Weak Gravitational Lensing Analysis of a Synthetic Galaxy Cluster Using the SIS Model

Candidate Number: 265113
Supervisor: Robert E. Smith

Word Count: 6139
Last Updated: May 7, 2025

University of Sussex

Abstract

This project investigates the mass distribution of a synthetic lens distribution at redshift $z = 0.3$ with weak gravitational lensing, using statistical analysis of complex ellipticities from $180,000$ background galaxies at redshift $z = 1$. A Single Isothermal Sphere method was implemented to calculate the Einstein radius, velocity dispersion and mass enclosed by Einstein radius. Measurements were derived for the tangential shear which was fitted using a reduced chi-squared method, with residual clipping and relative error clipping applied to improve the fit. The analysis considered three signal types: pure signal (without noise), noise signal (representing intrinsic shape noise) and a combined noise + pure signal, the realistic measurements that would be observed if carried out on a true galaxy cluster. The results from the pure signal matched closely with theoretical predictions, while the addition of shape noise led to overestimations of physical quantities, highlighting the impact of uncertainty. Deviations at small angular separations suggested potential benefits of using a more complex model, such as the Navarro–Frenk–White (NFW) profile. The study demonstrates the effectiveness and limitations of using the SIS model for deriving physical properties of a galaxy cluster via weak gravitational lensing in a realistic situation.

Contents

List of Figures

Preface

Chapter 1 is a review of the background material necessary to understand weak gravitational lensing and mass modelling of galaxy clusters. The source material for the introduction were independently selected following my own literature survey, while the source material for Section 1.1 and Section 1.2 was suggested by my supervisor.

The synthetic data presented in Section 2.1 were provided by my supervisor. The analysis techniques described in Section 2 were developed by me with support from my supervisor. The coding implementations used to present the results in Chapter 3 and carry out the analysis in Chapter 4 were written independently by myself and are included in Appendix A.

The results presented in Section 3, to the best of our knowledge, are original.

Chapter 1

Introduction

Albert Einstein Published General Relativity, the theory of gravitation, in 1915. General Relativity provides a unified description of gravity as a geometric property of the curvature of space and time. With Einstein's field equations, we have a relation between the curvature of space-time and all forms of energy and momentum. In addition, general relativity provides the description that light travels in the straightest path through curved space-time, called null geodesics. Combining these ideas allows us to understand how light is deflected around massive objects like galaxies from the curvature of space-time, gravitational lensing. In the decades following Einstein's groundbreaking work, the implications of general relativity became the subject of significant scientific interest. Although when Einstein published his paper in 1936 exploring the lensing effects of a star, he concluded that gravitational lensing was unobservable with the technology of the time (Einstein, 1936) Fritz Zwicky in 1937 extended the concept of gravitational lensing to galaxy clusters (Zwicky, 1937).

Rather than focusing on lensing by stars within the Milky Way, Zwicky proposed that entire galaxies or clusters along the line of sight of distant sources could be used as a lens. Based on his mass estimates, he predicted that such strong lensing effects would produce image separations which could be observed with telescopes, going on to lay the foundation for studying gravitational lensing beyond the Milky Way. Zwicky's theory remained largely theoretical until the 1970s, when astronomers first observed strong lensing with the double quasar (Q0957+561) (Walsh, Carswell, and Weymann, 1979). They observed light from one quasar split into two images by a foreground galaxy and thus provided the first proof of gravitational lensing.

Gravitational lensing can be categorized into three distinct regimes: Strong Gravitational Lensing, where multiple images and Einstein arcs can be produced when lensing occurs with a very large mass; Microlensing Gravitational Lensing – When the light rays are bent by stellar masses causing changes in the luminosity of the source image; and Weak Gravitational Lensing, which is the focus of this study, where only subtle changes in the shear and magnification of the background galaxies images are produced. Although the distortions from weak gravitational lensing are too faint to be observed individually, they can be detected statistically by analysing a large population of galaxies.

While the effects of strong lensing are dramatic and observable, it was rare, requiring near-perfect alignment of the source, lens, and observer. In contrast, the effects of weak gravitational lensing were common but subtle. The first observational evidence of weak gravitational lensing was produced by Tyson et al., who demonstrated that the distortion of background galaxies appeared coherent due to the mass distribution along the line of sight (J. A. Tyson, Valdes, and Wenk, 1990). This showed that weak gravitational lensing could provide us with measurements of the distortion of light from distant galaxies, allowing us to make various predictions about the structure and geometry of the universe.

A key area where weak gravitational lensing has had a profound impact is in the study of dark matter and energy. The deflection angle depends only on the mass density distribution of all

the material along the line of sight. This allows for the study of the mass distribution of cosmic structures without requiring assumptions about their composition. This is particularly useful for the study of dark matter, which does not emit, absorb, or reflect light, making assumptions about its composition largely speculative.

In 2000, weak gravitational lensing from isolated galaxies or clusters provided us with the first detection of cosmic shear. Three independent teams — Bacon, Refregier, and Ellis (2000); Van Waerbeke et al. (2000); and Wittman et al. (2000) — reported statistically significant shape distortion in background galaxies caused by the accumulative gravitational effect of the large-scale structure of the universe, often referred to as the cosmic web, a web-like network of galaxies, galaxy clusters and dark matter.

In this report, we start with Section 1.1, which goes over the setup of weak gravitational lensing and some definitions of variables. Section 1.2 covers the fundamental equations needed to understand this project. In Chapter 2, we will consider the Singular Isothermal Sphere model (SIS) to study the properties of a Galaxy Cluster and provide example code of my implementations, which can be found in Appendix A. We then proceeded with Chapter 3 and Chapter 4, my results and discussion section. Then a conclusion of the project can be found in Chapter 5.

1.1 Lensing Setup and Geometry

In the gravitational lensing setup (1.1), a concentration of mass, close to the line of sight, is situated on the lens plane. We define the optical axis as the line of sight, with the origin at the point where we take our measurements. Two planes are then defined: the lens plane at an angular diameter distance of D_d, and the source plane at an angular diameter distance D_s, with the distance between the two planes D_{ds}.

The source plane is where the many source galaxies, which produce the light that we conduct the analysis on, reside. We define the coordinates η_1 and η_2, which are the perpendicular distances from the optical axis describing the origin of each source galaxy on the source plane.

The lens plane is where weak gravitational lensing occurs, where the light that reaches the observer is deflected by the gravitational field of the mass distribution that we are studying. For the lens plane, we also define a 2-dimensional vector position ξ, with components ξ_1 and ξ_2, for the position of the galaxy on the lens plane from the optical axis.

With no other deflectors on or near the optical axis and the width of the mass profile along the optical axis is much smaller than the distances D_s and D_{ds}, we can approximate our model of the light rays, using the thin lens approximation, as a straight light rays which is deflected sharply at a single point on the lens plane rather than being smoothly curved as they pass the deflector, as it realistically behaves.

Lastly, three angles, β - the true angle to the source on the source plane, θ - the angle to the observed image on the lens plane, and $\hat{\alpha}$ - the deflection angle, are defined. Where the deflection angle is related to the mass distribution of the lens.

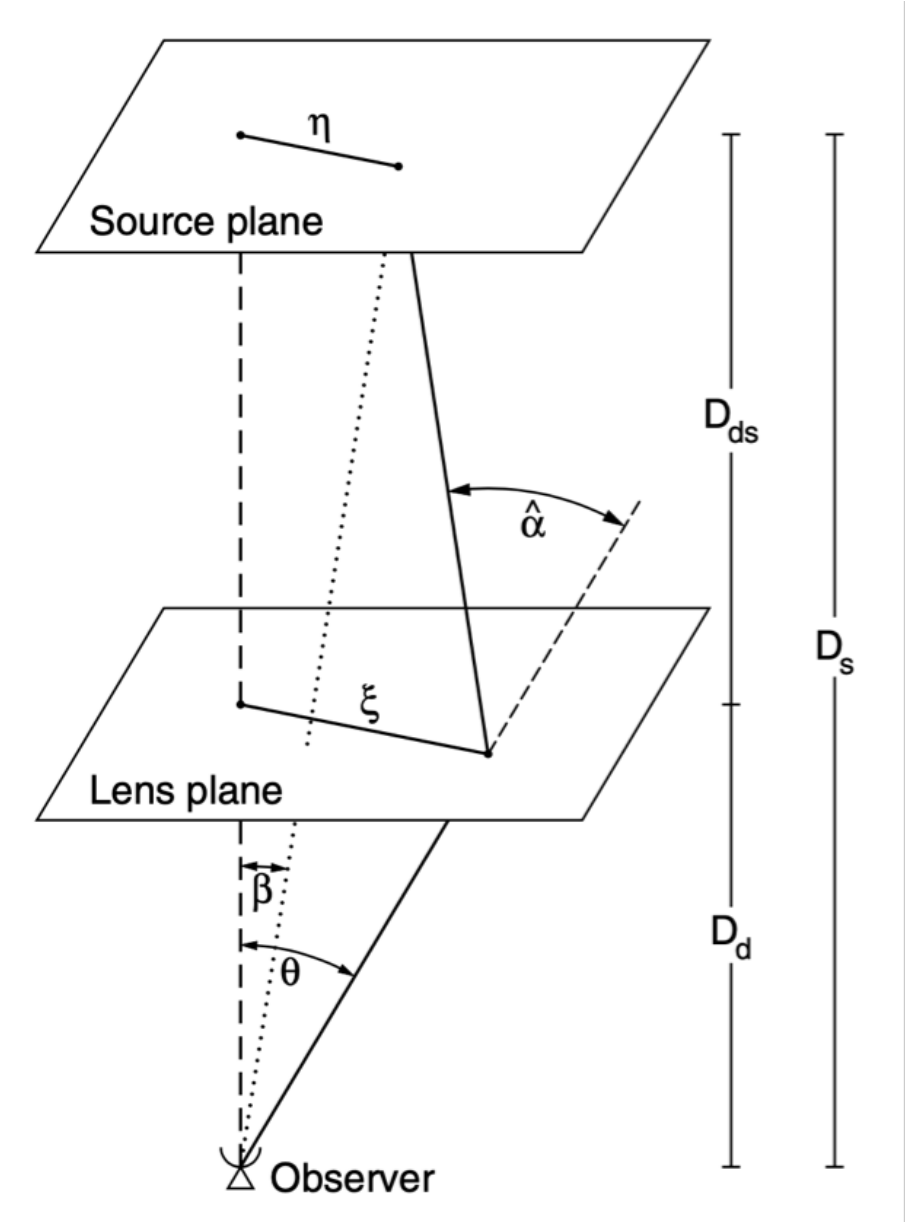

Figure 1.1: Diagram of gravitational lensing setup. A light ray from the a background source at position η on the source plane is incident on the lens plane at position ξ. The light ray is deflected by an angle of $\hat{\alpha}$ due to the mass distribution on the lens plane. The observer sees an image at angular position θ on the lens plane. The distance D_d, D_s, D_{ds} represent the diameter distance from the observer to the lens, to the source, and between the lens and source. Reproduced from Schneider (2006)

1.2 Fundamentals of Weak Gravitational Lensing

This section lays out the theoretical underpinnings of gravitational lensing, starting from the lens equation, covering the Jacobian matrix, which encapsulates the local lensing effects and the definition of the weak gravitational lensing regime, and leading to the definition of complex ellipticity.

Gravitational lensing refers to the deflection of light by mass, as predicted by general relativity. In the weak gravitational lensing regime, these deflections cause subtle shape distortions for background galaxies, which can be measured statistically and used to infer the distribution of matter in the universe.

1.2.1 Deflection Angle

Firstly, if we consider that the mass of the lens plane is a point mass where the optical axis meets the lens plane, then from Einstein's General Relativity we get that the prediction for the deflection angle is,

$$\hat{\alpha} = \frac{4GM}{c^2\xi}. \tag{1.1}$$

This expression holds if the distance $\xi << R_{Sch}$, where R_{Sch} is the Schwarzschild radius of the lens plane. In other words, the light ray subtends the lens plane outside the region around the mass that the light cannot escape. For a weak gravitational field, we can approximate the Einstein Field Equations as linear, allowing us to use the superposition principle. This allows us to say that the deflection angle for many point masses is given by,

$$\hat{\alpha} = \frac{4G}{c^2}\sum_{n=1}^{n} M_n \frac{\vec{\xi} - \vec{\xi}_n}{|\vec{\xi} - \vec{\xi}_n|^2}, \tag{1.2}$$

the sum of the resultant deflection angles from each point mass. In the limit of many point masses distributed continuously, we replace the discrete sum by an integral over the surface mass density $\Sigma(\vec{\xi\prime})$. This gives,

$$\hat{\alpha}(\vec{\xi\prime}) = \frac{4G}{c}\int_0^\infty \Sigma(\vec{\xi\prime}) \frac{\vec{\xi} - \vec{\xi\prime}}{|\vec{\xi} - \vec{\xi\prime}|^2}\, d^2\vec{\xi\prime}. \tag{1.3}$$

Where we have defined $\vec{\xi\prime}$, the 2-dimensional position vector that we integrate over, representing every possible position on the lens plane, and we have our angular diameter distances $\vec{\xi}$ and $\vec{\eta}$ covered in 1.1, we can write,

$$\vec{\theta} \approx \frac{\vec{\xi}}{D_d}, \qquad \vec{\beta} \approx \frac{\vec{\eta}}{D_s}, \tag{1.4}$$

given the small angle approximations. This allows us to write the surface mass density in terms of $\vec{\theta}$ instead of $\vec{\xi}$ as,

$$\Sigma(\vec{\theta}) = \frac{\Sigma(\vec{\xi})}{D_d}. \tag{1.5}$$

Converting our expression for $\hat{\alpha}(\vec{\xi\prime})$ to $\hat{\alpha}(\vec{\theta\prime})$ we get,

$$\hat{\alpha}(\vec{\theta\prime}) = \frac{4G}{c} D_d \int_0^\infty \Sigma(\vec{\theta\prime}) \frac{\vec{\theta} - \vec{\theta\prime}}{|\vec{\theta} - \vec{\theta\prime}|^2}\, d^2\vec{\theta\prime}, \tag{1.6}$$

for the deflection angle as a function of $\vec{\theta}$.

1.2.2 The Lens Equation and Reduced Deflection Angle

In the thin-lens approximation, all mass is projected onto a 2-dimensional lens plane perpendicular to the optical axis. The lens equation describes the transformation between $\vec{\beta}$, the true angle to source on the source plane, $\vec{\theta}$, the observed angle to the image on the lens plane, with $\hat{\alpha}$, the deflection angle

$$\vec{\beta} = \vec{\theta} - \vec{\alpha}(\vec{\theta}). \tag{1.7}$$

Here, we defined the reduced deflection angle $\vec{\alpha}$ which rescales the deflection angle normalising it by the distances between the observer lens and source and is related to $\hat{\alpha}$ by,

$$\vec{\alpha}(\vec{\theta}) = \frac{D_{ds}}{D_s}\hat{\alpha}(D_d\vec{\theta}). \tag{1.8}$$

1.2.3 Jacobi Matrix

The lens equation, which generally is non-linear, can be locally linearize if the source is small compared to the angular scale of which the lens properties vary and if the system lies within the weak gravitational lensing regime (covered in Section 1.2.4). This ensures the Jacobian matrix $\mathcal{A}$ is not near-singular. Under these condition, the lens equation is well approximated by the first-order Taylor expansion,

$$\vec{\beta} \approx \vec{\beta}(\vec{\theta_0}) + A(\vec{\theta_0})(\vec{\theta} - \vec{\theta_0}). \tag{1.9}$$

This allows us to express the mapping between $\vec{\beta}$ and $\vec{\alpha}$ in a small region $\vec{\theta_0}$ by the Jacobian matrix $\mathcal{A}$ given by,

$$\mathcal{A} = \frac{\partial\vec{\beta}}{\partial\vec{\theta}} = \begin{pmatrix} \frac{\partial\beta_1}{\partial\theta_1} & \frac{\partial\beta_1}{\partial\theta_2} \\ \frac{\partial\beta_2}{\partial\theta_1} & \frac{\partial\beta_2}{\partial\theta_2} \end{pmatrix}. \tag{1.10}$$

Using the second derivative of the gradient potential ψ, we express the Jacobian as,

$$\mathcal{A} = \begin{pmatrix} 1 - \frac{\partial^2\psi}{\partial\theta_1^2} & -\frac{\partial^2\psi}{\partial\theta_1\partial\theta_2} \\ -\frac{\partial^2\psi}{\partial\theta_1\partial\theta_2} & 1 - \frac{\partial^2\psi}{\partial\theta_2^2} \end{pmatrix}. \tag{1.11}$$

With this expression we can define two properties of the Jacobian matrix $\mathcal{A}$, which describes the full behaviour of the transformation from the source image to the lens image: the shear γ, which quantifies the anisotropic stretching of the source in the horizontal and diagonal directions and the convergence κ, which describes the uniform magnification of the source due to the projected mass density along the line of sight. These quantities directly related to the second derivatives of the lensing potential ψ as:

$$\kappa = \frac{1}{2}\left(\frac{\partial^2\psi}{\partial\theta_1^2} + \frac{\partial^2\psi}{\partial\theta_2^2}\right), \tag{1.12}$$

$$\gamma_1 = \frac{1}{2}\left(\frac{\partial^2\psi}{\partial\theta_1^2} - \frac{\partial^2\psi}{\partial\theta_2^2}\right), \tag{1.13}$$

$$\gamma_2 = \frac{\partial^2\psi}{\partial\theta_1\partial\theta_2}. \tag{1.14}$$

The terms, γ_1 and γ_2 are the two components of the complex shear $\gamma = \gamma_1 + i\gamma_2$. Given these expression we can re-express the Jacobian matrix $\mathcal{A}$ as:

$$\mathcal{A} = \begin{pmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{pmatrix}. \tag{1.15}$$

Where we can further simplify the Jacobian matrix $\mathcal{A}$ with the reduced shear g which is the shear observed from the galaxy shape distortions. The reduced shear includes the effects from both the shear and the convergence, and is defined as:

$$g = \frac{\gamma}{1-\kappa}. \tag{1.16}$$

This allows the expression of the Jacobian matrix as:

$$\mathcal{A} = (1-\kappa)\begin{pmatrix} 1-g_1 & -g_2 \\ -g_2 & 1+g_1 \end{pmatrix}. \tag{1.17}$$

One key property we can compute from our expression of the Jacobi matrix is the magnification of the image μ which is defined as:

$$\mu = det(\mathcal{A})^{-1}, \tag{1.18}$$

$$\mu = \frac{1}{(1-\kappa)^2 - |\gamma|^2}. \tag{1.19}$$

As opposed to the convergence κ which describes the uniform magnification of the image, μ includes both convergence and shear. Shear indirectly changes the area of the image and therefore contributes to the total magnification of the image.

1.2.4 Weak Gravitational Lensing Regime

Weak gravitational lensing occurs when the gravitational field of a foreground mass only slightly distorts the images of background sources. This ensures that the lens equation can be linearised, allowing for a one-to-one relationship between the true and observed angular distances $\vec{\beta}$ and $\vec{\theta}$. The lens equation can be linearised when the source is small compared to the angular scale over which the lens properties change, and if the Jacobian matrix $\mathcal{A}$ is not near singular. The Jacobian matrix is singular when the determinant is zero,

$$(1-\kappa)^2 - |\gamma|^2 = 0. \tag{1.20}$$

Assuming that the distortion is subtle $\gamma \ll 1$, we can define the limit for weak gravitational lensing as $\kappa \ll 1$. One observation from our magnification equation is that for $\kappa \to 1$ we get a diverging magnification, whereas within the weak gravitational lensing regime, the magnification $\mu \approx 1$. In the strong gravitational lensing regime, where $\kappa \geq 1$, the lens equation can no longer be linearised, allowing for multiple solutions for the deflection angle. This leads to the formation of multiple images on the lens plane.

1.2.5 Complex Ellipticity

In gravitational lensing, light is deflected but is not emitted or absorbed, meaning that the surface brightness is conserved. This is important because it allows us to define the shape of our galaxy in terms of the surface brightness, and any changes in the shape are due to lensing, not to a change in intrinsic light distribution. Therefore, we define the tensor of second brightness moment for the observed image of the galaxy in the sky,

$$Q_{ij} = \frac{\int d^2\theta\ I(\boldsymbol{\theta})\ q_I[I(\boldsymbol{\theta})]\ (\theta_i - \bar{\theta}_i)(\theta_j - \bar{\theta}_j)}{\int d^2\theta\ I(\boldsymbol{\theta})\ q_I[I(\boldsymbol{\theta})]}, \quad i,j \in \{1,2\}. \tag{1.21}$$

The tensor of second brightness moment describes the spatial distribution of light in a galaxy image, giving a mathematical description of the galaxy's shape. $q_I[I(\boldsymbol{\theta})]$ is an appropriately chosen cut-off function for the intensity, where the Heaviside step function is often used to

define the edge of the galaxy's image, and $\bar{\theta}_i$ is the centroid of the galaxy image. From this, we can then compute the complex ellipticity,

$$\epsilon \equiv \frac{Q_{11} - Q_{22} + 2iQ_{12}}{Q_{11} + Q_{22} + 2(Q_{11}Q_{22} - Q_{12}^2)^{1/2}}. \tag{1.22}$$

Some key properties of complex ellipticity are that $Re(\epsilon)$ describes the stretching along the horizontal and vertical axes, and $Im(\epsilon)$ measures the stretching along the diagonal axis at 45.

Chapter 2

Material and Methods

2.1 Data and Processing

The dataset used in the project included 3 categories of complex ellipticities for 108,000 synthetic background galaxies. The first category, refereed to as the pure signal, consisted of ellipticity measurements from gravitational lensing alone, without any shape noise; The second, termed the noise signal, included ellipticity measurements without any lensing applied only the random intrinsic shape noise; and the third, refereed the pure + noise signal, the combination of the ellipticity measurements for lensed background sources with shape noise, simulating realistic observational data. Our shape noise is given by the ellipticity dispersion σ_e, which is a measure of the intrinsic variation in galaxies' ellipticities. In our synthetic lensing data, the shape noise $\sigma_e = 0.16$.

2.2 SIS Model

For this report, I focused on analysing the mass distribution on the lens plane with the singular isothermal sphere (SIS) model widely used in weak gravitational lensing for galaxy clusters. The SIS model assumes that the mass distribution is spherically symmetric and the particles are randomly moving in all directions with an average speed that doesn't depend on the radial distance from the centre of the system, giving a constant velocity dispersion σ_v. This leads to the 3-dimensional mass density of a SIS model as,

$$\rho_{\mathrm{3D}}(\vec{r}) = \frac{\sigma_v^2}{2\pi G r^2}, \tag{2.1}$$

where r is the radial distance from the lens centre.

2.2.1 Surface Mass Density

I projected the 3D mass distribution along the optical axis r_3 to get the 2D mass distribution,

$$\Sigma(\vec{\xi}) = \int_{-\infty}^{\infty} \mathrm{d}r_3 \, \rho_{\mathrm{3D}}(\vec{\xi}, r_3) = \frac{\sigma_v^2}{2G\xi}, \tag{2.2}$$

where $\vec{r} = (\xi_1, \xi_2, r_3)$ is the radial vector from the centre of the galaxy cluster. Next, I converted the surface mass distribution into our angular distance coordinates with $\vec{\xi} = \vec{\theta} D_d$, giving:

$$\Sigma(\vec{\theta}) = \frac{\sigma_v^2}{2GD_d\theta}, \tag{2.3}$$

the surface mass distribution in terms of the velocity dispersion, distance to the lens, and θ

2.2.2 Convergence

In Section 1.2.3, convergence κ was introduced with the Jacobian matrix $\mathcal{A}$ where it was defined in terms of second derivatives of the lensing potential. While this is useful to understand local distortions mathematically, convergence can also be defined in terms of the surface mass density $\Sigma(\vec{\theta})$ and critical surface mass density $\Sigma_{crit}(\vec{\theta})$ of the lens as:

$$\kappa(\vec{\theta}) = \frac{\Sigma(\vec{\theta})}{\Sigma_{crit}(\vec{\theta})}, \tag{2.4}$$

$$\Sigma_{\text{crit}} = \frac{c^2}{4\pi G}\frac{D_s}{D_d D_{ds}}. \tag{2.5}$$

This defines convergence as the dimensionless surface mass density, with the same physical properties stated earlier. Substituting the surface mass density $\Sigma(\vec{\theta})$ and critical surface mass density $\Sigma_{crit}(\vec{\theta})$ and then simplifying with Einstein radius θ_E we get:

$$\kappa(\theta) = \frac{\sigma_v^2}{2GD_d\theta} \cdot \frac{4\pi G D_d D_{ds}}{c^2 D_s}, \tag{2.6}$$

$$\kappa(\theta) = \frac{\theta_E}{2\theta}. \tag{2.7}$$

Here, the Einstein radius θ_E is defined as the angular distance at which the light from the source galaxy is deflected equally in all directions. The result of this is that the light is focused into a ring, called an Einstein ring, centred around the lens centre with radius equal to the Einstein radius θ_E. This occurs at the boundary between weak gravitational lensing and strong gravitational lensing, when the source lies perfectly aligned with the centre of the lens and the observer. The Boundary between weak and strong gravitational lensing, as defined in Section 1.2.4, $\kappa = 1$, which implies for the SIS model that $\Sigma(\theta_E) = \Sigma_{\text{crit}}(\theta_E)$. Solving for θ_E, we get:

$$\theta_E = 4\pi \left(\frac{\sigma_v}{c}\right)^2 \frac{D_{ds}}{D_s}. \tag{2.8}$$

2.2.3 Relationship between Shear and Convergence

In the SIS model the relationship between the tangential shear γ_t and the convergence κ is given by,

$$\langle\gamma_t(\vec{\theta})\rangle_c = \bar{\kappa}(\vec{\theta}) - \langle\kappa(\vec{\theta})\rangle_c, \tag{2.9}$$

where $\bar{\kappa}(\vec{\theta})$ is the mean convergence inside a circular aperture of radius θ and $\langle\kappa(\vec{\theta})\rangle_c$ is the average convergence for a ring at angular separation θ. From the definitions of the convergence from the SIS model we find that

$$\langle\kappa(\vec{\theta})\rangle_c = \tfrac{\theta_E}{2\theta} = \kappa(\vec{\theta}), \quad \bar{\kappa}(\vec{\theta}) = \tfrac{\theta_E}{\theta}. \tag{2.10}$$

Therefore giving the relation between tangential shear γ_t and the convergence κ for the SIS model,

$$\langle\gamma_t(\vec{\theta})\rangle_c = \kappa(\vec{\theta}) = \frac{\theta_E}{2\theta}. \tag{2.11}$$

2.3 Shear Measurement

The shapes of galaxies are described with our complex ellipticity. However, individual galaxies have an intrinsic ellipticity due to their natural shape. This implies that,

$$\epsilon_{\text{obs}} \approx \epsilon_{\text{int}} + g, \tag{2.12}$$

the observed ellipticities of a galaxy ϵ_{obs}, are a combination of the intrinsic ellipticity ϵ_{int}, and the ellipticity from the effect of lensing g.

One key assumption that allows us to extract the shear from the complex ellipticity is that intrinsic ellipticities are randomly oriented. If it is assumed that the source galaxy's ellipticities are random and can be modelled by a Poisson distribution centred around 0, by averaging the observed ellipticity over a large sample of source galaxies, it is expected that the intrinsic ellipticity doesn't significantly contribute because $\langle \epsilon_{\text{int}} \rangle \approx 0$. This allows us, under the weak lensing limit, where $\kappa \ll 1$, to relate the mean observed ellipticity to the shear,

$$\langle \epsilon_{\text{obs}} \rangle \approx g \approx \frac{\gamma}{1-\kappa} \approx \gamma. \tag{2.13}$$

To extract the tangential shear and cross shear from the complex ellipticity we implement a coordinate transformation from Cartesian to Polar Coordinates relative to the lens centre. For each galaxy with position $\theta = (\theta_1, \theta_2)$, the polar angle is computed as,

$$\phi = \begin{cases} \cos^{-1}\left(\dfrac{\theta_1}{|\vec{\theta}|}\right) & (\theta_2 \geq 0) \\ -\cos^{-1}\left(\dfrac{\theta_1}{|\vec{\theta}|}\right) & (\theta_2 < 0) \end{cases} \tag{2.14}$$

The sign flip for the sign of θ_2 ensures that the angle spans the full $[0, 2\pi]$ range.

The complex shear is then decomposed into 2 components: it's tangential shear γ_t, the radial distortion around the lens centre, and cross shear γ_x, distortion at 45 to the radial direction from the lens centre,

$$\gamma_t(\vec{\theta}\,|\,\vec{\theta}_0) = -\text{Re}\left[\gamma(\vec{\theta}+\vec{\theta}_0)e^{-2i\phi}\right], \quad \gamma_\times(\vec{\theta}\,|\,\vec{\theta}_0) = -\text{Im}\left[\gamma(\vec{\theta}+\vec{\theta}_0)e^{-2i\phi}\right]. \tag{2.15}$$

This decomposition of the complex ellipticity is directly related to the Jacobi Matrix $\mathcal{A}$ introduced in Section 1.2.3. In theory, for an ideal axisymmetric lens, such as the lens described by the (SIS) model, the cross shear is expected to be zero. As such, cross shear provides a diagnostic for identifying asymmetries in the mass distribution. If a significant cross shear is measured, it may indicate that the lens is not truly axisymmetric and the (SIS) model is not appropriate.

2.4 Plotting and Model Fitting

After extracting the tangential shear from the complex ellipticity measurements, the next step is to analyse its radial dependence and fit the data with the SIS model with the idea of estimating the physical properties in the following section, Section 2.5. In this section, we cover the methods used to radially bin the shear measurements, calculate uncertainty, apply relative error, apply residual clipping, calculate the reduced chi-squared statistic, and perform model fitting using a reduced chi-squared fitting technique.

2.4.1 Radial Binning Method

To extract the radial dependence of the shear field, a radial binning method is used. This allows us to construct a radial shear profile that we can directly compare to lens models such as the SIS model. The tangential shear γ_t is measured as a function of θ, the radial distance from the lens centre $\vec{\theta}_0$. The radial binning method is used to average the measurements of γ_t over concentric annuli.

We divide the measurements of γ_t into a set of radial bins, each defined by an inner and outer radius. The bin sizes are logarithmically spaced so that the bins are smaller in width closer to

the centre of the lens. This provides bins that are evenly spaced when plotted on a logarithmic scale. The binning method can be written as,

$$\langle \gamma_t \rangle_c = \frac{\sum_{i=1}^{N} \mathbf{1}\left[\theta_i - \frac{\Delta\theta}{2}, \theta_i + \frac{\Delta\theta}{2}\right] \gamma_{t,i}}{\sum_{i=1}^{N} \mathbf{1}\left[\theta_i - \frac{\Delta\theta}{2}, \theta_i + \frac{\Delta\theta}{2}\right]}. \tag{2.16}$$

Within each bin, we compute the mean tangential shear $\langle \gamma_t \rangle$ by averaging γ_t as,

$$\langle \gamma_t \rangle \approx \frac{1}{N} \sum_{i \in \text{bins}} \gamma_t \tag{2.17}$$

The Python implementation of the radial binning algorithm is provided in Appendix A.4.

2.4.2 Uncertainty

In weak gravitational lensing, you have various sources of error. However, the most significant sources of error and the sources of error I have consider are the shape noise and the statistical errors from finite sampling. Implementations of the following equations can be found in Appendix A.5. For the three categories of data: pure signal is only subject to finite sampling noise σ_f, the noise signal is only subject to the shape noise σ_s and the pure + noise signal is subject to both the finite sampling noise and shape noise.

Shape noise, which arising from the intrinsic variation in galaxy ellipticities, is the dominant source of error in weak gravitational lensing. However because the intrinsic shape of galaxies are randomly orientated in the sky by averaging the shear over radial bins, with N galaxies, we reduce the shape noise as $\frac{1}{\sqrt{N}}$ and thus the calculated error for the shape noise σ_s, where σ_ϵ is the standard deviation of intrinsic galaxy ellipticities, is given by,

$$\sigma_s = \frac{\sigma_\epsilon}{\sqrt{N}} \tag{2.18}$$

In addition to shape noise, weak gravitational lensing measurements are subject to statistical uncertainty from finite sampling. Uncertainty from Finite Sampling occur because we are sampling a continuous shear field at discrete set of points. There are small fluctuations in our shear field across the bin which means our computed mean shear does not exactly represent the true mean shear of the bin. The uncertainty due to finite sample decreases with the number of galaxies in the bin N and I have computed it as,

$$\sigma_f = \frac{BinAverage}{\sqrt{N}}. \tag{2.19}$$

For the error on the pure + noise signal we used the standard error propagation formula for the addition of error,

$$\sigma_{pure+noise} = \sqrt{\sigma_{pure}^2 + \sigma_{noise}^2} \tag{2.20}$$

This is the realistic error that is expected in realistic observation of lensing data. Therefore the errors on each of my three sets of data:

$$\begin{aligned} \sigma_{pure} &= \tfrac{BinAverage}{\sqrt{N}}, \\ \sigma_{noise} &= \tfrac{\sigma_\epsilon}{\sqrt{N}}, \\ \sigma_{pure+noise} &= \sqrt{\sigma_{pure}^2 + \sigma_{noise}^2}. \end{aligned} \tag{2.21}$$

2.4.3 Relative Error Clipping

To improve the reliability of my fit, I applied relative error clipping. This is the technique of removing any points with a relative error of greater than a certain threshold, particularly important at small angular scales where the number of galaxies per bin is lower and shape noise is more significant. I chose a threshold of 0.8 which ensure that the signal to noise ratio $SNR > 1.25$. This value was chosen so that low-confidence measurements with a smaller signal to noise ratio than the commonly used $SNR = 2$ are still retained which is better for exploratory fitting however may add more noise to the fit. The code for the implementation of this can be found in Appendix A.6.

2.4.4 Residual Clipping

Another way I improved the reliability of my fit, I implemented a residual clipping procedure. I have removed data points with residuals exceeding 3σ from the initial fit commonly used in outlier rejection. This process reduced the influence of outliers which might be present as we probe the limits of theta that our model can describe. With high-residual points removed, the fitted profile should stabilise, leading to a more reliable measurement of θ_E. One consideration with residual clipping is that a measurements with a residual greater than 3σ might suggest that there is a physical property of the lens that it not being well described by the SIS model. The code for my relative error procedure can be found in the Appendix A.7.

2.4.5 Reduced Chi-Squared Statistic

The Reduced Chi-Squared statistic χ^2_ν provides a measure of the goodness of the fit for analysis. The method for this can be found in the Appendix A.10. Chi-Squared is defined as,

$$\chi^2 = \sum_i \frac{(y_i - y_{\text{model}}(x_i))^2}{\sigma_i^2}, \tag{2.22}$$

where y_i is the observed value, y_{model} is the models predicted value and σ_i is the error of each point. The Reduced Chi-Squared is the normalised Reduced Chi-Squared is then,

$$\chi^2_\nu = \frac{\chi^2}{\nu}, \tag{2.23}$$

where ν is the number of degree of freedom. A value of $\chi^2_\nu \approx 1$ indicates that the model provides a statistically acceptable fit for the data. Values much less or much greater than 1 suggest the the model is overfitting or underfitting the data.

2.4.6 Reduced Chi-squared Fitting Method

The Reduced Chi-squared Fitting method was used to fit our tangential shear measurements with the SIS model and the implementation can be found in Appendix A.3. The implemented Reduced Chi-square method can be summarised with the following steps.

- Performs Relative Error Clipping
- Fits the data
- Performs Residual Clipping
- Refits the data
- Compute the variance of free parameters
- Calculate the reduced chi-square statistic

- Plots the fit with radially binned tangential shear
- Plots Residuals

To fit the our data with the SIS model, we recall the relationship between the tangential shear ring average and the convergence, as described in Equation 2.11. The SIS model predicts a tangential shear γ_t that scales inversely with angular distance:

$$\gamma_t(\theta) = \frac{\theta_E}{2\theta}, \tag{2.24}$$

which when fitted in log-log space, becomes a linear relation:

$$\ln(\gamma_t) = m\ln(\theta) + \ln(\frac{\theta_E}{2}), \tag{2.25}$$

where $m = -1$. By performing a linear regression fit with in log-log space the was fit with the SIS model for the radially binned tangential shear we can assess how well the SIS model fits our data by comparing the slope of the fit to the SIS prediction.

2.5 Estimate of Physical Properties

Following the model fitting of the tangential shear profiles, we can extract the physical properties that describe the mass distribution of our synthetic galaxy cluster. In this section we outline the methods to calculate the Einstein radius, velocity dispersion and mass.

2.5.1 Einstein Radius

The Einstein radius, defined in Section 2.2.2, characterises the angular scale at which the effects of gravitational lensing maximally symmetric. From the shear profile obtained through our radial binning technique the Einstein radius σ_E was estimated by fitting the data with the SIS model. From our fit we have the linear relation in log-log space (previously introduced in Equation 2.25):

$$\ln(\gamma_t) = m\ln(\theta) + \ln(\frac{\theta_E}{2}), \tag{2.26}$$

where the y-intercept of the linear fit corresponds to $\ln(\frac{\theta_E}{2})$. This allows the Einstein radius to be extracted from our fit as:

$$\theta_E = 2 \cdot \mathrm{e}^{intercept}. \tag{2.27}$$

In the analysis, this computation was implemented as a function that takes the $y_{intercept}$ and its corresponding error $\sigma_{intercept}$. The code for this procedure can be found in Appendix A.14. The Error propagation was calculated accordingly, using standard propagation rules:

$$\sigma_{\theta_E} = 2 \cdot \mathrm{e}^{intercept} \cdot \sigma_{intercept}. \tag{2.28}$$

2.5.2 Velocity Dispersion

In the context of weak gravitational lensing, the velocity dispersion σ_v characterises the random motion of galaxies within the cluster. In the SIS model, the velocity dispersion is assumed to be constant given by the relation with the Einstein radius θ_E:

$$\theta_E = 4\pi(\frac{\sigma_v^2}{c})^2\frac{D_{ds}}{D_s}, \tag{2.29}$$

where c is the speed of light. By rearranging the equation we get that the velocity dispersion σ_v is,

$$\sigma_v = c(\frac{\theta_E D_s}{4\pi D_{ds}})^{1/2} \tag{2.30}$$

Following this the error on the velocity dispersion $\nabla\sigma_v$, where we use Δ to denote the error to avoid confusion, is given by:

$$\Delta\sigma_v = |\frac{\partial\sigma_v}{\partial\theta_E}| \cdot \sigma_v = \frac{\Delta\theta_E}{2\theta_E} \cdot \sigma_v. \tag{2.31}$$

The code for the procedure for measuring both the velocity dispersion and its error can be found in Appendix A.15

Predicted Velocity Dispersion

To quantify the correctness of our result for the velocity dispersion I calculated the circular velocity at orbital radius r_c for a typical galaxy cluster. Since the velocity dispersion is assumed constant with the SIS model, v_c provides an approximate result for σ_v. This is done by equating the centripetal force with the gravitational force:

$$\frac{mv_c^2}{r_c} \approx \frac{GMm}{r_c^2}. \tag{2.32}$$

where v_c is the circular velocity at orbital radius r_c and M is the total mass enclosed within radius r_c. This simplifies to,

$$v_c \approx \sqrt{\frac{GM}{r_c}}. \tag{2.33}$$

Substituting for the approximate values, provided by my supervisor, for the total mass of the galaxy cluster $M \approx 1.0 \times 10^{45} kg$ and $r_c = 3 \times 10^{22} m$ we get the expected result for the circular velocity of $v_c \approx 1.4 \times 10^6 ms^{-1}$. Therefore,we expect the velocity dispersion to be of the same order of magnitude.

2.5.3 Mass

Estimating the mass of the galaxy cluster is the main goal of weak gravitational analysis and the method for doing so along with the error calculation is given in Appendix A.16. The SIS model, which assume a spherically symmetric mass distribution with a constant velocity dispersion σ_v, provides a simple yet powerful framework for estimating the mass of the galaxy cluster on the our lens plane from our gravitational lensing signal.

The SIS model predicts a surface mass density, Equation 2.2.1, that decreases with increasing angular separation. By integrating this surface density over a circular region of the lens plane, we obtain the total mass enclosed within an angular radius:

$$M(<\theta_E) = \int_0^{2\pi}\int_0^{\theta_E} \Sigma(\theta) \cdot D_d^2\theta \, d\theta \, d\phi = \frac{\pi\sigma_v^2 D_d \theta_E}{G}. \tag{2.34}$$

We evaluate the mass enclosed within the Einstein radius θ_E where weak gravitational has perfect circular symmetry and the which makes the estimation for the mass particularly accurate.

I calculated the error on the mass estimate by propagating the uncertainties in the Einstein radius and the velocity dispersion using standard error propagation rules. The total error accounts for the fact that the mass depends quadratically on velocity dispersion and linearly on the Einstein radius θ_E and is given by:

$$\Delta M = M \cdot \sqrt{\left(2 \cdot \frac{\Delta\sigma_v}{\sigma_v}\right)^2 + \left(\frac{\Delta\theta_E}{\theta_E}\right)^2}. \tag{2.35}$$

Chapter 3

Results

In this section, we present the result of our weak gravitational lensing analysis of our synthetic lensing data at redshift $z = 0.3$. It contains three sets of ellipticity measurements (pure signal, noise, and pure + noise) of 108,000 distant galaxies observed on the lens plane. We calculated the tangential shear ring average, $\langle\gamma_t\rangle_c$, for our mass distribution on the lens plane for each set of data, then fit the Singular Isothermal Sphere (SIS) profile to each. From these fits fit, we calculated the significant physical properties for the mass distribution on the lens , such as; Einstein radius θ_E, velocity dispersion σ_v and the mass enclose by the Einstein radius.

3.0.1 Tangential Shear

The tangential shear ring average was calculated for 16 annular bins around the galaxy's centre, shown in Figure 3.1, Figure 3.2, and Figure 3.3. The bins span angular distances from $2.58 \times 10^{-5}\,rad$ to $1.23 \times 10^{-2} rad$ which correspond to distances $2.37 \times 10^{4} pc$ to $1.13 \times 10^{7} pc$. Figure 3.1 presents the pure signal tangential shear as a function of angular separation. The plot shows a clear trend, yielding a reduced $\chi^2 \approx 0.86$ with 9 degrees of freedom. The residuals for this plots are shown in Figure 3.4, which show a weak positive trend. Figure 3.2 shows the noise-only tangential shear signal, which reveals scattered points around zero across most radii. However, at low radii, a positive deviation is observed. Figure 3.3 combines the pure with the noise tangential shear signal, producing a realistic observation-like plot. It shows a detectable lensing signal with a reduced $\chi^2 \approx 0.36$ for 10 degrees of freedom. The corresponding residuals, which show no systematic trend, are presented in Figure 3.5.

3.0.2 Estimates of Physical Properties

For both the pure signal and pure + noise signal fit for the radially binned tangential shear, estimates of the physical properties can be seen in Table 3.1. These estimate were calculated using the methods outlined in Chapter 2.

Signal type	θ_E (10^{-5}rad)	σ_v $(10^5\,\text{ms}^{-1})$	$M(<\theta_E)$ $(10^{13}\,M_\odot)$
Pure	4.48 ± 0.29	8.50 ± 0.27	2.17 ± 0.20
Pure + Noise	9.08 ± 2.13	12.10 ± 1.42	8.92 ± 2.96

Table 3.1: Estimates of physical properties characterising the mass distribution from the SIS model fit for each signal type.

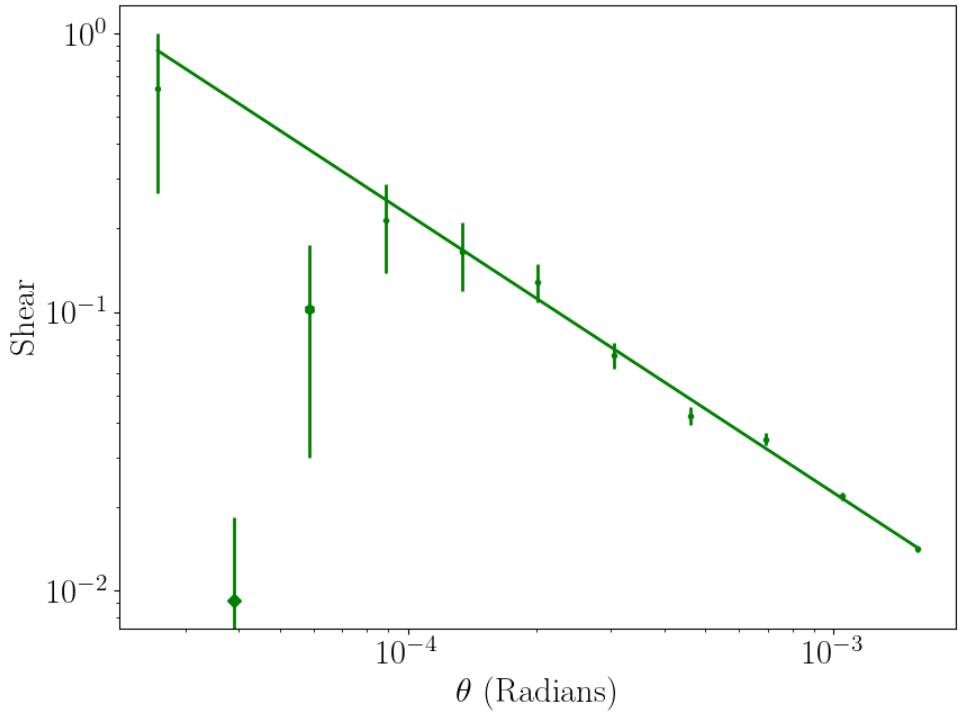

Figure 3.1: Tangential shear as a function of angular separation for the pure signal. The data points represent ring-averaged shear values across 16 annular bins, with vertical bars showing statistical uncertainties. The solid line corresponds to the best-fitting SIS model, which yields a reduced $\chi^2 \approx 0.86$ with 9 degrees of freedom, indicating a good fit.

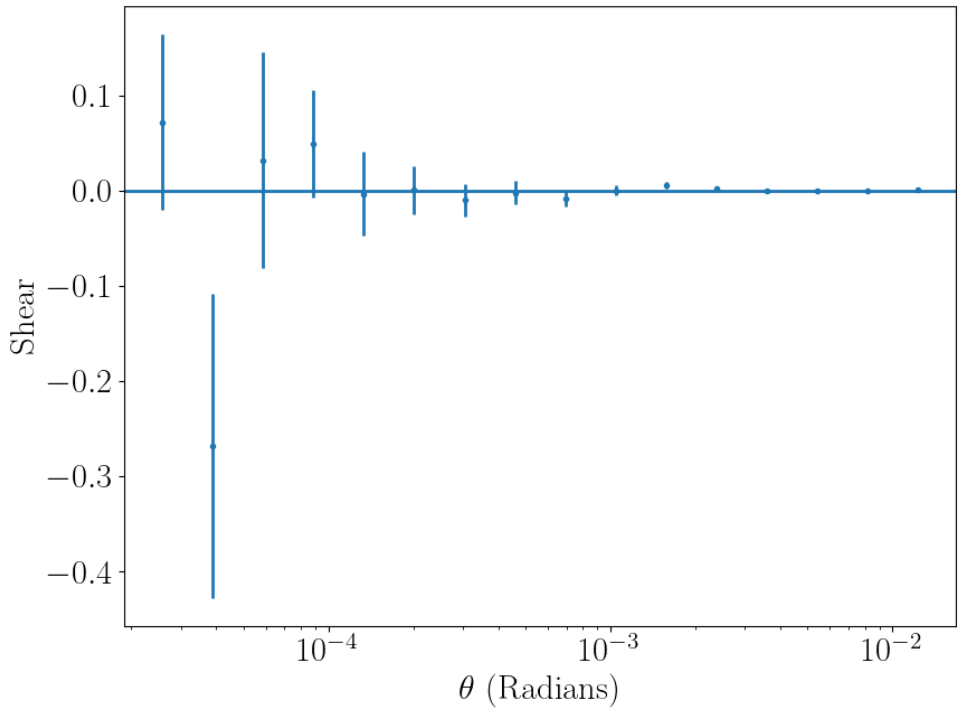

Figure 3.2: Tangential shear as a function of angular separation for the combined noise only signal. The plot shows no significant trend, as expected with shape noise. A positive deviation is shown as small angular separations, likely due to most probably due to random statistical fluctuations in low SNR bins.

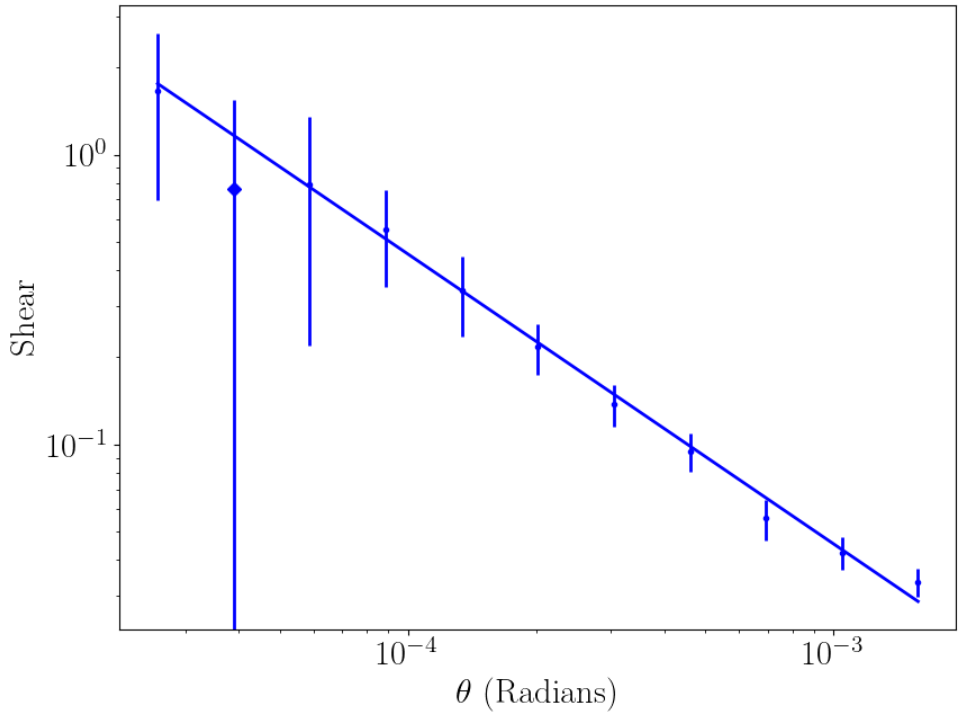

Figure 3.3: Tangential shear as a function of angular separation for the combined pure + noise signal. The plot shows a detectable lensing signal with reduced $\chi^2 \approx 0.36$ for 10 degrees of freedom. The solid line represents the best-fitting SIS model. This figure provides a realistic observation-like signal combining lensing and shape noise.

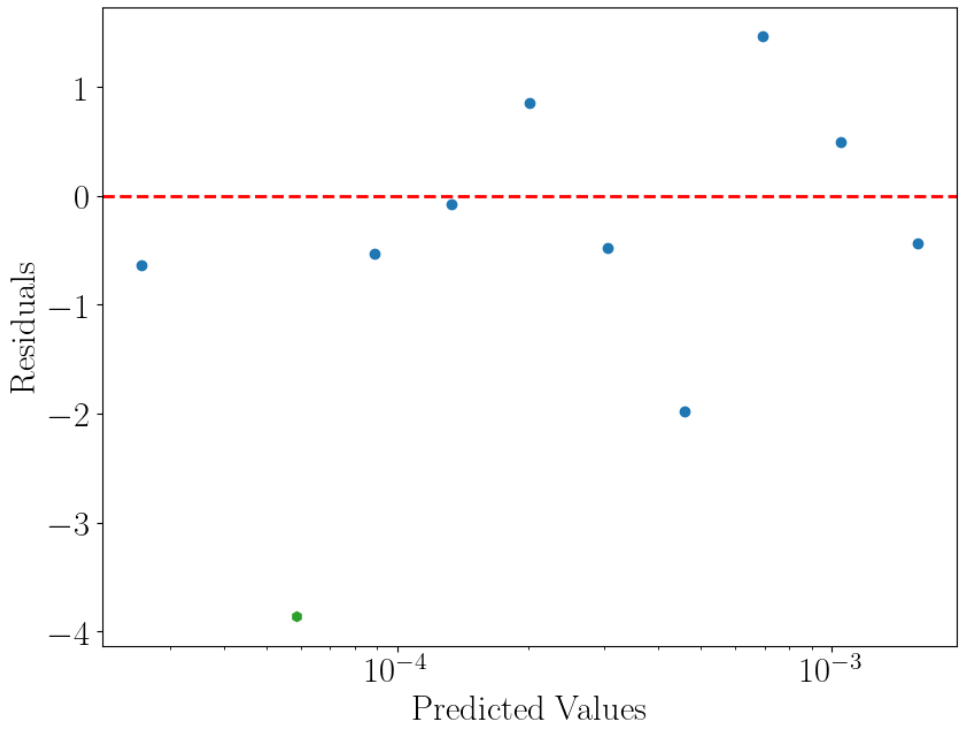

Figure 3.4: This plots displays the difference between the observed and model predicted radially binned tangential shear measurements for pure signal. It shows one significant outlier at low radii, marked by a green hexagon, with a weak positive systematic trend for the overall residuals. Another outlier with $SNR < 1.25$ has been omitted from the plot to improve clarity.

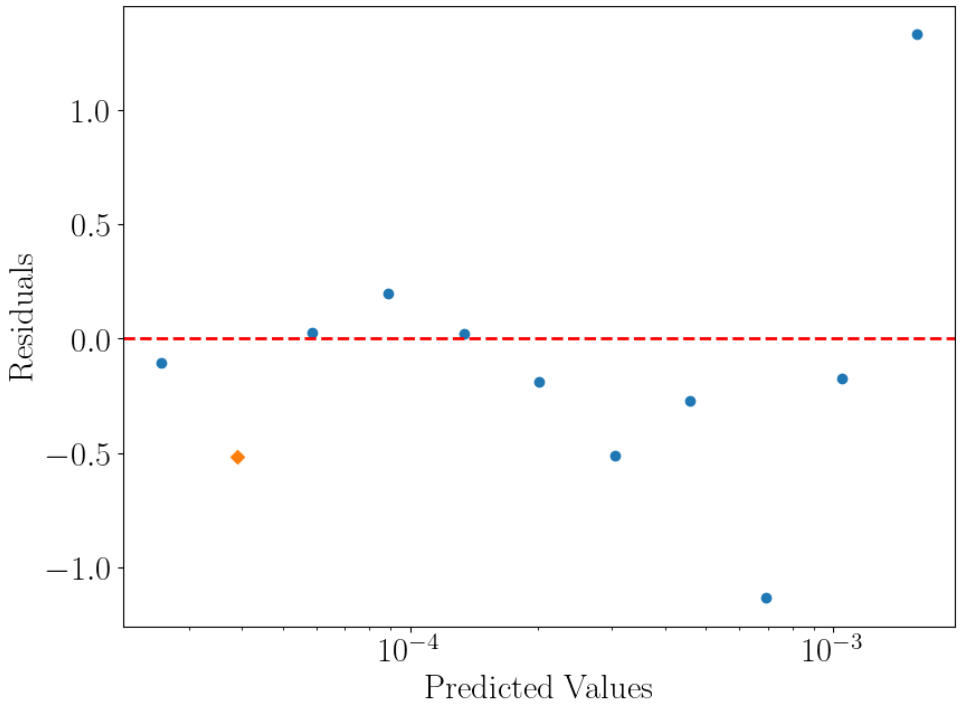

Figure 3.5: This plots shows no systematic trend for the residuals for the radially binned tangential shear measurements for pure + noise signal.

Chapter 4

Discussion

I measured the tangential shear for a synthetic galaxy cluster using weak gravitational lensing techniques, providing results for pure signal, noise signal, and pure + noise signal. We fitted my measurements with the Singular Isothermal Sphere (SIS) model and obtained a power-law trend that yielded a statistically acceptable fit, with a reduced chi-squared value of $\chi^2 \approx 0.86$ for the pure signal and $\chi^2 \approx 0.36$ for the pure + noise signal, which included shape noise.

From the best-fitting SIS model we obtained useful insights for the mass distribution of the synthetic galaxy cluster for both the pure and the pure + noise signals. The Einstein radius for the pure signal closely matched values observed, such as $\theta_E \approx 5 \times 10^{-5}$ rad, for the galaxy cluster CL0024+1654 at redshift $z = 0.39$ (J. Anthony Tyson, Kochanski, and Dell'Antonio, 1998). The Einstein radius increased from $\theta_E \approx (4.48 \pm 0.29) \times 10^{-5}$ rad to $\theta_E \approx (9.08 \pm 2.13) \times 10^{-5}$ rad from the pure signal to the pure + noise signal, respectively. This double in θ_E with the addition of shape noise highlighted the influence of shape noise on the model, which led to a significant overestimation of the Einstein radius. The estimate of velocity dispersion also followed this trend, increasing from $\sigma_v \approx (8.50 \pm 0.27) \times 10^5\,\mathrm{ms}^{-1}$ to $\sigma_v \approx (12.10 \pm 1.42) \times 10^5\,\mathrm{ms}^{-1}$. Despite this increase, the velocity dispersion remained the same order of magnitude as the prediction, $\sigma_v \approx 1.4 \times 10^6\,\mathrm{ms}^{-1}$, made in Section 2.5. Likewise, the enclosed mass within the Einstein radius grows considerably with the inclusion of noise increasing from $M(< \theta_E) \approx (2.17 \pm 0.20) \times 10^{13}\,M_\odot$ to $M(< \theta_E) \approx (8.92 \pm 2.96) \times 10^{13}\,M_\odot$. Although this demonstrates the sensitivity to noise of the mass estimates, both remain within a reasonable theoretical range for a galaxy cluster.

To improve the robustness of my fit, we implemented residual clipping, removing data points with residuals exceeding 3σ from the fit which reduced the impact of outliers. With high residual points removed, the fitted profile stabilised, leading to a more reliable measurement of θ_E. We further improved my fit, applying relative error filtering that eliminated data points with a relative error above 80% to maintain a signal-to-noise ratio $SNR > 1.25$. This ensured that we excluded the most unreliable measurements from my fit.

The residual analysis revealed two notable outliers. The first outlier, found in the pure signal, was omitted from its residual plot (Figure 3.4) for clarity of analysis. This outlier, marked by a diamond as seen in Figure 3.1, had a residual $r \approx 60$ suggesting a significant deviation from the predicted SIS model. However, for both the pure signal and the pure + noise signal, this bin contained only 1 measurement with $SNR < 1.25$ making it highly susceptible to statistical noise and shape noise. Therefore, the point was excluded from both fits with my relative error clipping method and provides no evidence that the data do not fit the predicted SIS model. The second outlier was revealed in Figure 3.4. Located at a small angular separation and marked by a hexagonal point, the data point was located approximately 4σ below the prediction of the SIS model. This implied a significant deviation from the expected behaviour of the SIS profile in the cluster core. However, its significance was limited as the bin contained only 2 measurements with a $SNR \approx 1.4$, suggesting high statistical uncertainty. While, both residual outliers could have reflected a true flattening of the shear profile close to the galaxy centre, which is a known

limitation of the SIS model, the outliers are not sufficiently credible to suggest a significant deviation from the SIS model on their own.

Further insight was gained from the pure signal residual plot (Figure 3.4), which showed a weak positive residual trend. This suggested that the model was over predicting the shear and low radii and under predicting at higher radii. This was not reproduced in the pure + noise signal residual plot (Figure 3.5), where the addition of noise masked any clear pattern. With the noise included, the residual then showed no clear systematic trend demonstrating that the SIS model described the data well for most radii.

However, analysis into the noise signal (Figure 3.2) showed that for low radii the noise had a positive deviation, likely due to statistical fluctuations in low SNR bins. These increased shear values will have distorted the fit at low radii, where the measurements were already highly susceptible to noise due to the limited number of measurements per bin. Consequently, the significant increase in θ_E, σ_v and $M(< \theta_E)$ estimates that are observed for the pure + noise signal are most plausibly explained by the positive deviation of the noise for low radii bins. Therefore, my model predictions likely would have been improved with a stricter threshold SNR value for my relative error clipping method.

Despite this, both the outliers and the pure signal weak positive residual trend were consistent with observational evidence that the inner shear profile of real galaxy clusters tended to be shallower, and the outer shear profile steeper, than predicted in the SIS model (Umetsu et al., 2011). These findings suggested that the fit could be improved by either applying constraints on angular separation at low and high radii or adopting a more complex mass profile. An improved model to describe the mass distribution of galaxy clusters is the Navarro-Frenk-White (NFW) model, proposed by Navarro, Frenk, and White (1996) and Navarro, Frenk, and White (1997), based on result from high resolution N-body simulations:

$$\rho(r) = \frac{\delta_c \, \rho_{\rm cr}(z)}{(r/r_s)(1 + r/r_s)^2}. \tag{4.1}$$

Here, r_s is the scaled radius and it characterises the radius at which the density profile transitions from the inner to the outer behaviour. Near the centre ($r \ll r_s$) the density goes as r^{-1}, while at large radii ($r \gg r_s$) it falls off as r^{-3}. This model avoids the unphysical mass predictions of the SIS model, specifically the divergent mass prediction as $r \to 0$. The NFW instead predict a mass profile $M(r) \propto r^2$ near the centre models which for $r \to 0$ the mass enclosed goes smoothly to zero rather than diverging to infinity, as in the SIS model. Therefore, NFW model gives a more realistic description of the mass distribution, particularly in the core and outer regions, providing an improvement over the model SIS model for a galaxy cluster.

Chapter 5

Conclusion

In this study, weak gravitational lensing techniques were used to explore the mass distribution of a synthetic lensing galaxy cluster. By analysing complex ellipticity measurements and fitting the SIS model we successfully recovered the key physical parameters, including the Einstein radius, the velocity dispersion and the enclosed mass. The results for both the pure signal and the pure + noise signal dataset showed good agreement with realistic theoretical predictions, demonstrating the effectiveness of the methods developed. However, for further insight into the limitations of the SIS model, a larger dataset was needed as the small amount of tangential shear measurements at lower radii hindered analysis of the SIS model.

Although the SIS model provided a useful initial framework, its use was limited by its assumptions such as a spherically symmetric mass distribution. If I had more time on the project the natural progression would be to have explored the use of a more complex model for the mass distribution such as the Navarro–Frenk–White (NFW) profile and comparing the results to those achieved using the SIS model.

Overall, this project demonstrated that even a simple model can yield meaningful astrophysical insights and provides with a strong foundation for further investigation with more complex models.

Acknowledgements

I would like to thank Robert E. Smith for his expertise and support as my supervisor on this project, and for guiding me through the concepts of weak gravitational lensing and providing valuable resources for independent study.

Bibliography

Bacon, D. J., A. R. Refregier, and R. S. Ellis (2000). "Detection of weak gravitational lensing by large-scale structure". In: *Monthly Notices of the Royal Astronomical Society* 318, pp. 625–640 (cit. on p. 3).

Einstein, Albert (Dec. 1936). "Lens-Like Action of a Star by the Deviation of Light in the Gravitational Field". In: *Science* 84.2188, pp. 506–507. DOI: 10.1126/science.84.2188.506 (cit. on p. 2).

Navarro, J. F., C. S. Frenk, and S. D. M. White (1996). "The Structure of Cold Dark Matter Halos". In: *The Astrophysical Journal* 462, pp. 563–575. DOI: 10.1086/177173 (cit. on p. 21).

— (1997). "A Universal Density Profile from Hierarchical Clustering". In: *The Astrophysical Journal* 490, pp. 493–508. DOI: 10.1086/304888 (cit. on p. 21).

Schneider, P. (Jan. 2006). "Part 1: Introduction to gravitational lensing and cosmology". In: *Saas-Fee Advanced Course 33: Gravitational Lensing: Strong, Weak and Micro*. Ed. by Georges Meylan et al., pp. 1–89 (cit. on pp. 5, 4).

Tyson, J. A., F. Valdes, and R. A. Wenk (Jan. 1990). "Detection of Systematic Gravitational Lens Galaxy Image Alignments: Mapping Dark Matter in Galaxy Clusters". In: 349, p. L1. DOI: 10.1086/185636 (cit. on p. 2).

Tyson, J. Anthony, Greg P. Kochanski, and Ian P. Dell'Antonio (May 1998). "Detailed Mass Map of CL 0024+1654 from Strong Lensing". In: 498.2, pp. L107–L110. DOI: 10.1086/311314. arXiv: astro-ph/9801193 [astro-ph] (cit. on p. 20).

Umetsu, Keiichi et al. (Sept. 2011). "A Precise Cluster Mass Profile Averaged from the Highest-quality Lensing Data". In: 738.1, 41, p. 41. DOI: 10.1088/0004-637X/738/1/41. arXiv: 1105.0444 [astro-ph.CO] (cit. on p. 21).

Van Waerbeke, L. et al. (2000). "Detection of correlated galaxy ellipticities from CFHT data: first evidence for gravitational lensing by large-scale structures". In: *Astronomy Astrophysics* 358, pp. 30–44 (cit. on p. 3).

Walsh, D., R. F. Carswell, and R. J. Weymann (May 1979). "0957+561 A, B: twin quasistellar objects or gravitational lens?" In: 279, pp. 381–384. DOI: 10.1038/279381a0 (cit. on p. 2).

Wittman, D. M. et al. (2000). "Detection of weak gravitational lensing distortions of distant galaxies by cosmic dark matter at large scales". In: *Nature* 405, pp. 143–148 (cit. on p. 3).

Zwicky, F. (Feb. 1937). "Nebulae as Gravitational Lenses". In: *Physical Review* 51.4, pp. 290–290. DOI: 10.1103/PhysRev.51.290 (cit. on p. 2).

Appendix A

Code

A.1 Shear Method

```
    def Shear(theta, gamma1, gamma2):
    """
    Calculates both the cross and tangential shear for each position.

    gamma_t = -Re[gamma * exp(-2i * phi)]
    gamma_x = -Im[gamma * exp(-2i * phi)]


    Paramaters:
        theta : 2d float array
            Theta is the angular position of each star. [x, y]

        gamma1 : float
            shear in the x direction

        gamma2 : float
            shear in the y direction

    Returns:
        gamma_t : float array
            Tangential Shear values

        gamma_x : float array
            Cross Shear values
    """

    gamma_t = []
    gamma_x = []

    for i in range(len(posx)):

        mag_theta = np.sqrt(np.square(theta[i][0]) + np.square(theta[i
           ][1]))

        gamma = gamma1[i] + 1j * gamma2[i]

        if theta[i][1] >= 0:
            phi = np.arccos(posx[i] / mag_theta)

        else:
            phi = - np.arccos(posx[i] / mag_theta)
```

```
41
        gamma_t.append(
43          - np.real(gamma * np.exp(-2j * phi))
            )
45      gamma_x.append(
            - np.imag(gamma * np.exp(-2j * phi))
47      )

49  return gamma_t, gamma_x
```

A.2 Convergence Method

```
1   def Kappa(theta, einstein_limit, einstein_limit_err):
    """
3   Calculate Kappa - the convergence.

5   Paramaters:
        theta : float array
7           Is the magnitude position from the origin in radians

9       einstein_limit : float
            Einstein Limit
11
    Returns:
13      kappa : float array
            array of kappa values for theta values
15  """

17  einstein_limit_  = np.ones((len(theta)))
    einstein_limit_ *= einstein_limit
19
    kappa = einstein_limit_ / (2 * theta)
21
    kappa_err = np.ones((len(theta)))/(2*theta) * einstein_limit_err
23
    return [theta, kappa, kappa_err]
```

A.3 Plotting Method

```
    def PlottingFunction(something, title, xlabel, ylabel, colour, label
       ):
2       """
        Plots the data something in log-log scale with relative error
           clipping, residual clipping, and chi_squared fitting. Then
           output the plot parameters
4       using print. Plots the data with error bars and the rejected
           points doesn't display the plot. Uses hexigons for relative
           error clipped points, diamonds
        for residual clipped points and circles for the included points
6
        Parameters:
8           something : 3 dimensional array
                Includes all the information of the data we are trying
                   to plot. something = [something_x, something_y,
                   something_error].
10
            title : string
12              The title of the plot.
```

```
            xlabel : string
                X label of the plot

            ylabel :string
                Y label of the plot

            colour : string
                colour of points and fit for the plot

            label : string
                label of the plot. Useful when plotting multiple thing
                    on the plot.
        """


        # Only fit the data if it isn't the noise plot
        if (label != "tangential_shear_ring_average_noise"):
            # Defines the fixed gradient of -1
            a1 = -1
            Var_a1 = 0

            # Removes any unwanted data
            something = np.array(DataSanitisation(something))

            # Reject points with relative error > 50% (diamond)
            something, points_big_error = RelativeErrorClipping(
                something)

            # Calculate the fit paramaters for a linear fit in log space
            a0 = FixedGradientFit(np.log(something[0]), np.log(something
                [1]), something[2] / something[1])

            # Calculate Residuals
            something, points_big_res = ResidualClipping(something, a0,
                a1)

            # Refit the data
            a0 = FixedGradientFit(np.log(something[0]), np.log(something
                [1]), something[2] / something[1])

            # y values from the model
            y_model = LinSpaceModel(np.log(something[0]), a1, a0)

            # Calculate the errors in the y-intercept and gradient
                estimates
            Var_a0 = FixedSlopeInterceptError(np.log(something[0]),
                something[2] / something[1], a1)

            # Adds units to the gradient (per radian)
            a1 *= u.rad**-1

            # Output the y-intercept a0 and gradient a1 fitting
                parameters
            print("a0:= %10.7f ; Err_a0 := %10.7f"%(a0,np.sqrt(Var_a0)))
            print("a1:= %10.7f %10s ; Err_a1 := %10.7f %10s"%(a1.value,
                a1.unit, np.sqrt(Var_a1), a1.unit))
```

```
            # Calculate reduced chi_sqaured
            dof = len(something[0])
            chisqA = ReducedChiSquared(np.log(something[0]), np.log(
                something[1]), something[2] / something[1], LogSpaceModel
                , a1.value, a0, dof)

            # Output the reduced chi_squared for the fit
            print("Reduced Chisquare for model A:= %14.7f, number of
                degrees of freedom:= %3d"%(chisqA,dof))

            # Plot the fit
            plt.plot(something[0], y_model, color=colour)

            # Add removed points
            if len(points_big_res) != 0:
                plt.errorbar(points_big_res[0], points_big_res[1], yerr=
                    points_big_res[2], fmt = f"{colour}h")

            if len(points_big_error) != 0:
                plt.errorbar(points_big_error[0], points_big_error[1],
                    yerr=points_big_error[2], fmt = f"{colour}D")


        # If it is the noise plot then set plotting parameters to 0
        else:
            a0 = 0
            a1 = 0
            Var_a0 = 0
            Var_a1 = 0

        plt.errorbar(something[0], something[1], yerr=something[2], fmt
            = f"{colour}.", label=label)

        # plt.title(title)
        plt.xlabel(xlabel)
        plt.ylabel(ylabel)

        plt.xscale("log")
        plt.yscale("log")

        # plt.legend()

        if (label != "tangential_shear_ring_average_noise" and label !=
            "kappa_ring_average" and label != "kappa_bar" and label != "
            Kappa_Ring_Average - Kappa_Bar vs Theta (Pure)"):
            plt.show()
            # Residual Plot
            ResidualPlot(something, points_big_error, points_big_res, a1
                , a0, label)

        return a0, a1, np.sqrt(Var_a0), np.sqrt(Var_a1)
```

A.4 Radial Binning Method

```
    def RadialBinningMethod(theta, y, signal_type, Nbins=16):
        bin_starts, bin_centres, bins, bin_averages, delta_theta =
            CreateLogScaleBins(theta, Nbins)
        bin_errors = np.array([])
```

```
        for i_bin in range(len(bin_starts)):
            values = np.array(BinValues(theta, delta_theta[i_bin], y,
                bin_starts[i_bin]))
            bins = np.append(bins, values)
            average = np.average(values)
            bin_averages = np.append(bin_averages, average)
            bin_errors = np.append(bin_errors, BinErrors(average, len(
                values), signal_type))

        bin_centres, bin_averages, bin_errors = RemoveEmptyBins(
            bin_centres, bin_averages, bin_errors)

        return bin_centres, bin_averages, bin_errors
```

```
    def BinValues(theta_array, delta_theta, gamma_t, bin_start):
        values = np.array([])

        for i in range(len(theta_array)):
            # Check if the gamma value is for a position that falls
                within the shell

            if theta_array[i] >= bin_start and theta_array[i] <
                bin_start + delta_theta:
                values = np.append(values, gamma_t[i])

        return values
```

```
    def CreateLogScaleBins(theta, Nbins):
        """
        Creates the bins for plotting the data.


        Paramaters:
            theta : float array
                Array of radial distances from the origin for each
                    galaxy. Measured in anglar units

            Nbins : int
                Number of bins

        Returns:
            bin_starts : array

            bin_centres : array

            bins : 2d array
                Array of values that are in each bin

            bin_averages : array
        """

        min_theta = np.log(min(theta))
        max_theta = np.log(max(theta))

        delta_theta = (max_theta - min_theta) / (Nbins - 1)

        bin_starts =  np.arange(min_theta - delta_theta / 2, max_theta +
            delta_theta / 2, delta_theta)
```

```
        bin_centres = bin_starts + delta_theta / 2

        bin_starts = np.exp(bin_starts)
        bin_centres = np.exp(bin_centres)

        # Calculate delta_theta for each bin in linear scales
        delta_theta = 2*(bin_centres - bin_starts)

        bins = np.array([[]])
        bin_averages = np.array([])

        return bin_starts, bin_centres, bins, bin_averages, delta_theta
```

```
        def RemoveEmptyBins(bin_centres, bin_averages, bin_errors):
            del_indexs = []

            for i in range(len(bin_centres)):
                if bin_errors[i] == 0 or bin_errors[i] == np.inf or
                   bin_averages[i] == 0:
                    del_indexs.append(i)

            bin_centres   = np.delete(bin_centres, del_indexs)
            bin_averages  = np.delete(bin_averages, del_indexs)
            bin_errors    = np.delete(bin_errors, del_indexs)

            return bin_centres, bin_averages, bin_errors
```

A.5 Radial Binning Uncertainty Method

```
    def BinErrors(bin_average, N, signal_type):
        # The error are the standard deviation / the number of values in
            the bin
        sigma_tangential_elipticity = 0.16

        # If nothing in bin then bin has no error
        if N == 0:
            return 0

        pure_error = bin_average / np.sqrt(N)
        noise_error = sigma_tangential_elipticity / np.sqrt(N)

        if signal_type == "pure":
            return pure_error

        elif signal_type == "noise":
            return noise_error

        else: # noise + pure
            return np.sqrt(pure_error**2 + noise_error**2)
```

A.6 Relative Error Clipping Method

```
    def RelativeErrorClipping(something, relative_error_threshold=0.8):
        """
        Rejects and points with a relative error >
           relative_error_threshold

        Parameters:
```

```
            something : 3 dimensional array
                Includes all the information of the data we are trying
                    to plot. something = [something_x, something_y,
                    something_error].

            relative_error_threshold: float
                The relative error we should clip at. Default 0.8

        Returns:
            something : 3 dimensional array
                new something array with points removed

            points_big_error : 3 dimensional array
                Array of data including the points that have been
                    clipped.
        """

        mask = np.array((something[2] / something[1]) <
            relative_error_threshold)
        points_big_error = np.array([something[0][~mask], something[1][~
            mask], something[2][~mask]])
        something = np.array([something[0][mask], something[1][mask],
            something[2][mask]])

        return something, points_big_error
```

A.7 Residual Clipping Method

```
    def ResidualClipping(something, a0, a1):
        """
        Rejects and points with a large residuals

        Parameters:
            something : 3 dimensional array
                Includes all the information of the data we are trying
                    to plot. something = [something_x, something_y,
                    something_error].

            a0 : float
                y-intercept from the fit

            a1 : float
                gradient from the fit

        Returns:
            something : 3 dimensional array
                new something array with points removed

            points_big_res : 3 dimensional array
                Array of data including the points that have been
                    removed.
        """

        res = np.array(something[1] - LinSpaceModel(np.log(something[0])
            , a1, a0))

        # Reject points with residuals > 3 * error (hexagon)
        mask = np.array(np.abs(res) < 3 * something[2])
```

```
        points_big_res = np.array([something[0][~mask], something[1][~
            mask], something[2][~mask]])
        something      = np.array([something[0][mask], something[1][mask
            ], something[2][mask]])

        return something, points_big_res
```

A.8 Reduced Chi-Squared Fitting Method

```
    def FixedGradientFit(x, y, sig, fixed_slope=-1):
        """
        Performs a linear fit with a fixed slope.

        Parameters:
        x : array
            x-values
        y : array
            y-values
        sig : array
            y-error bars
        fixed_slope : float
            Slope to fix during the fitting process (default = -1)

        Returns:
        a0_est : float
            Estimated intercept for the fixed slope fit
        """
        sig2 = sig**2
        residuals = y - fixed_slope * x
        weights = 1 / sig2
        a0_est = np.sum(residuals * weights) / np.sum(weights)
        return a0_est
```

A.9 Reduced Chi-Squared Fitting Error Method

```
    def FixedSlopeInterceptError(x, sig, m):
        """
        Computes the variance of the intercept for a fixed-slope linear
            fit.

        Parameters:
            x : array
                x-values
            sig : array
                y-error bars
            m : int
                gradient

        Returns:
            Var_a0 : float
                Variance of the intercept
        """

        sig2 = sig**2
        W = 1.0 / sig2
        S = np.sum(W)
        Sx = np.sum(x * W)
```

```
        Var_a0 = np.sum(W * (1 + m * x)**2) / S**2

        return Var_a0
```

A.10 Reduced Chi-Squared Method

```
    def ReducedChiSquared(x, y, sig, ymodel, m, c, dof):
        """
        Parameters:
            x : array
                x values.

            y : array
                y values.

            sig : array
                Errors.

            ymodel : func
                theoretical model

            m : float
                Gradient of fit

            c : float
                Intercept of fit

            dof : int
                Number of degrees of freedom.

        Returns:
            redChiSq : float
                The reduced chi-squared of the fit.
        """

        ytheory = ymodel(x, m, c)
        ZZ = (y-ytheory)/sig
        ZZSq = ZZ**2
        chiSq = np.sum(ZZSq)
        redChiSq = chiSq / dof
        return redChiSq
```

A.11 Linear Space Model Method

```
    def LinSpaceModel(x, m, c):
        return np.exp(m*x + c)
```

A.12 Log Space Model Method

```
    def LogSpaceModel(log_x, m, c):
        return m * log_x + c
```

A.13 Data Sanitisation Method

```
    def DataSanitisation(something):
        remove_list = []
```

```
        # Remove any values with a theta value of above threshold
        for i in range(len(something[1])):
            if something[0][i] > 0.002 or something[1][i] <= 0:
                remove_list.append(i)

        something[0] = np.delete(something[0], remove_list)
        something[1] = np.delete(something[1], remove_list)
        something[2] = np.delete(something[2], remove_list)

        return something
```

A.14 Einstein Radius Method

```
    def EinsteinRadius(y_intercept, y_intercept_err):
        """
        Calculate the Einstein radius from the y intercept of the fit

        Parameters:
            y_intercept : float
                The y_intercept of the plot.

            y_intercept_error : float
                The error of the y_intercept of the plot

        Returns
            einstein_radius : float
                The calculated einstein radius

            error : float
                The error on the calculated einstein radius
        """

        einstein_radius = 2*np.exp(y_intercept)
        error = np.exp(y_intercept) * y_intercept_err
        return einstein_radius, error
```

A.15 Velocity Dispersion Method

```
        def VelocityDispersion(D_s, D_ds, einstein_radius,
            einstein_radius_err):
        """
        Calculates Velocity Dispersion

        Paramaters:
            D_s : float
                Distance to the source plane

            D_ds : float
                Distance to the source plane from the lens plane

            c : float
                y-intercept of the tangential_shear_ring_average graph

        Returns:
            velocity_dispersion : float
                The amount that the velocity of the galaxies deviate
                    from the mean velocity [m s  -1]
        """
```

```
        velocity_dispersion = scipy.constants.c * np.sqrt((
            einstein_radius*D_s)/(4*np.pi*D_ds)) * u.m * u.s**-1
        error = (einstein_radius_err/(2*einstein_radius)) *
            velocity_dispersion

        return velocity_dispersion, error
```

A.16 Mass Method

```
    def Mass(D_d, velocity_dispersion, velocity_dispersion_err,
        einstein_radius, einstein_radius_err):
        """
        Calculate the estimated mass from the fit

        Parameters:
            D_d : float
                Distance to the lens plane from the observer [parsec]

            velocity_dispersion : float
                The velocity dispersion estimated from the fit [ms^-1]

            velocity_dispersion_err : float
                The error on the velocity dispersion [ms^-1]

            einstein_radius : float
                The einstein radius estimated from the fit [unitless /
                    rad]

            einstein_radius_err : float
                The error on the einstein radius [unitless / rad]

            Returns:
                M : float
                    The estimated mass [kg]

                error : float
                    The error on the estimated mass [kg]

        """
        # Convert D_d from parsecs to metres
        D_d = D_d.to(u.m)
        M = (np.pi*velocity_dispersion**2*D_d*einstein_radius) / (scipy.
            constants.G * u.m**3 * u.kg**-1 * u.s**-2)
        error = np.sqrt(4 * (velocity_dispersion_err /
            velocity_dispersion)**2 +(einstein_radius_err /
            einstein_radius)**2) * M
        return M, error
```