu_{\nu\beta} u^\beta}^{F} \tag{B.1.10}$$

$$C \xrightarrow{\mu=\nu=r} (\partial_r u^r)^2 \tag{B.1.11}$$

$$D \xrightarrow{\mu=r} (\partial_r u^r)\Gamma^r_{rr} u^r + \left(\partial_r u^t\right)\Gamma^r_{tt} u^t, \quad E = D. \tag{B.1.12}$$

For the last term F, we note that all non-zero contributions are of the form $u^\alpha u^\beta$, hence 2^{nd} order in ϕ and can hence be neglected, so that one has:

$$(\nabla_\mu u_\nu)(\nabla^\nu u^\mu) \approx (\partial_r u^r)^2 + 2\left((\partial_r u^r)\Gamma^r_{rr} u^r + \left(\partial_r u^t\right)\Gamma^r_{tt} u^t\right) \tag{B.1.13}$$

The last contraction of the derivative term is:

$$(\nabla_\mu u_\nu)(\nabla^\mu u^\nu) = g^{\lambda\mu} g_{\kappa\nu} (\nabla_\lambda u^\kappa)(\nabla_\mu u^\nu) = g^{\lambda\mu} g_{\kappa\nu} \left(\partial_\lambda u^\kappa + \Gamma^\kappa_{\lambda\alpha} u^\alpha\right)\left(\partial_\mu u^\nu + \Gamma^\nu_{\mu\beta} u^\beta\right) = \tag{B.1.14}$$

$$g^{\lambda\mu} g_{\kappa\nu} \left(\overbrace{\partial_\lambda u^\kappa \partial_\mu u^\nu}^{G} + \overbrace{\partial_\lambda u^\kappa \Gamma^\nu_{\mu\beta} u^\beta}^{H} + \overbrace{\Gamma^\kappa_{\lambda\alpha} u^\alpha \partial_\mu u^\nu}^{I} + \overbrace{\Gamma^\kappa_{\lambda\alpha} u^\alpha \Gamma^\nu_{\mu\beta} u^\beta}^{J} \right) \tag{B.1.15}$$

$$g^{\lambda\mu} g_{\kappa\nu} G \xrightarrow{\lambda=\mu=r,} g^{rr} \left(g_{rr} (\partial_r u^r)^2 + g_{tt} \left(\partial_r u^t\right)^2 \right) = (\partial_r u^r)^2 + g^{rr} g_{tt} \left(\partial_r u^t\right)^2 \tag{B.1.16}$$

where in the last step it was used that the metric is diagonal and therefore $g_{rr}^{-1} = g^{rr}$. Continuing with the calculation:

$$g^{\lambda\mu}g_{\kappa\nu}H \xrightarrow{\lambda=\mu=r} \left(\partial_r u^r\right)\Gamma^r_{rr}u^r + g^{rr}g_{tt}\left(\partial_r u^t\right)\Gamma^t_{rt}u^t, \quad I = H. \tag{B.1.17}$$

Once again, J only has non-zero contributions of the form $u^\alpha u^\beta$, which can be neglected as they are 2^{nd} order in ϕ, giving overall:

$$\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) \approx \left(\partial_r u^r\right)^2 + g^{rr}g_{tt}\left(\partial_r u^t\right)^2 + 2\left(\left(\partial_r u^r\right)\Gamma^r_{rr}u^r + g^{rr}g_{tt}\left(\partial_r u^t\right)\Gamma^t_{rt}u^t\right) \tag{B.1.18}$$

The next term appearing in the E.O.M.s is the derivative of the strain tensor:

$$\nabla_\nu \epsilon^\nu_\mu = \frac{1}{2}\left(\overbrace{\nabla_\nu\left(\nabla_\mu u^\nu\right)}^{L} + \overbrace{\nabla_\nu\left(\nabla^\nu u_\mu\right)}^{M}\right), \tag{B.1.19}$$

which is best calculated by treating the first covariant derivative as a tensor and taking its covariant derivative, as done with the Schwarzschild case:

$$L = \overbrace{\partial_\nu\left(\nabla_\mu u^\nu\right)}^{L_1} - \overbrace{\Gamma^\lambda_{\nu\mu}\nabla_\lambda u^\nu}^{L_2} + \overbrace{\Gamma^\nu_{\nu\lambda}\nabla_\mu u^\lambda}^{L_3}. \tag{B.1.20}$$

With the light-like normalisation, there is now an additional aspect to be considered, that was absent when assuming u^μ to be time-like: as two components of u^μ are non vanishing, u^r and u^t, there will be two separate equations of motion stemming from the Lagrangian. The clearest approach is to write down each expression separately, starting from $\mu = r$:

$$L_1 = \partial_r\partial_\nu u^\nu + \partial_\nu\left(\Gamma^\nu_{r\alpha}u^\alpha\right) \xrightarrow{\nu=\alpha=r} \partial_r^2 u^r + \partial_r\left(\Gamma^r_{\mu r}u^r\right) \tag{B.1.21}$$

$$L_2 = \Gamma^\lambda_{\nu r}\left(\partial_\lambda u^\nu + \Gamma^\nu_{\lambda\alpha}u^\alpha\right) \xrightarrow{\lambda=r} \Gamma^r_{rr}\partial_r u^r + \left(\left(\Gamma^t_{tr}\right)^2 + \left(\Gamma^r_{rr}\right)^2 + 2\left(\Gamma^\theta_{\theta r}\right)^2\right)u^r \tag{B.1.22}$$

$$L_3 = \Gamma^\nu_{\nu\lambda}\left(\partial_r u^\lambda + \Gamma^\lambda_{r\alpha}u^\alpha\right) \xrightarrow{\lambda=r} \left(\Gamma^t_{tr} + \Gamma^r_{rr} + 2\Gamma^\theta_{\theta r}\right)\left(\partial_r u^r + \Gamma^r_{rr}u^r\right). \tag{B.1.23}$$

Moving onto the next term, and again considering the first covariant derivative as a 2^{nd} rank tensor, and adjusting the indices accordingly:

$$M = g_{\mu\alpha}g^{\nu\beta}\nabla_\nu\left(\nabla_\beta u^\alpha\right) = g_{\mu\alpha}g^{\nu\beta}\left[\overbrace{\partial_\nu\left(\nabla_\beta u^\alpha\right)}^{M_1} + \overbrace{\Gamma^\alpha_{\nu\lambda}\nabla_\beta u^\lambda}^{M_2} - \overbrace{\Gamma^\lambda_{\nu\beta}\nabla_\lambda u^\alpha}^{M_3}\right] \tag{B.1.24}$$

$$g_{\mu\alpha}g^{\nu\beta}M_1 = g_{\mu\alpha}g^{\nu\beta}\partial_\nu\left(\partial_\beta u^\alpha + \Gamma^\alpha_{\beta\lambda}u^\lambda\right) \xrightarrow{\beta=\nu=r} \partial_r^2 u^r + \partial_r\left(\Gamma^r_{rr}u^r\right) \tag{B.1.25}$$

$$g_{\mu\alpha}g^{\nu\beta}M_2 = g_{\mu\alpha}g^{\nu\beta}\Gamma^\alpha_{\nu\lambda}\left(\partial_\beta u^\lambda + \Gamma^\lambda_{\beta\kappa}u^\kappa\right) \xrightarrow{\lambda=r} \Gamma^r_{rr}\left(\partial_r + \Gamma^r_{rr}\right)u^r + g_{rr}g^{tt}\Gamma^r_{tt}\Gamma^t_{tr}u^r \tag{B.1.26}$$

$$g_{\mu\alpha}g^{\nu\beta}M_3 = g_{\mu\alpha}g^{\nu\beta}\left(\Gamma^\lambda_{\nu\beta}\partial_\lambda u^\alpha + \Gamma^\lambda_{\nu\beta}\Gamma^\alpha_{\lambda\kappa}u^\kappa\right) \xrightarrow{\lambda=\kappa=r} \tag{B.1.27}$$

$$g_{rr}\left[2g^{\varphi\varphi}\Gamma^r_{\varphi\varphi} + \left(g^{rr} - g^{tt}\right)\Gamma^r_{rr}\right]\left(\partial_r u^r + \Gamma^r_{rr}u^r\right), \tag{B.1.28}$$

where in the computation of M_3 it was used that $\Gamma^r_{rr} = -\Gamma^r_{tt}$.

Appendix C

The Newtonian Correspondence in Flat Spacetime

C.1. The MOND Correspondence in Flat Space-Time

At this point, one could make a guess on the form of the Lagrangian, choosing coefficients based on assumptions such as the conservation of the Energy-Momentum tensor in a specific curved background (e.g. de Sitter space as in [26]). Instead, another route is chosen here. The only point of contact of Hossenfelder's formulation of CEG with observational data is given by the retrieval of the MOND equation in a flat spacetime background. As this is a crucial result, it is first and foremost necessary to demonstrate that the formulation with a light-like field proposed here can reproduce the same MOND PDE. On the other hand, as there is no rest frame for a light-like four vector, different considerations have to be made:

1. As shown in section 3.4, the conservation of angular momentum allows one to limit the analysis of the geodesic equations to the $\theta = \frac{\pi}{2}$ plane. The same can then be done for the E.O.M.s, since the case considered is that of a single test particle acted on by the imposter field source by a point mass at the origin of the coordinate system.

2. The coordinate system can be aligned so that the vector connecting the point mass to the test particle coincides with one of the Cartesian axes. Then, there is only one non-zero spatial component, which can be identified with the r component in spherical coordinates. When enforcing the normalisation from [21] on the spatial part in a Minkowski background, this leads to:

   $$u^\mu = \phi L \left(1, 1, 0, 0\right). \tag{C.1.1}$$

   This also gives a way to check the consistency of the model: as one has $u^t = u^r$, the equations of motion for the r and the t component should coincide.

3. In section 3.4, the similarity between the spacetime metric from [21] and the Schwarzschild metric the 1^{st} order in ϕ was noted. In fact, to 1^{st} order in ϕ, the Newtonian contributions to timelike orbits exactly coincided, and the general relativistic contribution

only differed by a factor of 2. It should hence be sufficient to consider terms $\mathcal{O}(\phi)$ for the E.O.M.s.

4. As previously explained, the only way to fix coefficients so to produce a theory that can explain observations is to keep the coefficients explicit in the derivation and only assign them once the E.O.M.s have been calculated, in order to retrieve MOND.

To obtain the E.O.M.s in Minkowski space-time in spherical coordinates, it is useful to note that all Christoffel symbols that are non-vanishing are also non-vanishing in Verlinde's metric. One can hence simply reuse the expressions worked out in section B.1, as in obtaining these the Christoffel symbols were not written out explicitly. Hence, by using equations B.1.5-B.1.28 and directly substituting $u^r = u^t = \phi L$, one has:

$$\nabla_\mu u^\mu = -L\partial_r\phi, \quad \left(\nabla_\mu u^\mu\right)^2 = \left(L\partial_r\phi\right)^2, \quad \left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right) = \left(L\partial_r\phi\right)^2, \tag{C.1.2}$$

$$\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) = \frac{2}{r^2}\left(L\phi\right)^2 \approx 0. \tag{C.1.3}$$

These terms appear identically in both the u^r and the u^t E.O.M.s as they have no free indices. The remaining terms are different for the t and r components. The t component is treated first. For $\mu = t$:

$$\nabla_\nu \epsilon_\mu^\nu = \frac{L}{2}\partial_r^2\phi, \quad \nabla_\nu\left(\epsilon\delta_\mu^\nu\right) = 0, \quad \epsilon_\mu^\nu + \epsilon\delta_\mu^\nu = -L\partial_r\phi \tag{C.1.4}$$

It is now possible to write the E.O.M.s without specifying the coefficients, with the Lagrangian for the free field:

$$L = \chi^{\frac{3}{2}}, \quad \chi = \alpha\epsilon_{\mu\nu}\epsilon^{\mu\nu} + \beta\epsilon, \tag{C.1.5}$$

where the antisymmetric combination was neglected due to the physical quantity of interest being a symmetric tensor, as previously explained for 2.1.13. The E.O.M.s hence read:

$$\nabla_\nu\left(\frac{\partial\chi^{\frac{3}{2}}}{\partial\left(\nabla_\nu u^t\right)}\right) = \tag{C.1.6}$$

$$3\left[\partial_r\sqrt{\frac{\alpha}{2}\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) + \frac{\alpha}{2}\left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right) + \beta\left(\nabla_\mu u^\mu\right)^2}\left(\alpha\epsilon_t^r + \beta\epsilon\right) + \right. \tag{C.1.7}$$

$$\left.\sqrt{\frac{\alpha}{2}\left(\nabla_\mu u_\nu\right)\left(\nabla^\mu u^\nu\right) + \frac{\alpha}{2}\left(\nabla_\mu u_\nu\right)\left(\nabla^\nu u^\mu\right) + \beta\left(\nabla_\mu u^\mu\right)^2}\nabla_\nu\left(\alpha\epsilon_r^\nu + \beta\epsilon\delta_r^\nu\right)\right] \approx \tag{C.1.8}$$

$$3L\left[\partial_r\sqrt{\frac{\alpha}{2}\left(\partial_r\phi\right)^2 + \beta\left(\partial_r\phi\right)^2}\left(-\alpha\partial_r\phi\right) + \sqrt{\frac{\alpha}{2}\left(\partial_r\phi\right)^2 + \beta\left(\partial_r\phi\right)^2}\left(\frac{1}{2}\alpha\partial_r^2\phi\right)\right] = \tag{C.1.9}$$

$$3\left[\sqrt{\frac{\alpha}{2} + \beta}\left(-\frac{|\partial_r\phi|}{\partial_r\phi}\partial_r\phi\partial_r^2\phi\alpha\right) + \sqrt{\frac{\alpha}{2} + \beta}\frac{\alpha}{2}\partial_r^2\phi\right]. \tag{C.1.10}$$

It is now convenient to look back at 2.4.92 to be able to compare the MOND equation with spherical symmetry. Under these conditions, it becomes:

$$\nabla \cdot (|\nabla \phi| \, \nabla \phi) \to 2 \left| \frac{\partial u^0}{\partial r} \right| \left(\frac{\partial^2 u^0}{\partial r^2} + \frac{1}{r} \frac{\partial u^0}{\partial r} \right). \tag{C.1.11}$$

One can see that by re-writing C.1.10 as

$$3\sqrt{\frac{\alpha}{2} + \beta} \left[\partial_r^2 \phi \, (-\alpha) + \frac{\alpha}{2} \partial_r^2 \phi \right], \tag{C.1.12}$$

noting that for the Newtonian potential the following holds,

$$\phi = -\frac{GM}{r} \to \partial_r^2 \phi = -2 \frac{1}{r} \partial_r \phi, \tag{C.1.13}$$

and substituting the relation in the second term of C.1.12, one can finally obtain:

$$-3 \, |\partial_r \phi| \, \alpha \sqrt{\frac{\alpha}{2} + \beta} \left(\partial_r^2 \phi + \frac{1}{r} \partial_r \phi \right), \tag{C.1.14}$$

which is in fact the L.H.S. of the MOND equation up to a constant factor given by the α and β coefficients. Carrying out a similar calculation for u^r yields:

$$\nabla_\nu \left(\frac{\partial \chi^{\frac{3}{2}}}{\partial \left(\nabla_\nu u^r \right)} \right) = 3 \, |\partial_r \phi| \left[\sqrt{\frac{\alpha}{2} + \beta} \, (\alpha - \beta) \left(-2 \frac{1}{r} \partial_r \phi \right) + \sqrt{\frac{\alpha}{2} + \beta} \left(-\frac{\alpha}{2} - \beta \right) \partial_r^2 \phi \right]. \tag{C.1.15}$$

For the coefficients to be equal, and for the two equations to be equivalent (which is needed given that $u^r = u^t$ one needs:

$$-2\sqrt{\frac{\alpha}{2} + \beta} = -\sqrt{\frac{\alpha}{2} + \beta} \left(\frac{\alpha}{2} + \beta \right) \to \beta = \pm \frac{\alpha}{2}. \tag{C.1.16}$$

This relation can hence be used to fix the coefficients in the kinetic term of the Lagrangian so to achieve MOND in a flat background.

Appendix D

The Ritz Method for FEM

The Ritz method proceeds as follows (see e.g. [81]):

1. Verify the operator[1] used in the PDE satisfies the following requirements: linearity, self-adjointness(symmetry), and positiveness. These requirements are tested by putting the PDE in operator form, such as:

   $$L\left(u\left(\vec{r}\right)\right)=f\left(\vec{r}\right), \tag{D.0.1}$$

   where L is the differential operator used, such as the gradient, divergence or Laplacian.

2. The PDE is converted into a minimisation problem.

3. From the minimisation problem a weak form is obtained.

4. The linear system of Element Matrix and Element Vector are obtained from the weak form.

As the MOND PDE is nonlinear, the Ritz method cannot be used. To formally establish that this is indeed the case, the linearity of a (differential) operator is introduced as follows:

$$L(\alpha u+\beta v)=\alpha L\left(u\right)+\beta L\left(v\right)\,\forall\left\{\alpha,\beta\right\},\,\forall\left\{u,v\right\}\in\Sigma. \tag{D.0.2}$$

Here, α and β are arbitrary constants and u and v are functions living in the function space Σ. As the non-linearity comes from the $\frac{|\nabla\phi|}{a_0}\gg 1$ case, it is sufficient to falsify the linearity for this limit, as it is known that the Newton-Poisson equation to which eq.1.4.20 reduces when $\frac{|\nabla\phi|}{a_0}\gg 1$ is linear. The RHS of (D.0.2) gives:

$$L\left(\alpha u+\beta v\right)=\nabla\cdot\left[\frac{|\alpha\nabla u+\beta\nabla v|}{a_0}\left(\alpha\nabla u+\beta\nabla v\right)\right]= \tag{D.0.3}$$

$$\alpha\nabla\cdot\left(\frac{|\alpha\nabla u+\beta\nabla v|}{a_0}\nabla u\right)+\beta\nabla\cdot\left(\frac{|\alpha\nabla u+\beta\nabla v|}{a_0}\nabla v\right). \tag{D.0.4}$$

[1] By operator one means the differential operator used in the PDE. As examples of operators in known PDEs, in the Poisson equation, $\nabla^2 u=f$, the operator is ∇^2.

On the other hand, for the RHS of (D.0.2) one obtains:

$$\tag{D.0.5}$$

$$\alpha L\left(u\right)+\beta L\left(v\right)=\alpha\nabla\cdot\left(\frac{|\nabla u|}{a_0}\nabla u\right)+\beta\nabla\cdot\left(\frac{|\nabla v|}{a_0}\nabla v\right), \tag{D.0.6}$$

$$L\left(\alpha u+\beta v\right)\neq\alpha L\left(u\right)+\beta L\left(v\right). \tag{D.0.7}$$

This shows that eq.1.4.23 is non-linear, and the Ritz method can hence not be utilised.

Bibliography

[1] S. W. Hawking, “Is the end in sight for theoretical physics?,” *Physics Bulletin*, vol. 32, no. 1, p. 15, 1981.

[2] R. Sanders, “Anti-gravity and galaxy rotation curves,” *Astronomy and Astrophysics*, vol. 136, pp. L21–L23, 1984.

[3] A. O. Hodson and H. Zhao, “Generalizing mond to explain the missing mass in galaxy clusters,” *Astronomy & Astrophysics*, vol. 598, p. A127, 2017.

[4] P. J. E. Peebles and B. Ratra, “The cosmological constant and dark energy,” *Reviews of modern physics*, vol. 75, no. 2, p. 559, 2003.

[5] R. Haberman, *Applied partial differential equations.* 2003.

[6] S. M. Carroll, *Spacetime and geometry.* Cambridge University Press, 2019.

[7] M. Sereno and P. Jetzer, “Dark matter versus modifications of the gravitational inverse-square law: results from planetary motion in the solar system,” *Monthly Notices of the Royal Astronomical Society*, vol. 371, no. 2, pp. 626–632, 2006.

[8] V. C. Rubin and W. K. Ford Jr, “Rotation of the andromeda nebula from a spectroscopic survey of emission regions,” *The Astrophysical Journal*, vol. 159, p. 379, 1970.

[9] E. Corbelli and P. Salucci, “The extended rotation curve and the dark matter halo of m33,” *Monthly Notices of the Royal Astronomical Society*, vol. 311, no. 2, pp. 441–447, 2000.

[10] J. F. Navarro, C. S. Frenk, and S. D. White, “A universal density profile from hierarchical clustering,” *The Astrophysical Journal*, vol. 490, no. 2, p. 493, 1997.

[11] M. Milgrom, “A modification of the newtonian dynamics as a possible alternative to the hidden mass hypothesis,” *The Astrophysical Journal*, vol. 270, pp. 365–370, 1983.

[12] J. Bekenstein and M. Milgrom, "Does the missing mass problem signal the breakdown of newtonian gravity?," *The Astrophysical Journal*, vol. 286, pp. 7–14, 1984.

[13] M. Milgrom, "Solutions for the modified newtonian dynamics field equation," *The Astrophysical Journal*, vol. 302, pp. 617–625, 1986.

[14] R. B. Tully and J. R. Fisher, "A new method of determining distances to galaxies," *Astronomy and Astrophysics*, vol. 54, pp. 661–673, 1977.

[15] S. S. McGaugh, J. M. Schombert, G. D. Bothun, and W. De Blok, "The baryonic tully-fisher relation," *The Astrophysical Journal Letters*, vol. 533, no. 2, p. L99, 2000.

[16] S. S. McGaugh, "The baryonic tully-fisher relation of galaxies with extended rotation curves and the stellar mass of rotating galaxies," *The Astrophysical Journal*, vol. 632, no. 2, p. 859, 2005.

[17] S. L. Jaki, "Johann georg von soldner and the gravitational bending of light, with an english translation of his essay on it published in 1801," *Foundations of Physics*, vol. 8, no. 11-12, pp. 927–950, 1978.

[18] J. D. Bekenstein, "Black holes and entropy," *Physical Review D*, vol. 7, no. 8, p. 2333, 1973.

[19] T. Jacobson, "Thermodynamics of spacetime: the einstein equation of state," *Physical Review Letters*, vol. 75, no. 7, p. 1260, 1995.

[20] E. Verlinde, "On the origin of gravity and the laws of newton," *Journal of High Energy Physics*, vol. 2011, no. 4, pp. 1–27, 2011.

[21] E. P. Verlinde, "Emergent gravity and the dark universe," *SciPost Phys*, vol. 2, no. 3, p. 016, 2017.

[22] M. Milgrom, "Quasi-linear formulation of mond," *Monthly Notices of the Royal Astronomical Society*, vol. 403, no. 2, pp. 886–895, 2010.

[23] R. Takahashi and T. Chiba, "Weak lensing of galaxy clusters in modified newtonian dynamics," *The Astrophysical Journal*, vol. 671, no. 1, p. 45, 2007.

[24] M. Milgrom and R. H. Sanders, "Rings and shells of "dark matter" as mond artifacts," *The Astrophysical Journal*, vol. 678, no. 1, p. 131, 2008.

[25] H. Katz, S. McGaugh, P. Teuben, and G. Angus, "Galaxy cluster bulk flows and collision velocities in qumond," *The Astrophysical Journal*, vol. 772, no. 1, p. 10, 2013.

[26] S. Hossenfelder, “Covariant version of verlinde’s emergent gravity,” *Physical Review D*, vol. 95, no. 12, p. 124018, 2017.

[27] L. D. Landau, *The classical theory of fields*, vol. 2. Elsevier, 2013.

[28] Y.-K. Lim and Q.-h. Wang, “Field equations and particle motion in covariant emergent gravity,” *Physical Review D*, vol. 98, no. 12, p. 124029, 2018.

[29] Y.-K. Lim and Q.-h. Wang, “Gravitational lensing in conformal and emergent gravity,” in *EPJ Web of Conferences*, vol. 206, p. 07002, EDP Sciences, 2019.

[30] M. Peskin, *An introduction to quantum field theory.* CRC press, 2018.

[31] D.-C. Dai and D. Stojkovic, “Comment on “covariant version of verlinde’s emergent gravity”,” *Physical Review D*, vol. 96, no. 10, p. 108501, 2017.

[32] T. Müller and F. Grave, “Catalogue of spacetimes,” *arXiv preprint arXiv:0904.4184*, 2009.

[33] H. Wang, F.-W. Zhang, Y.-Z. Wang, Z.-Q. Shen, Y.-F. Liang, X. Li, N.-H. Liao, Z.-P. Jin, Q. Yuan, Y.-C. Zou, *et al.*, “The gw170817/grb 170817a/at 2017gfo association: some implications for physics and astrophysics,” *The Astrophysical Journal Letters*, vol. 851, no. 1, p. L18, 2017.

[34] J.-J. Wei, B.-B. Zhang, X.-F. Wu, H. Gao, P. Mészáros, B. Zhang, Z.-G. Dai, S.-N. Zhang, and Z.-H. Zhu, “Multimessenger tests of the weak equivalence principle from gw170817 and its electromagnetic counterparts,” *Journal of Cosmology and Astroparticle Physics*, vol. 2017, no. 11, p. 035, 2017.

[35] S. Boran, S. Desai, E. Kahya, and R. Woodard, “Gw170817 falsifies dark matter emulators,” *Physical Review D*, vol. 97, no. 4, p. 041501, 2018.

[36] I. I. Shapiro, “Fourth test of general relativity,” *Physical Review Letters*, vol. 13, no. 26, p. 789, 1964.

[37] J. M. Ezquiaga and M. Zumalacárregui, “Dark energy after gw170817: dead ends and the road ahead,” *Physical review letters*, vol. 119, no. 25, p. 251304, 2017.

[38] D. Radice, A. Perego, F. Zappa, and S. Bernuzzi, “Gw170817: joint constraint on the neutron star equation of state from multimessenger observations,” *The Astrophysical Journal Letters*, vol. 852, no. 2, p. L29, 2018.

[39] J. D. Bekenstein, “Relativistic gravitation theory for the modified newtonian dynamics paradigm,” *Physical Review D*, vol. 70, no. 8, p. 083509, 2004.

[40] T. Zlosnik, P. Ferreira, and G. D. Starkman, “Vector-tensor nature of bekenstein’s relativistic theory of modified gravity,” *Physical Review D*, vol. 74, no. 4, p. 044037, 2006.

[41] M. Gasperini, “Singularity prevention and broken lorentz symmetry,” *Classical and Quantum Gravity*, vol. 4, no. 2, p. 485, 1987.

[42] T. Jacobson and D. Mattingly, “Gravity with a dynamical preferred frame,” *Physical Review D*, vol. 64, no. 2, p. 024028, 2001.

[43] R. S. Shankland, “Michelson-morley experiment,” *American Journal of Physics*, vol. 32, no. 1, pp. 16–35, 1964.

[44] C. Eling, T. Jacobson, and D. Mattingly, “Einstein-aether theory,” in *Deserfest*, pp. 163–179, World Scientific, 2006.

[45] C. M. Will, *Theory and experiment in gravitational physics.* Cambridge university press, 2018.

[46] T. G. Zlosnik, P. G. Ferreira, and G. D. Starkman, “Modifying gravity with the aether: An alternative to dark matter,” *Physical Review D*, vol. 75, no. 4, p. 044017, 2007.

[47] T. Baker, E. Bellini, P. G. Ferreira, M. Lagos, J. Noller, and I. Sawicki, “Strong constraints on cosmological gravity from gw170817 and grb 170817a,” *Physical review letters*, vol. 119, no. 25, p. 251301, 2017.

[48] S. Dodelson, *Modern cosmology.* Elsevier, 2003.

[49] R. H. Sanders and S. S. McGaugh, “Modified newtonian dynamics as an alternative to dark matter,” *Annual Review of Astronomy and Astrophysics*, vol. 40, no. 1, pp. 263–317, 2002.

[50] M. Milgrom, “Can the hidden mass be negative?,” *The Astrophysical Journal*, vol. 306, pp. 9–15, 1986.

[51] T. H. Reiprich and H. Boehringer, “The mass function of an x-ray flux-limited sample of galaxy clusters,” *The Astrophysical Journal*, vol. 567, no. 2, p. 716, 2002.

[52] J. Brownstein and J. Moffat, “Galaxy cluster masses without non-baryonic dark matter,” *Monthly Notices of the Royal Astronomical Society*, vol. 367, no. 2, pp. 527–540, 2006.

[53] G. O. Abell, H. G. Corwin Jr, and R. P. Olowin, “A catalog of rich clusters of galaxies,” *The Astrophysical Journal Supplement Series*, vol. 70, pp. 1–138, 1989.

[54] M. Meneghetti, G. Davoli, P. Bergamini, P. Rosati, P. Natarajan, C. Giocoli, G. B. Caminha, R. B. Metcalf, E. Rasia, S. Borgani, *et al.*, “An excess of small-scale gravitational lenses observed in galaxy clusters,” *Science*, vol. 369, no. 6509, pp. 1347–1351, 2020.

[55] R. Brada and M. Milgrom, "Exact solutions and approximations of mond fields of disc galaxies," *Monthly Notices of the Royal Astronomical Society*, vol. 276, no. 2, pp. 453–459, 1995.

[56] B. Qin, X. Wu, and Z. Zou, "An attempt to empirically evaluate the gravitational deflection of light in the modified newtonian dynamics.," *Astronomy and Astrophysics*, vol. 296, p. 264, 1995.

[57] V. Parthasarathy, C. Graichen, and A. Hathaway, "A comparison of tetrahedron quality measures," *Finite Elements in Analysis and Design*, vol. 15, no. 3, pp. 255–261, 1994.

[58] P. Su and R. L. S. Drysdale, "A comparison of sequential delaunay triangulation algorithms," *Computational Geometry*, vol. 7, no. 5-6, pp. 361–385, 1997.

[59] V. Fuetterling, C. Lojewski, and F.-J. Pfreundt, "High-performance delaunay triangulation for many-core computers.," in *High Performance Graphics*, pp. 97–104, 2014.

[60] T.-T. Cao, A. Nanjappa, M. Gao, and T.-S. Tan, "A gpu accelerated algorithm for 3d delaunay triangulation," in *Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 47–54, 2014.

[61] J.-D. Boissonnat, O. Devillers, M. Teillaud, and M. Yvinec, "Triangulations in cgal," in *Proceedings of the sixteenth annual symposium on Computational geometry*, pp. 11–18, 2000.

[62] J.-C. Nédélec, "Mixed finite elements in 3," *Numerische Mathematik*, vol. 35, no. 3, pp. 315–341, 1980.

[63] R. C. Kirby, "Algorithm 839: Fiat, a new paradigm for computing finite element basis functions," *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, no. 4, pp. 502–516, 2004.

[64] R. C. Kirby, "Fiat: numerical construction of finite element basis functions," in *Automated solution of differential equations by the finite element method*, pp. 247–255, Springer, 2012.

[65] R. C. Kirby, "Optimizing fiat with level 3 blas," *ACM Transactions on Mathematical Software (TOMS)*, vol. 32, no. 2, pp. 223–235, 2006.

[66] N. Hanzlikova and E. R. Rodrigues, "A novel finite element method assembler for co-processors and accelerators," in *Proceedings of the 3rd Workshop on Irregular Applications: Architectures and Algorithms*, pp. 1–8, 2013.

[67] P. Gottschling and D. Lindbo, "Generic compressed sparse matrix insertion: algorithms and implementations in mtl4 and fenics," in *Proceedings of the 8th workshop on Parallel/High-Performance Object-Oriented Scientific Computing*, pp. 1–8, 2009.

[68] W. Logg, G. Wells, and F. Bengzon, "Dolfin user manual," 2010.

[69] T. Dupont, J. Hoffman, C. Johnson, R. C. Kirby, M. G. Larson, A. Logg, and L. R. Scott, *The fenics project.* Chalmers Finite Element Centre, Chalmers University of Technology, 2003.

[70] M. S. Alnæs, A. Logg, K.-A. Mardal, O. Skavhaug, and H. P. Langtangen, "Unified framework for finite element assembly," *International Journal of Computational Science and Engineering*, vol. 4, no. 4, pp. 231–244, 2009.

[71] G.-R. Liu and S. S. Quek, *The finite element method: a practical course.* Butterworth-Heinemann, 2013.

[72] D. L. Logan, *A first course in the finite element method.* Cengage Learning, 2011.

[73] G. Strang and G. J. Fix, "An analysis of the finite element method," 1973.

[74] A. Plaza and M.-C. Rivara, "Mesh refinement based on the 8-tetrahedra longest-edge partition.," in *IMR*, pp. 67–78, 2003.

[75] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals.* Elsevier, 2005.

[76] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The fenics project version 1.5," *Archive of Numerical Software*, vol. 3, no. 100, 2015.

[77] A. Logg, K.-A. Mardal, and G. Wells, *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84. Springer Science & Business Media, 2012.

[78] J. Reddy, *An introduction to the finite element method*, vol. 1221. McGraw-Hill New York, 2004.

[79] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proceedings of the April 18-20, 1967, spring joint computer conference*, pp. 483–485, 1967.

[80] J. L. Gustafson, "Reevaluating amdahl's law," *Communications of the ACM*, vol. 31, no. 5, pp. 532–533, 1988.

[81] A. Segal and F. Vermolen, *Numerical methods in scientific computing.* VSSD, 2008.