Dissertation

submitted to the

Combined Faculties for the Natural Sciences and for Mathematics

of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

presented by

Diplom-Physiker Jarosław P. Rzepecki

born in Olsztyn, Poland

Oral examination: December 12, 2007

On the Inversion Methods

of

Strong Gravitational Lensing

Referees:	Prof. Dr. Matthias Bartelmann
	Prof. Dr. Joachim Wambsganss
Supervisors:	Dr. Marco Lombardi
	Dr. Piero Rosati

Zusammenfassung

In dieser Dissertation präsentiere ich eine neue Methode zur Inversion der Linsengleichung. Der präsentierte Algorithmus basiert auf dem Modifizierten Hausdorff-Abstand zwischen beobachteten und modell- erzeugten Bögen als eine Funktion der Fit-Güte. Die Minimierung dieser Funktion der Fit-Güte hat sich als sehr erfolgreich zur Auffindung des Linsenpotentials herausgestellt, da sie relativ einfach zu berechnen ist und die volle Information nicht nur der Bogenpositionen sondern auch ihrer Form enthält. Weitere Minimierungs- Schemata, darunter Powell, Markov Chain Monte Carlo und Genetische Algorithmen, werden untersucht, und es wird gezeigt, dasz sie kombiniert die besten Resultate liefern. Die entwickelte Methode wird mit groszem Erfolg auf den Starklinsen-Galaxienhaufen RCS0224-0002 angewandt. Das resultierende Modell ist in der Lage, alle beobachteten Starklinsen-Eigenschaften dieses Haufens im Detail wiederzugeben, be- vorzugt das NIS radiale Profil gegenueber NFW und sagt eine verblüffende Verschiebung um fuenf Bogensekunden zwischen dem Massenschwerpunkt und dem Zentrum der Roentgenemission voraus. Zum Abschlusz wird die Doku- mentation des Softwarepakets (C++ und IDL) zur Implementierung der Inversionsmethode praesentiert.

Abstract

In this dissertation I present a novel method of inverting the lensing equation. The presented algorithm is based on Modified Hausdorff Distance between observed and model produced arcs as a goodness of fit function. Minimizing this goodness of fit function has proven to be very successful in finding the lensing potential, since it is relative simple to compute and contains the full information about not only the arcs positions, but also their shapes. Further various minimization schemes, including Powell, Markov Cain Monte Carlo and Genetic Algorithms, are investigated and proved to give the best results if combined together. The developed method is applied, with great success, to the strong lensing galaxy cluster RCS0224-0002. The resulting model is able to reproduce, in detail, all the observed strong lensing features of this cluster, favors the NIS radial profile over NFW, and predicts an intriguing 5 arc sec shift between center of the mass and the X-ray emission. Finally the documentation of the software package (C++ and IDL) implementing the inversion method is presented.

Contents

Acknowledgments

Above all, I would like to thank Dr. Marco Lombardi and Dr. Piero Rosati for their supervision and getle guidance through the world of Astronomy. My gratitude also goes to the European Southern Organization for hosting me during my PhD studies and providing a wonderful scientific atmosphere.

Jarosław P. Rzepecki
Garching
October 16, 2007

Chapter 1

Introduction to Gravitational Lensing

To mistrust science and deny the validity of the scientific method is to resign your job as a human. You'd better go look for work as a plant or wild animal.

P. J. O'Rourke

Abstract

This chapter presents the introduction to the gravitational lensing. After a brief historical note we focus on the physical origins and properties of this phenomenon. We start from investigating how a matter (energy) over density affects the light ray path in the framework of the General Relativity. Then we introduce and justify a number of simplifications which, together with the thin lens approximation allow us to derive many simple and useful equations and to describe some crucial properties of gravitational lensing.

1.1 Introduction

Gravitational lensing is an effect that arises due to the fact that gravity influences the paths of light rays. The more massive an object is, the more it will change the trajectory of a light ray passing by. This effect is properly described in the framework of General Relativity. Gravitational lensing is a very rapidly developing branch of astrophysical research. It receives much attention since it allows for the most direct study of the dark matter on different astrophysical scales, regardless of its component and dynamical state (Peacock and Schneider 2006). Depending on the degree to which the light rays trajectories are changed by the presence of the lensing object (hereafter called *lens*), and on the nature of the lens and the background source (hereafter called *source*)

being lensed, three "regimes" of gravitational lensing might be distinguished: strong lensing, weak lensing, cosmic shear, pixel lensing and microlensing.

Strong lensing is when there is enough distortion caused by the lens in the path of the light rays from the source to the observer, to produce easily visible features such as Einstein rings, arcs, and multiple images. It is most often observed in the cores of galaxy clusters (with a source being a distant galaxy) and in galaxies (with the source being a distant quasar). However, there are cases of quasar being lenses by a galaxy cluster, or galaxies being lenses by another galaxy. When the distortion of the background sources is so small that statistical methods need to be used to detect it, then we are in the weak lensing regime. This kind of lensing is commonly observed in the outer regions of galaxy clusters. The cosmic shear is the weak distortion to the shape and luminosity of high redshift objects, caused by the large scale structure of the Universe. The third type of lensing, microlensing, is a time dependent variation of the observed flux of the source due to the change in the alignment of the lens and the source. The common example of microlensing is lensing of stars in another galaxy by stars from the Milky Way.

In this dissertation I will mainly focus on the strong gravitational lensing caused by the clusters of galaxies, and on methods of lenses mass profiles reconstruction based on the observed strong lensing features.

There are a number of works available, containing general information on gravitational lensing, for example a book by Schneider, P. and Ehlers J. and Falco E. (1992), or the review by Meylan et al. (2006). Most of the cosmology books, also have chapters dedicated to lensing effects (see e. g. Peacock 1999, chap. 2).

1.1.1 Importance of lensing

All types of lensing have very interesting applications to astronomy, cosmology, and astrophysics. The most direct usage of gravitational lensing is the measurement of mass and mass profiles. Observations of the positions of multiple images (arcs) allow a rough estimate of the mass enclosed by those images. When more than one arc system is observed and additional constrains are taken into account (like arc shapes, fluxes, and if available, time delays among images), than a more precise mass distribution estimate is

possible. In the case of galaxy clusters, this combined with the weak lensing constrains results in a detailed maps of dark matter, both in the core and in the outer regions of the clusters (see e. g. Cacciato et al. 2006).

Determination of the H_0

The Observation of more than one arc system with known redshift, and/or time dilatation between images, allows the calculation of the Hubble constant. The values of H_0 obtained from lensing are usually below $72\ \mathrm{km\ s^{-1}\ Mpc^{-1}}$ and have significant errors associated with them, mostly due to uncertainties in the lenses modeling. The recent result obtained by Saha et al. (2006) is that the Hubble constant is

$$H_0^{-1} = 13.5^{+2.5}_{-1.2}\mathrm{Gyr} \quad (H_0 = 72^{+8}_{-11}\mathrm{km\,s^{-1}\ Mpc^{-1}}) \tag{1.1}$$

at 68% confidence level. The Fig. 1.1 shows the ensemble of Hubble times obtained by investigating 10 different lenses with measured time delays. The lenses were modeled with *PixeLens* software (see Sect. 2.1.1), which resulted in a large number of produced models, that were later averaged.

Cosmological parameters

Other cosmological parameters can also be obtained from lensing, for example weak lensing caused by large scale structure is sensitive to the matter density parameter Ω_M and to the density fluctuations normalization parameters σ_8. The results based on 22 deg^2 of *CFHTLS*[1] Wide Synoptic Survey data analyzed by Hoekstra et al. (2006) are presented in Fig. 1.2. The study puts the upper limit of -0.3 on the dark energy state parameter with 99.7% confidence. The degeneracy between Ω_m and σ_8 is not parallel to the one that arises in the *WMAP*(Spergel et al. 2007) data analysis. This technique of cosmological parameters estimation is promising, especially in the perspective of the new surveys that will provide researchers with coverage of the big portions of the sky in many bands, which is important for the photometric redshift measurements.

[1] http://www.cfht.hawaii.edu/cfhtls/

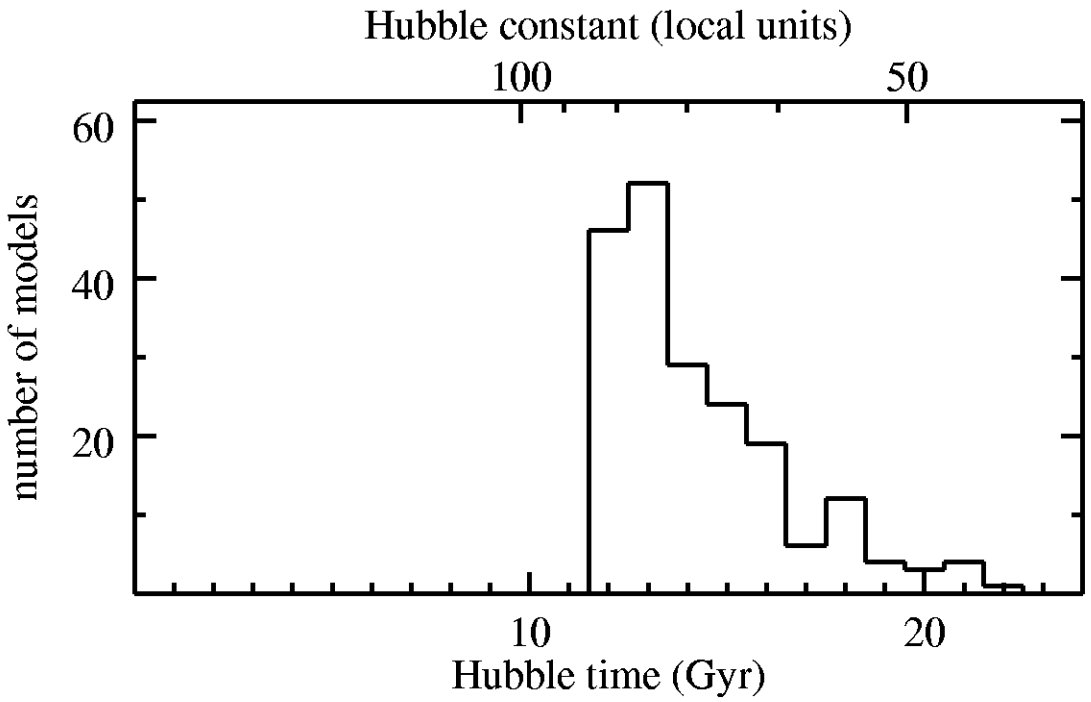

Figure 1.1: *Histogram of the ensemble of H_0^{-1} values. The unbinned distribution gives $H_0^{-1} = 13.5^{+2.5}_{-1.2}\mathrm{Gyr}$ at 68% confidence level and $13.5^{+5.6}_{-1.6}\mathrm{Gyr}$ at 90% confidence level (Saha et al. 2006).*

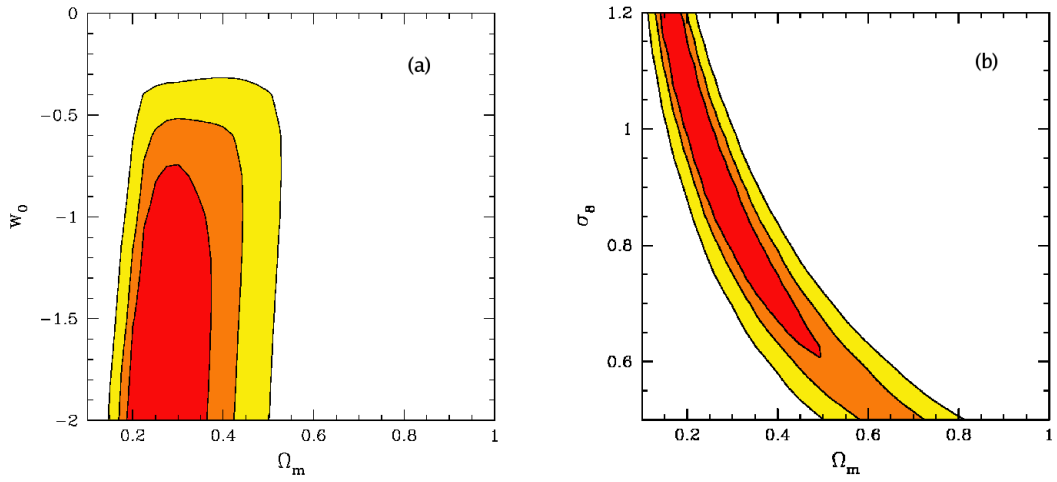

Figure 1.2: *(a): Dark energy constraints using the measurements from the W1 and W3 fields of CFHTLS. The contours indicate the 68.3%, 95.4%, and 99.7% confidence limits on two parameters jointly. (b): Join constrains on Ω_m and σ_8 (Hoekstra et al. 2006).*

Mass distribution over different scales in the Universe

The lensing cross-section being the probability that a source from a defined sample (galaxies, stars, quasars, etc...) will be multiple imaged or stretched beyond a given threshold, puts a strong constrains on the abundance of objects of different masses in the Universe. The lack of lensing systems without a visible lens proves that there is no significant amount of mass in compact dark objects. The number of lensing events predicted by theories is usually much higher then the one observed. However we do not yet, fully understand the selection effects and the impact that the substructure have on number of arcs. For the in-depth discussion on the lensing cross-section see Bartelmann (1995) or Meneghetti et al. (2003) and references within.

Lensing as a natural telescope

Gravitational lenses, act as a natural telescopes, giving us the possibility to study very faint and distant objects, that would otherwise be out of the reach of current instruments. A recent study of a $z = 4.87$ galaxy has been carried out by Swinbank et al. (2007) using the galaxy cluster RCS0224-0002 as a lens. The strong lensing modeling allowed for the source reconstruction, presented in Fig. 1.3, which in turn revealed interesting properties of this hight redshift galaxy. Its mass and size were determined, together with the velocity gradient. More interestingly, the spectral analysis showed that there is a galactic-scale bipolar outflow which has recently bursted out of the system. Such a detailed spectroscopic study of a galaxy at $z \approx 5$ would not be possible with today's class telescopes if strong lensing was not involved.

Dynamical history of clusters

The comparison of the lensing inferred dark matter maps with the luminous and X-ray component gives us the possibility to study the dynamical history of the lens (cluster). Two interesting examples of such a studies are the IE 0657-558 cluster (aka "The Bullet Cluster") (see e.g. Clowe et al. 2006; Bradač et al. 2006), and the Cl 0024+17 cluster (Jee et al. 2007). In both cases the discrepancy between the dark matter, X-rays and luminous matter distributions is observed. This is a hint for the collision (merger) of two or more

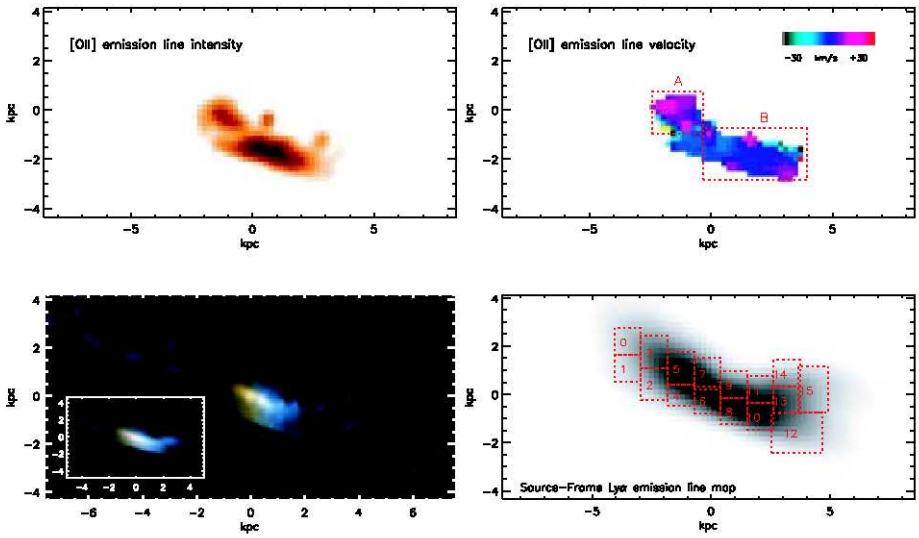

Figure 1.3: *Source-plane observations of the z = 4.88 galaxy from the HST, VIMOS and SINFONI observations. Top left-hand panel: [O II] emission-line intensity of the galaxy (dark regions represent regions of highest intensity). Top right-hand panel: [O II] emission-line velocity structure of the galaxy which shows a maximum velocity shift of* $60 \pm 20\ kms^{-1}$ *along the long axis of the galaxy. Bottom left-hand panel: reconstructed true clour HST VI image of the z = 4.88 arc. Inset: reconstructed HST image after a smoothing scale of 0.8 arcsec has been applied to the sky-plane image. Bottom right-hand panel: reconstructed* $Ly\alpha$ *emission-line map of the z = 4.88 arc (Swinbank et al. 2007).*

clusters. Its also one of the most direct proofs for the existences of the dark matter component in galaxy clusters. The Bullet Cluster (Fig. 1.4) is supposed to undergo an collision seen from the side, while the Cl 0024+17 (Fig. 2.2) is a merger seen face on. The significance of those results is however still controversial, and not widely accepted in the community.

1.2 Historical note

The possibility that gravity can interact with light was already noted by Sir Isaac Newton in his *Optics* (Newton 1704). In the first query Newton asks: "Do not Bodies act upon Light at a distance, and by their action bend its rays, and is not this

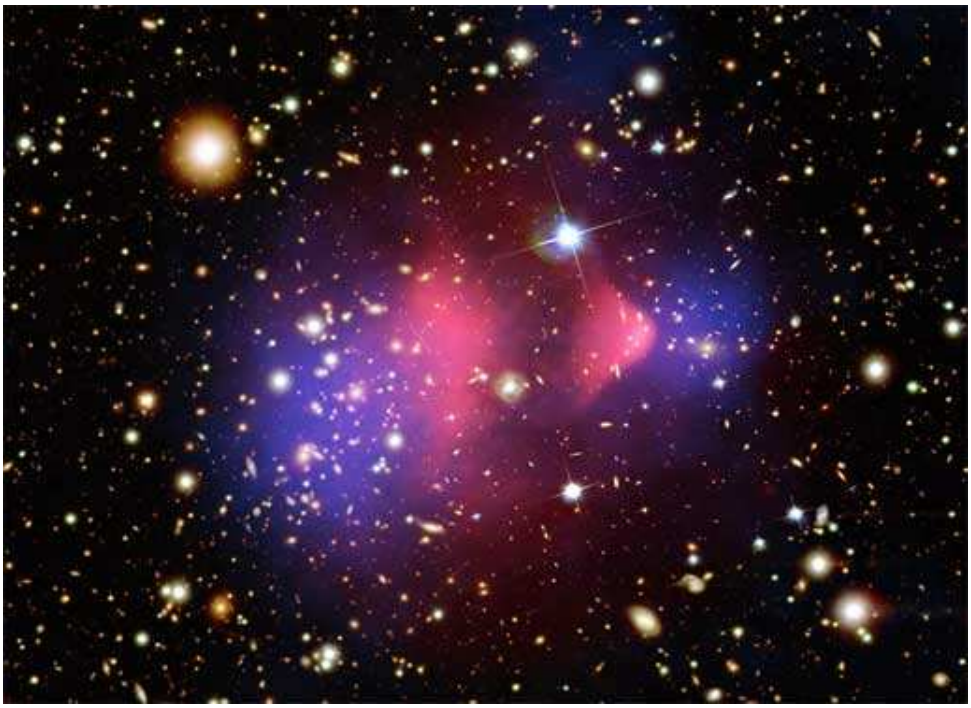

Figure 1.4: *The Bullet Cluster. The Dark Matter (blue) and X-ray emission (red) over plotted on the luminous component. Image credit: X-ray: NASA/CXC/CfA/M.Markevitch et al.; Optical: NASA/STScI; Magellan/U.Arizona/D.Clowe et al.; Lensing Map: NASA/STScI; ESO WFI; Magellan/U.Arizona/D.Clowe et al. .*

action (cteris paribus) strongest at the least distance?" However, for the nearly next 80 years the problem was not investigated any further. In 1783 John Michell discussed in the letter to Henry Cavendish, the existence of object so dense that the light can not escape their gravity. He also stated that those "black" objects could be detected by observation of the companion stars revolving around these invisible objects. At the beginning of 19th century, the Munich astronomer J. Soldner investigated the error in determination of the positions of starts due to the deflection of light by massive objects (Soldner 1804). The result he obtained, within the framework of Newtonian gravity, was that the light ray passing close to the surface of the Sun, would be deflected by the angle

$$\alpha \approx \frac{2GM_\odot}{c^2 R_\odot} \approx 0.83 \text{ arcsec} .$$

More than a century later, Einstein, unaware of Soldner work, derived tha same equation and noted that it would be important if astronomers tested it ("Es wäre dringend zu wünschen, daß sich Astronomen der hier aufgerollten Frage annähmen, auch wenn die im vorigen gegebenen Überlegungen ungengend fundiert oder gar abenteuerlich erscheinen sollten.", Einstein 1911). In 1913 Einstein contacted the director of the Mount Wilson Observatory, George Ellery Hale, and asked him if it would be possible to measure the positions of the stars close to the Sun in order to measure the deflection angle. Freundlich of the Royal Observatory in Berlin got sufficiently interested in Einstein ideas that he organized an expedition to Russian Crimea peninsula, where the total solar eclipse was expected in 1914. However, a few weeks after the expedition arrived the World War I broke off, and the scientists were arrested before they could carry out the observations.

After the completion of the General Relativity, Einstein was the first one to use it to derive the deflection angle α of a light ray passing at a distance r from the center of spherical object of mass M. He obtained the value

$$\alpha \approx \frac{4GM}{c^2 r} \ ,$$

which is twice bigger than the value resulting from the Newton's theory of gravity (Einstein 1915). For the light ray grazing the Sun, Einstein predicted the deflection angle of 1.74 arcsec. Confirmation of this prediction, with 20% accuracy, by Eddington and his group in 1919 during the solar eclipse was the second observational confirmation of General Relativity (the first one was the explanation of Mercury's perihelion shift) and it brought attention to Einstein and his new theory[2].

In the following decades the influence of gravity on light rays was not part of the mainstream research. Mainly because extragalactic astronomy was just beginning to develop and gravitational lensing effects within our Galaxy were beyond the reach of observational techniques (except for the already discussed effect of the displacement of the angular positions of stars by the Sun). However, some theoretical works has still

[2]In 1995 the value of the deflection angle predicted by General Relativity was confirmed, with radio-interferometric methods, to an accuracy better than 0.02% (Lebach et al. 1995)

been carried out. Eddington investigated the conditions under which multiple images of a lensed star (by another star) could appear. He also calculated (wrongly) the fluxes of the multiple images (Eddington 1920). The possibility of appearance of ring-shaped images (now days called *Einstein Rings*) was first discussed by Chwolson (1924). In this paper he considered the situation where a distant background star is lensed by a foreground star, what leads to formation (depending on the two stars alignment) of two images of the background star, one of witch might be very close to the foreground star – this would form a fictitious double star, which would not be resolved specially, but its spectrum should consist of a superposition of two (probably) different spectra. He noted that if the two stars are aligned perfectly, than the ring-like image will occur. After talking to a Czech engineer Mandl, Einstein published the results of his calculations of the star on star lensing (Einstein 1936). He was however, very skeptical about observational feasibility of this effect, and he noted that "there is no great chance of observing this phenomenon."

The breakthrough was done by Fritz Zwicky in 1937, when he published two papers in which he considered the galaxies ("extragalactic nebulae") as lenses (Zwicky 1937a,b). Those papers were truly insightful and described many of the most important properties and applications of gravitational lensing. Remarkably Zwicky noticed: "The discovery of images of nebulae which are formed through the gravitational fields of nearby nebulae would be of considerable interest for a number of reasons.

(1) It would furnish an additional test for the General theory of Relativity.

(2) It would enable us to see nebulae at distances greater than those ordinarily reached by even greatest telescopes. Any such *extension* of the known parts of the Universe promises tho throw very welcome new light on a number of cosmological problems.

(3) The problem of determining nebular masses at present has arrived at a stalemate... Observations of the deflection of light around nebulae may provide the most direct determination of nebular masses." All of those three points are nowadays the main focus of the gravitational lensing research.

Lack of observations resulted in not much interest of the researchers in gravitational lensing. This situation changed with the advent of radio astronomy and the discovery of quasars (Schmidt 1963). Quasars being a distant point like objects with strong emission

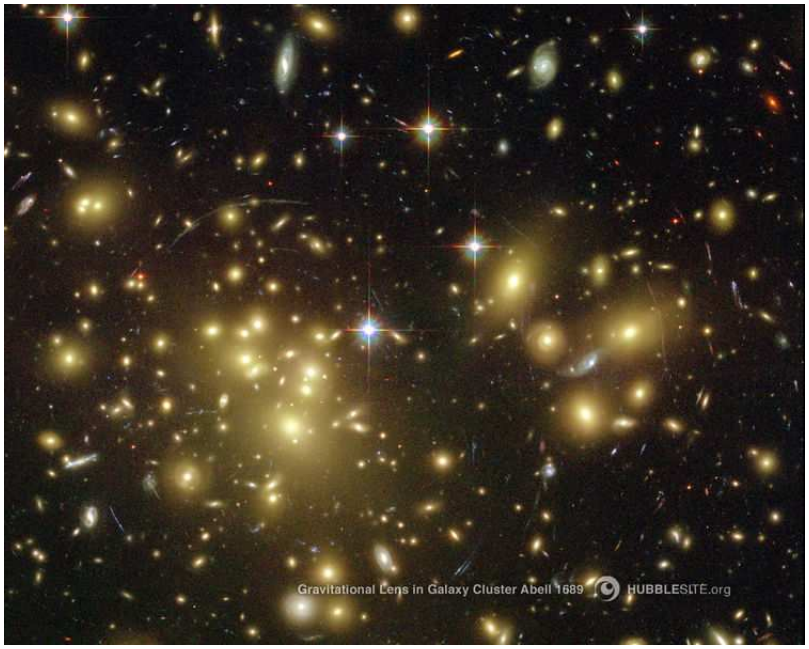

Figure 1.5: *The HST image of the galaxy cluster Abell 1689. It is the richest gravitational lens discovered up to date. Many giant luminous arcs and arclets are visible around the two cluster cores.*

lines and hight luminosity were an ideal source candidates for gravitational lensing. This new discovery coincided with further theoretical work in the subject. Liebes (1964) considered the lensing of stars in Andromeda galaxy by stars in the Milky Way, and Refsdal (1964b) derived the basic equations of the gravitational lens theory. Refsdal also gave a detailed description of the properties of point-like gravitational lens, and showed how a time dilatation between two images of one source can lead to the determination of the Hubble constant (Refsdal 1964a). At the same time Sachs (1961) and Kantowski (1969) studied the propagation of light ray bundles in inhomogeneous universe.

Finally, in 1979 the first observational confirmation of the gravitational lensing effect was done. In that year Walsh, Carswell and Weymann (1979) announced the detection of a double lensed quasar Q0957+561, with the source being at $z \approx 1.4$ and the lensing galaxy at $z = 0.36$. The two optical images were separated by approximately 6 arcseconds (Fig. 1.6). Seven years later the detection of giant luminous arcs was reported by Lynds and Petrosian (1986). Those new "objects" were described as located in the clusters of galaxies, elongated, narrow, arc-shaped features, with luminosities comparable

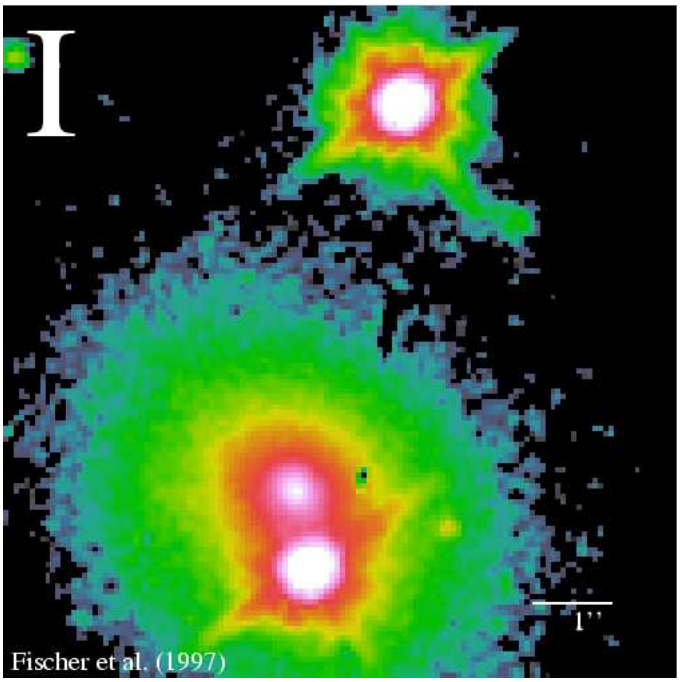

Figure 1.6: *The I band image of the double lensed quasar Q0957+561 (Fischer P. 1997). The two images are clearly visible (one to the north and one to the south), together with the lensing galaxy (above the souther image).*

to those of giant elliptical galaxies. From that point on gravitational lensing become one of the main-stream fields of astrophysical research, and one of the most important cosmological tools. In 1990 the first case of weak lensing was observed by Tyson, Wenk and Valdes (1990), and three years later Alcock et al. (1993) reported the detection of the first microlensing event. Now there are many gravitational lens systems known, and each year new ones are discovered. I will mention here two of them – Abell 1689 and J1004+411. Abell 1689 is the galaxy cluster with the most lensing features up to date Fig 1.5. Broadhurst et al. (2005) identified more than 100 extended images of more than 30 sources in the.field of this cluster. J01004+411 on the other hand has both the multiple images of a normal background galaxy and multiple images of a background quasar (Sharon et al. 2005).

1.3 Physics behind lensing

Inthis section we use the notation where the repetitive indexes on different levels are summed over, and a comma followed by an index denotes a partial differentiation with respect to the variable labeled by that index. Also, we use the orthogonal coordinates $x^0 = ct$, $\boldsymbol{x} = (x^i)$. In General Relativity light rays travel along null-geodesics, which are determined by the underlying metric $g_{\alpha\beta}$. The metric is, in turn, defined by the distribution of energy in the Universe through Einstein gravitational field equation

$$R^{\alpha\beta} - \frac{1}{2}Rg^{\alpha\beta} = \frac{8\pi G}{c^4}T^{\alpha\beta} \ , \tag{1.2}$$

where $R^{\alpha\beta}$ is the Ricci tensor with the trace R, $T^{\alpha\beta}$ is the stress-energy tensor, c is the speed of light in vacuum, and G is the gravitational constant. The Ricci tensor is obtained by contracting the Riemann tensor

$$R_{\alpha\beta} = R^{\gamma}{}_{\alpha\gamma\beta} \ , \tag{1.3}$$

which is defined by connection coefficients (Christoffel symbols)

$$R^{mu}{}_{\alpha\beta\gamma} = \Gamma^{\mu,\beta}{}_{\alpha\gamma} - \Gamma^{\mu,\gamma}{}_{\alpha\beta} + \Gamma^{\mu}{}_{\sigma\beta}\Gamma^{\sigma}{}_{\gamma\alpha} - \Gamma^{\mu}{}_{\sigma\gamma}\Gamma^{\sigma}{}_{\beta\alpha} \ . \tag{1.4}$$

And finally, the connection coefficients can be expressed in terms of the metric

$$\Gamma^{\alpha}{}_{\beta\gamma} = \frac{1}{2}g^{\alpha\delta}(g_{\delta\beta,\gamma} + g_{\delta\gamma,\beta} - g_{\beta\gamma,\delta}) \ . \tag{1.5}$$

Therefore, by defining the distribution of mass (energy) in the space through the stress-energy tensor $T^{\alpha\beta}$, it is possible to use the Einstein equation (1.2) to calculate the metric, which is enough to describe gravitational lensing effect. In practice the solution of the field equation is possible only for very simple forms of the stress-energy tensor. If we assume that the space is empty everywhere, except for a small local concentration of matter (energy), then we can linearize the filed equation (1.2). The empty space is described by a flat Minkowski metric $\eta_{\alpha\beta} = \mathrm{diag}(1, -1, -1, -1)$. We can write the metric $g_{\alpha\beta}$ in the form of

$$g_{\alpha\beta} = \left(1 - \frac{1}{2}\right)\eta_{\alpha\beta} + h_{\alpha\beta} \ , \tag{1.6}$$

$$h = \eta^{\alpha\beta} h_{\alpha\beta} \ . \tag{1.7}$$

Assumption that the space is empty everywhere, except for a small local over-density of energy (weak field approximation) means that $\|h_{\alpha\beta}\| \ll 1$. Due to the symmetries (energy conservation and Bianchi identity see e. g. Peacock 1999, chap. 1) the Einstein field equation (1.2) has four degrees of freedom, and therefore we are free to choose a Lorenz gouge

$$h^{\alpha\beta}{}_{,\beta} = 0 \ . \tag{1.8}$$

Applying (1.6) to the (1.2) and noticing that in the weak field approximation we can use the background Minkowski metric $\eta_{\alpha\beta}$, instead of the full $g_{\alpha\beta}$ metric, to transform the coordinates from covariant to contravariant, results in the linearized form of the field equation

$$\left(\triangle - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\right) h^{\alpha\beta} = \frac{16\pi G}{c^4} T^{\alpha\beta} \ . \tag{1.9}$$

This is a wave equation that has a retarded solution in the form of

$$h^{\alpha\beta}(t, \boldsymbol{x}) = -\frac{4G}{c^4} \int \frac{T^{\alpha\beta}\left(t - \frac{|\boldsymbol{y}|}{c}, \boldsymbol{x} + \boldsymbol{y}\right)}{|\boldsymbol{y}|} d^3y \ . \tag{1.10}$$

To proceed further some form of the stress-energy tensor needs to be assumed. For most astrophysical cases it is enough to model the matter as a perfect fluid, and write $T^{\alpha\beta}$, as

$$T^{\alpha\beta} = \left(\rho c^2 + p\right) u^\alpha u^\beta - p g^{\alpha\beta} \ . \tag{1.11}$$

Here ρ donates the matter density and p the pressure (both in the comoving reference frame), and $u^\alpha = \gamma(c, \boldsymbol{v})$ is the four-velocity, normalized to one $u_\alpha u^\alpha = 1$. If we further

assume that matter moves slowly in the x^α coordinate system, i. e. $v^i := \frac{\mathrm{d}x^i}{\mathrm{d}t}$ obeys $|\boldsymbol{v}| < c$, and $|p| \ll \rho c^2$, then the stress-energy tensor simplifies to

$$T^{00} = \rho c^2 \ , \ T^{0i} = c\rho v^i \ , \ T^{ij} = \rho v^i v^j + p\delta^{ij} \ , \tag{1.12}$$

where the terms of order v^2/c^2 and $p/(\rho c^2)$ have been omitted. By applying those approximations and by introducing the retarded potentials

$$U(t, \boldsymbol{x}) = -G \int \frac{\rho\left(t - \frac{|\boldsymbol{y}|}{c^2}\,,\, \boldsymbol{x} + \boldsymbol{y}\right)}{|\boldsymbol{y}|} d^3y \ , \tag{1.13}$$

$$\boldsymbol{V}(t, \boldsymbol{x}) = -G \int \frac{\rho\boldsymbol{v}\left(t - \frac{|\boldsymbol{y}|}{c^2}\,,\, \boldsymbol{x} + \boldsymbol{y}\right)}{|\boldsymbol{y}|} d^3y \ , \tag{1.14}$$

we can calculate the metric and the length element $\mathrm{d}s$ from (1.10) and (1.6)

$$ds^2 = g_{\alpha\beta} dx^\alpha dx^\beta = \left(1 + \frac{2U}{c^2}\right) c^2 dt^2 - 8cdt \frac{\boldsymbol{V} \cdot d\boldsymbol{x}}{c^3} - \left(1 - \frac{2U}{c^2}\right) dx^2 \ . \tag{1.15}$$

The weak filed approximation holds if and only if $|U| \ll c^2$, which also implies $\left|\frac{\boldsymbol{V}}{c^3}\right| \leq \left|\frac{\boldsymbol{v}}{c}\right| \left|\frac{\boldsymbol{U}}{c^2}\right| \ll 1$. This allows us to write the metric (1.15) in the form of

$$ds^2 = e^{\frac{2U}{c^2}} \left(cdt - \frac{4V_i}{c^3} dx^i\right)^2 - e^{\frac{-2U}{c^2}} dx^2 \ . \tag{1.16}$$

Since light rays travel along null curves $ds^2 = 0$, then

$$cdt = e^{\frac{-2U}{c^2}} dl + \frac{4V_i}{c^3} dx^i \ , \tag{1.17}$$

where dl is the length element $dl = |d\boldsymbol{x}|$. If the light emitted at $t = 0$ travels along the null path γ parameterized by λ ($x^i = x^i(\lambda)$, then the arrival time is

$$t = \int_\gamma \left(e^{\frac{-2U}{c^2}} dl + \frac{4V_i}{c^3} dx^i\right) \ . \tag{1.18}$$

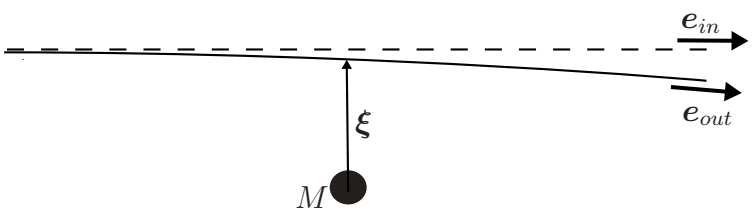

Figure 1.7: *Deflection of the light ray, passing by a mass M, with the impact vector $\boldsymbol{\xi}$. The deflection angle is defined as $\hat{\boldsymbol{\alpha}} = \boldsymbol{e}_{in} - \boldsymbol{e}_{out}$.*

Light will travel the spatial path given by the Fermat's principle

$$\delta \int_{\gamma} \left(e^{\frac{-2U}{c^2}} dl + \frac{4V_i}{c^3} dx^i \right) = 0 \, . \tag{1.19}$$

This means that, in the weak field approximation, gravity acts as a "medium" with a effective index of refraction n

$$n = 1 - \frac{2U}{c^2} + \frac{4}{c^3} \boldsymbol{V} \cdot \boldsymbol{e} \, , \tag{1.20}$$

where $\boldsymbol{e} = \frac{d\boldsymbol{x}}{dl}$. By applying the Euler-Lagrange equation to the variational principle $\int_{\gamma} n dl = 0$ we find the spatial path of the light rays

$$\frac{d\boldsymbol{e}}{dl} = \frac{-2}{c^2} \left(\nabla U - \boldsymbol{e}(\boldsymbol{e} \cdot \nabla U) \right) + \frac{4}{c^3} \boldsymbol{e} \times (\nabla \times \boldsymbol{V}) \, . \tag{1.21}$$

The first term in Eq (1.21), $\frac{-2}{c^2} \left(\nabla U - \boldsymbol{e}(\boldsymbol{e} \cdot \nabla U) \right) = \nabla_{\perp} U$, is the projection of the ∇U onto a plane perpendicular to the direction $\boldsymbol{e}$ of the light ray propagation, and represents an attractive "force" towards the deflecting mass. The second term, $\frac{4}{c^3} \boldsymbol{e} \times (\nabla \times \boldsymbol{V})$, is due to the gravitomagnetic field produced by moving matter, it is related to the famous "dragging of inertial frames" effect, which has been lately measured by Ciufolini et al. (2007), and is an important test for general relativity. The second term is however much smaller than the first one, and therefore we will neglect it in our further considerations. If we define the defection angle $\hat{\boldsymbol{\alpha}}$ as

$$\hat{\boldsymbol{\alpha}} := \boldsymbol{e}_{in} - \boldsymbol{e}_{out} \, , \tag{1.22}$$

then from (1.21) we obtain

$$\hat{\boldsymbol{\alpha}} = \frac{2}{c^2}\int (\nabla U - \boldsymbol{e}(\boldsymbol{e}\cdot\nabla U))\, dl\ . \tag{1.23}$$

Since in most astrophysical situations the deflection angles are small, we can integrate Eq (1.23) along the unperturbed path $\boldsymbol{x}(l) = l\boldsymbol{e}$, instead of the real one (see Fig. 1.7). Also, the light ray undergoes most of the deflection in the small region around the minimum distance between the ray and the deflecting mass. Those simplifications allow us to calculate the deflection angle for a point mass, with the potential $U(\boldsymbol{x}) = -\frac{GM}{|\boldsymbol{x}|}$,

$$\hat{\boldsymbol{\alpha}} = \frac{4GM}{c^2}\frac{\boldsymbol{\xi}}{|\boldsymbol{\xi}|^2}\ . \tag{1.24}$$

If the deflecting mass in not point like but has some spatial distribution, then (in addition to small $\hat{\boldsymbol{\alpha}}$), we have to assume that the extent of mass in the direction of the incoming ray is so small that the $\nabla_\perp U$ along the unperturbed ray deviates little from that on the actual one. This assumption holds very well since in all the astrophysical cases the distance from the source to the lens and from the lens to the observer is orders of magnitude bigger than the size of the deflecting mass distribution. Under those assumptions we can again integrate along the unperturbed path, and since all the mass elements along the incoming ray direction have the same impact parameters $\boldsymbol{\xi}$, then we can project them onto one plane (so called *lens plane*) and use their surface mass density $\Sigma(\boldsymbol{\xi})$ to calculate the deflection angle. This is the *thin lens approximation*. The resulting deflection angle will be a sum of "point mass" like deflection angles form each element of the lens plane

$$\hat{\boldsymbol{\alpha}} = \frac{4G}{c^2}\int_{\mathcal{R}^2}\frac{(\boldsymbol{\xi}-\boldsymbol{\xi}')\Sigma(\boldsymbol{\xi}')}{|\boldsymbol{\xi}-\boldsymbol{\xi}'|}d^2\xi'\ . \tag{1.25}$$

The integral in the above equation is carried out in the lens plane and $\boldsymbol{\xi}$ is a two-dimensional vector in that plane. Eq. (1.25) gives the value of a deflection angle for a light ray passing by an arbitrary (within the above assumptions) mass density distribution, and is the basic equation of the gravitational lensing theory.

Since in the presence of the deflecting mass light travels a different trajectory then it would in absence of it, an interesting effect of a time delay arises. The time it takes the light ray to travel from the source to the observer will be longer in the presence of the lensing effect, due to two factors. The first one is purely geometrical - the path SLO is longer than the direct, unperturbed path SO (see Fig. 1.8). The second one is due to time dilatation in the presence of gravitational filed - the Shapiro Effect (Shapiro 1964). From Eq. (1.18), after neglecting the movement of the matter, we get the time needed for a light ray traveling along path SLO to reach the observer

$$t = c^{-1} \int \left(1 - \frac{2U}{c^2}\right) = c^{-1} l - 2c^{-3} \int U dl \ , \tag{1.26}$$

where l is the length of the path SLO. In the thin lens approximation one can substitute the deflected light ray with the asymptotic path - light travels along a straight line from source to the lens plane SL, then it undergoes the rapid, non smooth, deflection , and then it again travels in a straight line to the observer LO. From Fig 1.8 one can calculate the length of the deflected light path l

$$l = \sqrt{(\boldsymbol{\xi} - \boldsymbol{\eta})^2 + D_{ls}^2} + \sqrt{\boldsymbol{\xi}^2 + D_l^2} \tag{1.27}$$

$$\approx D_{ls} + D_l + \frac{1}{2D_{ls}}(\boldsymbol{\xi} - \boldsymbol{\eta})^2 + \frac{1}{2D_l}\boldsymbol{\xi}^2 \ , \tag{1.28}$$

where the D_l, D_s, and D_{ls} denote angular distances from the observer to the lens plane, from the observer to the source, and from the lens plane to the source, respective. The unperturbed direct path from the source to the observer has the length of $l_0 \approx D_s + \frac{1}{2D_s}\boldsymbol{\eta}^2$. Therefore, the geometrical time delay caused by lensing is

$$c\Delta t = l - l_0 = \frac{D_l D_s}{2D_{ls}} \left(\frac{\boldsymbol{\xi}}{D_l} - \frac{\boldsymbol{\eta}}{D_s}\right)^2 + \text{const.}. \tag{1.29}$$

To calculate the time delay in the Eq. (1.26) caused by Shapiro effect, we need to calculate the integral of the potential U along the path SLO. First we assume the point mass potential and calculate the integral along the partial path SL is

$$\int_{SL} Udl = GM \left(\ln \frac{|\boldsymbol{\xi}|}{D_{ls}} + \frac{\boldsymbol{\xi}(\boldsymbol{\eta} - \boldsymbol{\xi})}{D_{ls}^2} + \frac{|\boldsymbol{\xi}| - |\boldsymbol{\eta}|}{D_{ls}^2} \right) , \tag{1.30}$$

since $D_{ls} \gg |\boldsymbol{\xi}|$ then we can use only the first term, which does not depend on $\boldsymbol{\eta}$

$$\int_{SL} Udl = GM \left(\ln \frac{|\boldsymbol{\xi}|}{D_{ls}} \right) = GM \ln \frac{|\boldsymbol{\xi}|}{\xi_0} + \text{const} , \tag{1.31}$$

where ξ_0 is an arbitrary length scale in the lens plane. The integral along the full path SLO is , within precision of an additive constant, equal to two times the partial integral

$$\int_{SLO} Udl = 2GM \ln \frac{|\boldsymbol{\xi}|}{\xi_0} + \text{const} . \tag{1.32}$$

Using the linearity of U in the mass distribution we can write the time delay caused by Shapiro effect in the form of

$$\frac{-2}{c^3} \int Udl = \frac{-4G}{c^3} \int \Sigma(\boldsymbol{\xi}') \ln \left(\frac{|\boldsymbol{\xi} - \boldsymbol{\xi}'|}{\xi_0} \right) + \text{const} , \tag{1.33}$$

where $\Sigma(\boldsymbol{\xi}')$ is the surface mass density in the source plane. Adding the two effects together we get an expression for the time delay of a lensed light ray in respect to the direct one

$$c\Delta t = \hat{\phi}(\boldsymbol{\xi}, \boldsymbol{\eta}) + \text{const} , \tag{1.34}$$

where $\hat{\phi}(\boldsymbol{\xi}, \boldsymbol{\eta})$ is called the *Fermat potential* and it is given by

$$\hat{\phi}(\boldsymbol{\xi}, \boldsymbol{\eta}) = \frac{D_l D_s}{2D_{ls}} \left(\frac{\boldsymbol{\xi}}{D_l} - \frac{\boldsymbol{\eta}}{D_s} \right)^2 - \hat{\psi}(\boldsymbol{\xi}) , \tag{1.35}$$

and the *deflection potential* $\hat{\psi}(\boldsymbol{\xi})$ is defined as

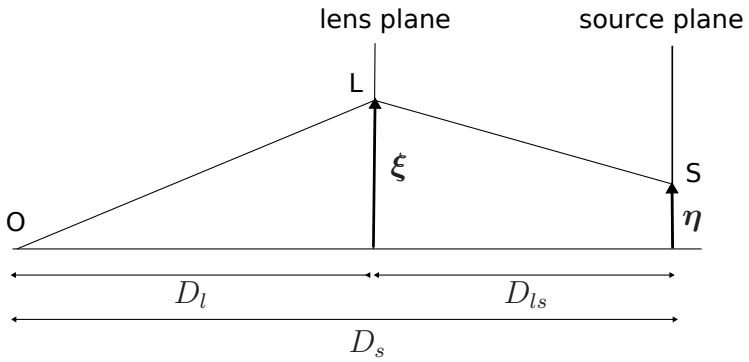

Figure 1.8: *The basic geometry of gravitational lensing. The light emitted by a source at the position $\boldsymbol{\eta}$ in the source plane, travels along the path SL to the lens plane, where it is deflected at the position $\boldsymbol{\xi}$ and then travels along the path LO to the observer.*

$$\hat{\psi}(\boldsymbol{\xi}) = \frac{4G}{c^2} \int \Sigma(\boldsymbol{\xi}') \ln\left(\frac{|\boldsymbol{\xi} - \boldsymbol{\xi}'|}{\xi_0}\right) . \tag{1.36}$$

According to the Fermat's principle, the actual light ray path is the one, for which the arrival time is stationary in respect to the variation of the deflection point $\boldsymbol{\xi}$

$$\frac{\partial(c\Delta t)}{\partial \boldsymbol{\xi}} = \nabla_{\boldsymbol{\xi}} \hat{\phi}(\boldsymbol{\xi}, \boldsymbol{\eta}) = 0 . \tag{1.37}$$

The above condition is fulfilled if

$$\boldsymbol{\eta} = \frac{D_s}{D_l}\boldsymbol{\xi} - D_{ls}\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi}) . \tag{1.38}$$

The obtained equation is called *lens mapping*, and it is a basic equation in gravitational lensing theory, that binds the position of the source $\boldsymbol{\eta}$, the position of the image $\boldsymbol{\xi}$, and the properties of the lens $\hat{\boldsymbol{\alpha}}(\boldsymbol{\xi})$. It is important to note that given a source position one needs to invert this equation to obtain the position of the image. This inverted equation might have more than one solution in which case lensing leads to the appearance of multiple images on one source. Also in general case of surface mass density distribution $\Sigma(\boldsymbol{\xi}')$ in the lens plane, it is impossible to find solutions to the inverted lens mapping equation analytically.

1.4 Basic Lensing Theory

The lensing mapping equation (1.38) obtained in the previous section, is the starting point for the discussion om many interesting properties of gravitational lensing. We begin by introducing the scaled variables

$$\boldsymbol{x} = \frac{\boldsymbol{\xi}}{\xi_0} \,, \tag{1.39}$$

$$\boldsymbol{y} = \frac{\boldsymbol{\eta} D_l}{\xi_0 D_s} \,, \tag{1.40}$$

together with the dimensionless surface mass density

$$\kappa(\boldsymbol{x}) = \frac{\Sigma(\xi_0 \boldsymbol{x})}{\Sigma_{cr}} \,, \tag{1.41}$$

where Σ_{cr} is a *critical surface mass density* defined as

$$\Sigma_{cr} = \frac{c^2 D_s}{4\pi G D_l D_{ls}} \,. \tag{1.42}$$

If we also introduce the scaled deflection angle and the scaled deflection potential

$$\boldsymbol{\alpha}(\boldsymbol{x}) = \frac{D_l D_{ls}}{\xi_0 D_s} \hat{\boldsymbol{\alpha}}(\xi_0 \boldsymbol{x}) = \frac{1}{\pi} \int_{\mathcal{R}^2} \kappa(\boldsymbol{x}') \frac{\boldsymbol{x} - \boldsymbol{x}'}{|\boldsymbol{x} - \boldsymbol{x}'|^2} d^2 x' \,, \tag{1.43}$$

$$\psi(\boldsymbol{x}) = \frac{D_l D_{ls}}{D_s \xi_0^2} \hat{\psi}(\xi_0 \boldsymbol{x}) = \frac{1}{\pi} \int_{\mathcal{R}^2} \kappa(\boldsymbol{x}') \ln(|\boldsymbol{x} - \boldsymbol{x}'|) d^2 x' \,, \tag{1.44}$$

then we can rewrite the lensing mapping equation (1.38) in a simple form of

$$\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{\alpha}(\boldsymbol{x}) \,. \tag{1.45}$$

The lensing equation (1.45) provides a mapping from the image (lens) plane to the source plane. This mapping is characterized by the Jacobian matrix

$$\mathcal{A}_{ij}(\boldsymbol{x}) = \frac{\partial y_i(\boldsymbol{x})}{\partial x_j} = \left(\delta_{ij} - \frac{\partial^2 \psi(\boldsymbol{x})}{\partial x_i \partial x_j} \right) = \begin{bmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{bmatrix} \,, \tag{1.46}$$

where we have introduced the components of the complex *shear* $\gamma = \gamma_1 + \mathrm{i}\gamma_2 = |\gamma|\mathrm{e}^{2\mathrm{i}\varphi}$,

$$\gamma_1 = \frac{1}{2}\left(\frac{\partial^2\psi(\boldsymbol{x})}{\partial^2 x_1} - \frac{\partial^2\psi(\boldsymbol{x})}{\partial^2 x_2}\right) , \qquad \gamma_2 = \frac{\partial^2\psi(\boldsymbol{x})}{\partial x_1 \partial x_2} , \tag{1.47}$$

and the *convergence* κ has been already defined by Eq. (1.41). We list here some properties of this matrix, that will be useful later:

$$\mathrm{det}\mathcal{A} = (1-\kappa)^2 - |\gamma|^2 , \tag{1.48}$$

$$\mathrm{tr}\mathcal{A} = 2(1-\kappa) , \tag{1.49}$$

$$a_{1,2} = 1 - \kappa \mp |\gamma| , \tag{1.50}$$

$$\tag{1.51}$$

where $a_{1,2}$ are the eigenvalues of $\mathcal{A}$.

Because of the Liouville's theorem and the lack of production/absorption of photons during the process of gravitational lensing, the surface brightness is conserved i. e. the resolved images have the same surface brightness as the unlensed source. The flux is however not conserved. If in the absence of lensing the source occupies d^2y portion of the sky, and has the surface brightness of I, then the flux S_0, of this unlensed source is $S_0 = Id^2y$. In the presence of lens between the source and the observer, the source will be starched to the size of d^2x, and thus its flux will be $S = Id^2x$. Assuming d^2x is small compared to the scale on which the properties of the lens change, we can use the Jacobian matrix $\mathcal{A}$ of the mapping (1.45) to calculate the distortion caused by lensing

$$d^2y = |\mathrm{det}\mathcal{A}(\boldsymbol{x})|d^2x . \tag{1.52}$$

If we define the *magnification* μ as S/S_0, then from Eq. (1.52), we get

$$\mu(\boldsymbol{x}) = \frac{1}{\mathrm{det}\mathcal{A}(\boldsymbol{x})} . \tag{1.53}$$

Therefore the flux of the source is increased/decreased by a factor of $|\mu(\boldsymbol{x})|$ and the sign of the Jacobian gives the information about the parity of the image. The magnification of the resolved images is a weighted mean of the magnification across the image area

$$\mu = \left[\int I(\boldsymbol{y}) d^2y\right]^{-1} \int I(\boldsymbol{x})\mu(\boldsymbol{x}) d^2x \ , \tag{1.54}$$

however the size of the source in usually not known, therefore one can only obtain relative magnifications between multiple images of a given source. It is possible that $\det\mathcal{A}(\boldsymbol{x}) = 0$, the points in the lens plane for which this occurs form smooth, closed curves, called *critical curves*. If we map those curves back to the source plane using Eq. (1.45) then we obtain closed, but not necessary smooth curves in the source plane called *caustics*.

If the source is point-like and located on the caustic, then it will produce image(s) close to the critical curves with infinite magnification. In reality the infinite magnification does not occur because of two reasons 1) All physical sources are not point-like and then Eq. (1.54) leads to finite magnification. 2) Close to the critical curves the geometrical approach to light propagation in space, used to derive the mapping equation (1.45), brakes down and the proper treatment of the problem within the framework of wave optics is needed. Nevertheless if the source is on (or close) to the caustic, then its image(s) will be highly magnified and located close to the critical curve.

The lens mapping equation is invertible everywhere except on critical curves. Therefore the only possibility to increase/decrease number of images of a given source is to move this source through a caustic. This will change the number of images by ± 2 (because wavefronts are continues), depending on the direction of crossing. If we assume a lens with a smooth potential and a deflection angle that goes to 0 as we move to a large distance from the lens center (which is the case for all real gravitational lenses), then we can conclude that the number of images produced by the lens is always odd. Indeed if the source is far away from the center of the lens then only one image will be produced (since deflection angle approaches 0), then if we start moving the source closer to the center of the lens, each time we cross the caustic the number of images will change by ± 2, so the total number of images is always odd. Moreover the images produced by crossing the caustic will have different parities, since they will appear on the opposite sides of the critical curves and the magnification Eq. (1.53) changes sign when we cross the critical curve.

If we assume that the source is not on the caustic, and $\boldsymbol{X}$ and $\boldsymbol{W}$ are two displacement vectors in the source plane, center of the source, then we can define the handedness of the source as the sign of $\boldsymbol{X} \times \boldsymbol{W}$. The handedness of the image of this source will be then the sign of $\boldsymbol{Y} \times \boldsymbol{Z}$, where $\boldsymbol{Y} = \mathcal{A}\boldsymbol{X}$,and $\boldsymbol{Z} = \mathcal{A}\boldsymbol{W}$. Since

$$\boldsymbol{Y} \times \boldsymbol{Z} = \mathcal{A}\boldsymbol{X} \times \mathcal{A}\boldsymbol{W} = \det \left[\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} X_1 & W_1 \\ X_2 & W_2 \end{pmatrix} \right] \hat{\boldsymbol{k}} = \det\mathcal{A}\ \boldsymbol{X} \times \boldsymbol{W} \ , \tag{1.55}$$

then we indeed see that negative $\det\mathcal{A}$ will result in the change of the parity of the image in respect to the parity of the source.

Not only the parity of the images but also their shapes are locally distorted by lensing. To see that, let us consider the circular source centered at the position $\boldsymbol{y}$, and with radius R, bounded by the curve $\boldsymbol{c}(t)$,

$$\boldsymbol{c}(t) = \boldsymbol{y} + R(\cos t, \sin t) \ . \tag{1.56}$$

If R is small compared to the scale on which the lensing properties change, then the corresponding bounding curve $\boldsymbol{d}(t)$ in the image plane will be

$$\boldsymbol{d}(t) = \boldsymbol{x} + \mathcal{A}^{-1} R(\cos t, \sin t) \ . \tag{1.57}$$

Applying Eq. (1.46) leads to the conclusion that the resulting image is an ellipse centers on $\boldsymbol{x}$ with semi-axis of length

$$\Lambda_{\pm} = \frac{R}{|1 - \kappa \mp |\gamma||} \ , \tag{1.58}$$

parallel to the principal axes of $\mathcal{A}$, given by the position angles

$$\tan\varphi_{\pm} = \frac{\pm|\gamma| - \gamma_1}{\gamma_2} \ . \tag{1.59}$$

If the source is located close to the caustic, then at least one of the eigenvalues of $\mathcal{A}$ is close to 0, which will lead to very hight stretching of the image in the direction of corresponding eigenvector. This explains the appearance of giant luminous arcs close to the critical curves. As expected, the area of the image is bigger then the area of the source by a factor $|\Lambda_-\Lambda_+| = |\mathrm{det}\mathcal{A}|^{-1} = |\mu|$.

One of the most important applications of gravitational lensing is the determination of the mass profile of the lens, based on shapes, relative brightness and positions of the multiple images. There is however a fundamental degeneracy that allows us to determine this profile only up to an additive constant. This is called the *mass-sheet degeneracy*. Indeed, lets assume we have found a mass density distribution $\kappa(\boldsymbol{x})$, of the lens, that is able to perfectly reproduce all the features of the observed images. If we substitute this mass density with a one of the form

$$\kappa_\lambda(\boldsymbol{x}) = (1-\lambda) + \lambda\kappa(\boldsymbol{x}) \ , \tag{1.60}$$

where λ is an arbitrary constant, then the new lensing equation will become

$$\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{\alpha}_\lambda(\boldsymbol{x}) \ , \tag{1.61}$$

with

$$\boldsymbol{\alpha}_\lambda(\boldsymbol{x}) = (1-\lambda)\boldsymbol{x} + \lambda\boldsymbol{\alpha}(\boldsymbol{x}) \ , \tag{1.62}$$

which results from Eq. (1.43) and $\boldsymbol{\alpha}(\boldsymbol{x})$ is a deflection angle caused by the $\kappa(\boldsymbol{x})$ mass distribution. Combining the Eq. (1.61) and Eq. (1.62), leads to the equation

$$\frac{\boldsymbol{y}}{\lambda} = \boldsymbol{x} - \boldsymbol{\alpha}(\boldsymbol{x}) \ . \tag{1.63}$$

The above relation between $\boldsymbol{y}$ and $\boldsymbol{x}$ is identical to the lensing equation of a lens with mass density profile $\kappa(\boldsymbol{x})$, with the exception of the length in the source plane being scaled by a factor λ^{-1}, which leads to the scaling of the magnification by a factor λ^{-2}.

In general however, the sizes in the source plane and fluxes of undistorted sources are not directly observed. Therefore, if nothing sets an absolute scale in the source plane (size of flux) and there are no indicators of the total mass of the lens (dynamics, X-ray, etc..), then it is impossible to distinguish the lens with mass density profile κ and κ_λ. The work around this problem might be the use of the fundamental plane of elliptical galaxies to fix the fluxes of lensed galaxies, as proposed by Bertin and Lombardi (2006). Another possibility arises if the source has flux that varies in time, the measurement of the time delays between images can lead to the breaking of the mass-sheet degeneracy. Indeed, if we consider the scaled Fermat potential (with $\xi_0 = D_l$)

$$\phi(\boldsymbol{x}, \boldsymbol{y}) = \frac{D_{ls}}{D_l D_s}\hat{\phi}(\boldsymbol{x}D_l, \boldsymbol{y}D_s) = \frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{y}\right)^2 - \psi(\boldsymbol{x}) \ , \tag{1.64}$$

then we notice that with a transformation $\kappa \to \kappa_\lambda$ it scales as

$$\phi_\lambda(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{y})^2 - \psi_\lambda(\boldsymbol{x}) = \lambda\phi(\boldsymbol{x}, \boldsymbol{y}/\lambda) + \text{const} \ . \tag{1.65}$$

Since the Fermat potential is proportional to the time delay between images, then if the source is variable in time, it is possible to break the mass-sheet degeneracy by measuring those time delays. Another possible way to break the mass-sheet degeneracy is when there is more then one source being lensed and the sources have different redshift, then since κ depends on the redshift of the source (through Σ_{cr}) the degeneracy is broken.

Chapter 2

Computational methods

It is unworthy of excellent men to lose hours like slaves in the labour of calculation which could safely be relegated to anyone else if machines were used.

Gottfried Wilhelm Leibniz

Abstract

In this chapter we introduce two parametric lens models – Non-Singular Isothermal Ellipsoid, and NFW profile. Those profiles are basic building blocks out of which one can create more complex lenses. We also introduce the three different goodness of fit functions used to determine how well the lens model reproduces the observed strong lensing features.

2.1 Introduction

A typical problem in gravitational lensing analysis is determination of the mass density distribution of the lens (galaxy, galaxy cluster, etc...) based on the observed features, such as the positions and shapes of arcs, their fluxes, and if available the time delays between different images. The mass measurement of various astrophysical objects based on gravitational lensing is very important, and often superior to other methods (like Sunyev-Zel'dovich effect, galaxy dynamics or X-ray temperature) because it probes directly the total mass content of a given object and it does not need any assumptions about its state or composition (see e.g. *ESA-ESO Working Groups, Report No. 3*, Peacock and Schneider 2006).

For any realistic case the analytical solution for $\kappa(\boldsymbol{x})$ of the set of lensing equations (one for each image) (1.45) does not exist, therefore finding appropriate mass distribu-

tion is based on numerical methods. The usual method is to parameterize the mass distribution (the lens model) with a set of parameters, and then change those parameters using one of the minimization algorithms, in such a way that a predefined distance (or goodness of fit) between the observed arcs and those produced by the model becomes minimal.

There are two major approaches to the parameterization of the lens mass density. The first one is based on the assumption that the mass distribution is approximated by the sum of components that have a straight physical interpretation, like the isothermal profile or the NFW (Navarro et al. 1996) profile. Each component in this approach is assigned a number of free parameters (scale, position of the center, ellipticity, position angle etc...) and then those parameters are changed to reproduce the observed lensing features. This approach is often refereed in the literature as the *parametric* one. The other method is to discretise the lens plane into a grid of pixels of often variable size, and then to assign a single parameter (the mass) to each of those pixels. The advantage of this "non-parametric" method is that we do not assume a priori any profile the mass distribution should follow, as it is th case with the parametric approach. The disadvantage is that the number of "parameters" – pixels – is much higher then in the first method. This is more computationally demanding and what is more important often leads to the problem of mass distribution finding being heavily under constrained (the number of constrains we get from the observations is much smaller then the number of parameters). The "non-parametric" algorithms also suffer from a problem of finding non-physical solutions. In the following subsection, the commonly used inversion techniques will be presented.

2.1.1 Inversion Methods

Generally, inversion techniques can be divided in to four domains, presented in Fig. 2.1. The method can take into account the full information about the shapes (and luminosity distribution) of the observed arcs (Extended arcs methods), or work in the point like images approximation (point-like images methods). Each of those approaches can either use the parameterization or, be parameter "free" and perform the inversion on a grid. There is one more class of algorithms, used mostly for weak lensing, which is a

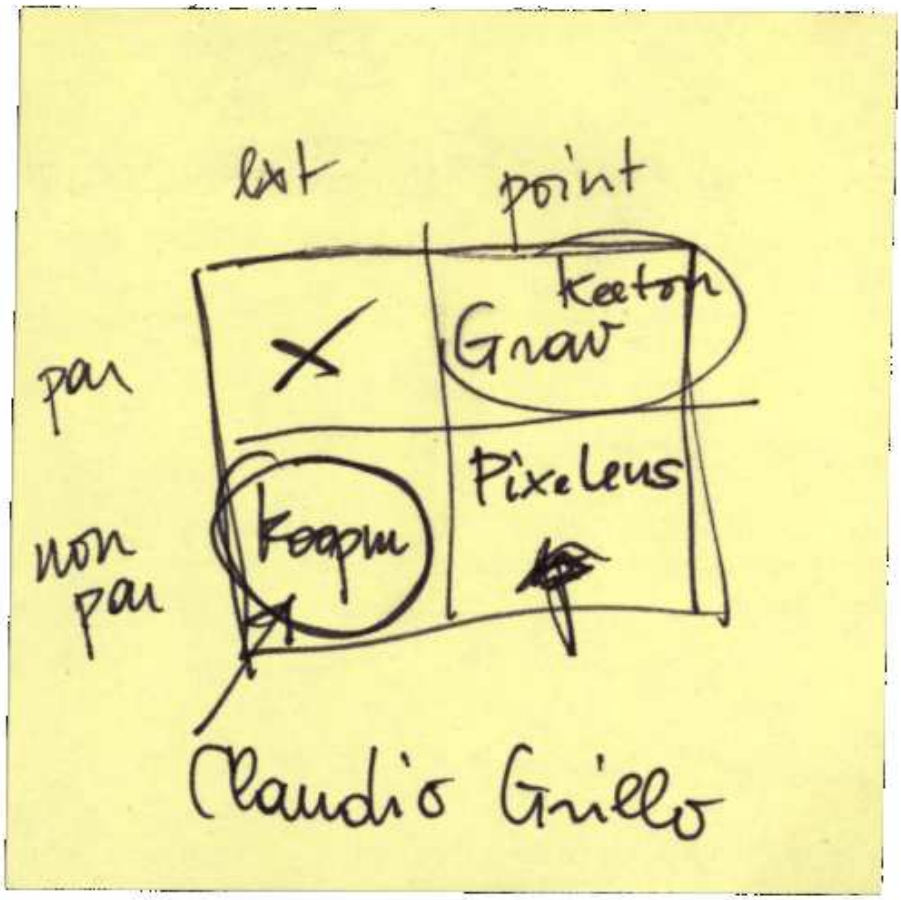

Figure 2.1: *The inversion methods. Behold Claudio Grillo's artism!*

"non-parametric" reconstruction of the lensing potential, based on statistical properties of (weakly) distorted images.

Parametric approach with point images

This is the simplest, and fastest technique of inverting the lens. It has two main variations – one can minimize the chi-square in the image plane or in the source plane. Since this approach is used as the first step of minimization used in our scheme, it is described in details in the Sects. 2.3.1 and 2.3.2. It is also an underlying method of the *GravLens* software package (Keeton 2001b).

Non-parametric approach with point images

This technique is implemented in the *PixeLens* software Read (2007). The algorithm works by reconstructing a pixelised mass map for the lens, an idea first implemented by Saha and Williams (1997). PixeLens generates the whole ensemble of models rather then just one best fit model, and this helps to address the uniqueness problem that many strong lensing studies very often face. The parameter space mapping is done using the Monte Carlo Markov Chains (see Sect. 3.3.1). The main advantages of PixeLens is that it allows a modeling of several lenses simultaneously, enforcing consistency of H_0 across different time-delay lenses. As a result the lenses can be used to constrain other lenses in a interesting way. The main disadvantage is that the modeling is based on point images (together with magnifications and time delays if measured), which results in the model being often under constrained. This is however, somehow compensated by the MCMC approach.

Another interesting algorithm of this class was presented by Trotter et al. (2000), where the lens potential is expressed as a series of multipole expansion. The benefit of this approach is that one can utilize the whole mathematical apparatus already developed to study multipoles, the disadvantage is however that one needs to use very high order expansion to be able to reconstruct the small scale structure, or extraordinary features like rings (see next subsection).

Non-parametric approach with extended images

This class of algorithms is the most promising one, since one uses the full information bout the observed arcs and arclets shapes, and luminosity distribution, to produce the mass model that is not confined to any composition of "predefined" analytic models. This allows for a spectacular discoveries, like the ring of dark matter (Fig. 2.2) discovered in the cluster Cl 0024+17 by Jee et al. (2007). One of the lately presented non-parametric methods is the one proposed by Koopmans (2005), where the author proposes to reconstruct not only the lenses potential, but also the sources brightness distribution in a non-parametric way. The method is shown schematically in Fig. 2.3. The figure illustrates how the surface brightness of each image pixel can be represented

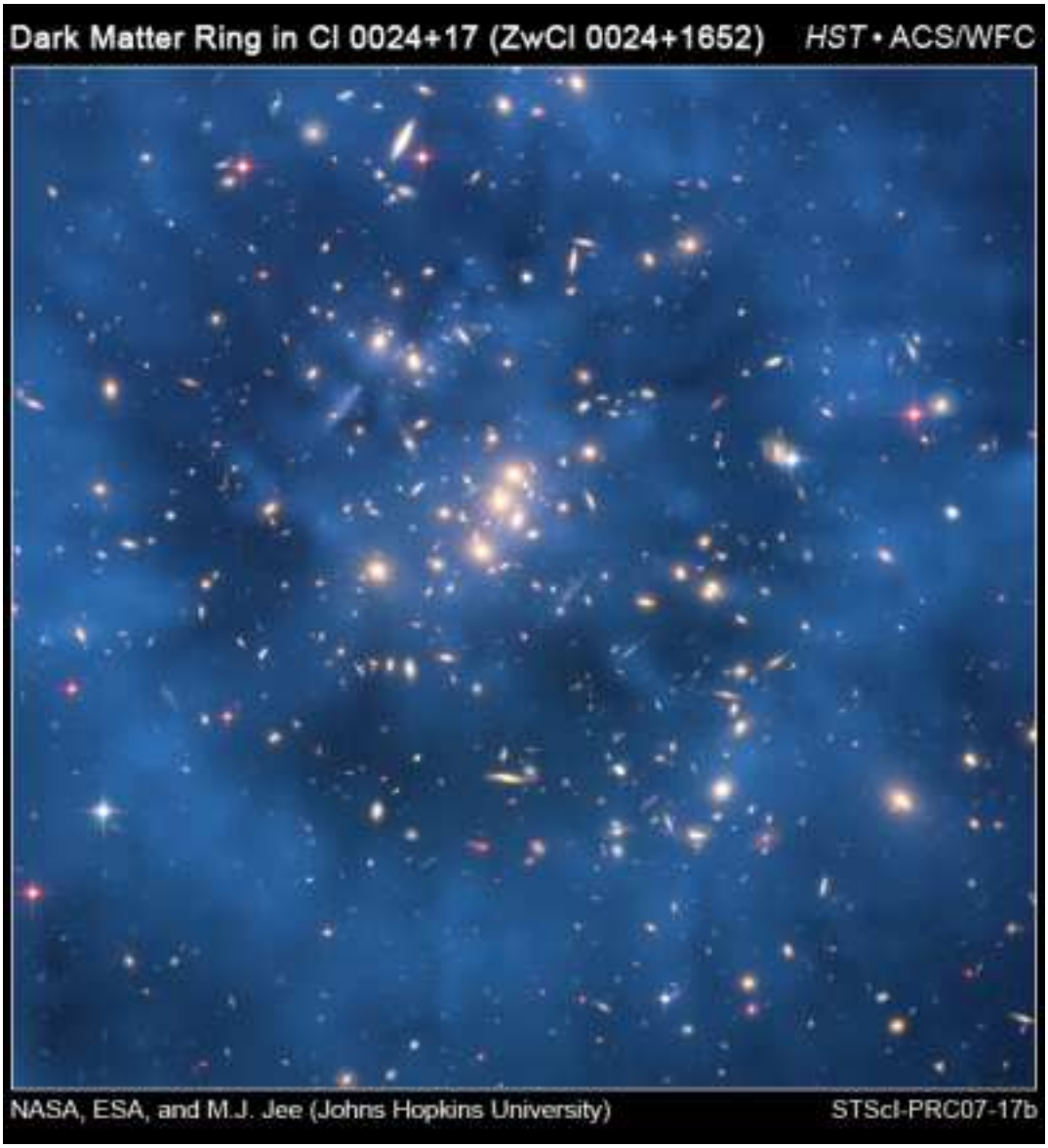

Figure 2.2: *Ring of dark matter discovered through combined strong and weak lensing studies of the Cl 0024+17 cluster (Jee et al. 2007)*

through a weighted linear superposition of the (unknown) surface brightnesses at four source pixels. Hence, one can represent this as a simple linear equation (see Koopmans and Treu 2002) and because this holds for each of the $M \times N$ images pixels, one obtains a set of $M \times N$ coupled linear equations. This set of equations is constrained by the $M \times N$ observed surface brightness values and has $K \times L$ free parameters (the unknown surface brightness values on the source grid). The advantage of this method is

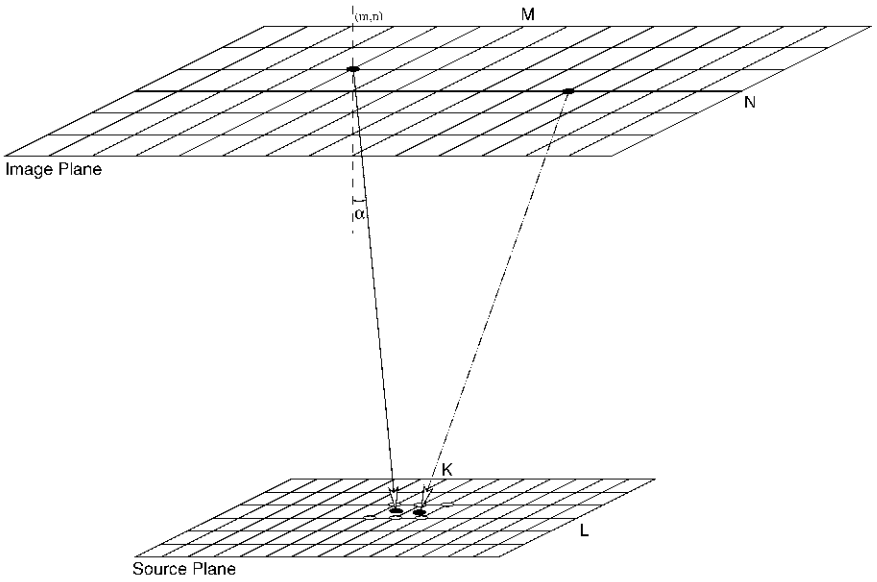

Figure 2.3: *A schematic overview of the method of non-linear image (i.e. source) reconstruction, as implemented by Koopmans (2004). In the absence of blurring or averaging inside the pixels, the position ($\boldsymbol{x}$) of each pixel (m, n) in the image plane corresponds to a position in the source plane ($\boldsymbol{y}$), through the lens equation $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{\alpha}(\boldsymbol{x})$. The surface brightnesses at these corresponding points – conserved through lensing – are the same. Because the source brightness distribution is reconstructed on a fixed grid, the surface brightness at $\boldsymbol{y}$, i.e. $\Sigma(\boldsymbol{y}) = \Sigma(\boldsymbol{x})$, is represented by a linear superposition of the surface brightnesses at the four pixels that enclose $\boldsymbol{y}$ (open circles). The weights for each of these source pixels are the bilinear interpolation weights, whose sum add to unity to conserve the flux (higher-order interpolation is also possible; Koopmans (2004, see)). This way – because of the multiple nature of the lensed images – there can be more than one constraint on a single source pixel (depending on its size and the number of multiple images). In addition, because there are multiple solutions of four weighted brightnesses adding to the observed brightness at $\boldsymbol{x}$, regularization is often required to ensure a relatively smooth and more physical source brightness distribution for lower S/N ratio data (Koopmans 2004).*

that after a successfully minimization one obtains not only the lens mass distribution but also the reconstructed source, which then might be the subject of further studies. The disadvantage is that the solution is usually under constrained, and one needs to

choose the regularization (a priori properties of lens potential, and source brightness) very carefully.

Another, more advanced, method of non-parametric inversion of lensing problem was recently introduced by Liesenborgs et al. (2006). The two key properties of this method are the usage of the adaptive grid in the image plane, and the usage of genetic algorithm for minimization. The lens mass distribution is reconstructed on a dynamic grid as a sum of basis functions, one per grid cell. A genetic algorithm then determines the mass distribution of the lens by forcing images of a single source, projected back on to the source plane, to coincide as well as possible (similarly to the method described in Sect. 2.3.2). Averaging several tens of solutions removes the random fluctuations that are introduced by the reproduction process of genomes in the genetic algorithm and highlights the features common to all solutions. Fig. 2.4 shows the reconstruction of a simulated data. One see that the algorithm is able to reproduce the positions of all the images fairly well. The disadvantage of this approach is that it does not uses

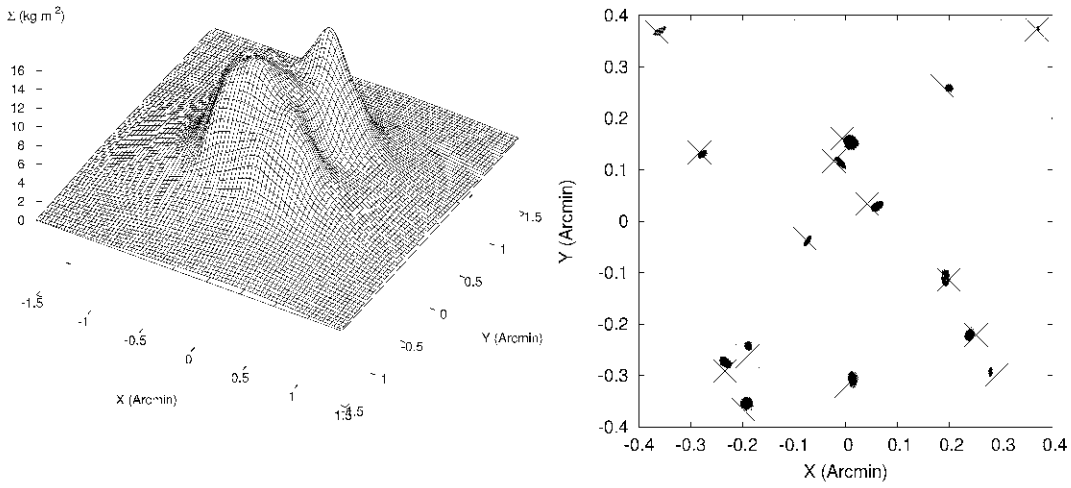

Figure 2.4: *Left-hand panel: the average of 25 individual solutions for the simulation. Right-hand panel: the positions of the back-projected images within the source plane for the averaged solution of the simulation. The true source positions are marked with crosses (Liesenborgs et al. 2006).*

the information encoded in the shapes of the observed images, and that it uses a very simple metric in a source plane to calculate the goodness-of-fit for a given model. On

the other hand, however, it shows that genetic algorithms can be successfully used in inverting the lensing problems.

Non-parametric reconstruction in weak lensing regime

This method presented by Bartelmann et al. (1996) is based on the idea of minimizing the difference between observed and model produced, statistical lensing properties over a large portions of the field of interest. The area where the mass is to be reconstructed is divided into a small (10×10) number of cells. In each cell the lensing properties such as shear and magnification are calculated from averaged measurements of ellipticities of images in that cell and then compared with the shear and magnification predicted by the model for this cell. The quantity being minimized is:

$$\chi^2 = \sum_{k,l} \left\{ (\frac{1}{\sigma_g^2(k,l)}[g_i(k,l) - \hat{g}_i(k,l)]^2 + \frac{1}{\sigma_r^2(k,l)}[r(k,l) - \hat{r}(k,l)]^2 \right\} , \tag{2.1}$$

where the $g_i(k,l) = \frac{\gamma_i}{1-\kappa}$ is the reduced shear in the cell (k,l) calculated from the data, the $r(k,l)$ is the inverse of magnification in the call (k,l), and the corresponding hatted variables are the quantities as predicted by the model (the summation over the two components of g is implied). The use of magnification and image distortions at the same time, to fit the model, brakes the mass-sheet degeneracy (Broadhurst et al. 1995). The Fig. 2.5 shows the effect of reconstruction of a simulated lensing potential.

The inversion method introduced in this work

The inversion method introduced in this work, is somehow complimentary to the ones presented in the previous paragraphs. The most important innovation presented in this work is the usage of Modified Hausdorff Distance (see Sect. 2.3.3) as a goodness-of-fit function, instead of simple chi-square based on image positions. This new metric however, might be easily incorporated into any of the inversion methods that have been developed by other researchers. The advantage of using it is that the fitting is based on full information available through the shapes of the observed images, and can be modified to also take into account the light distribution within observed luminous arcs (see Sect. 2.3.3).

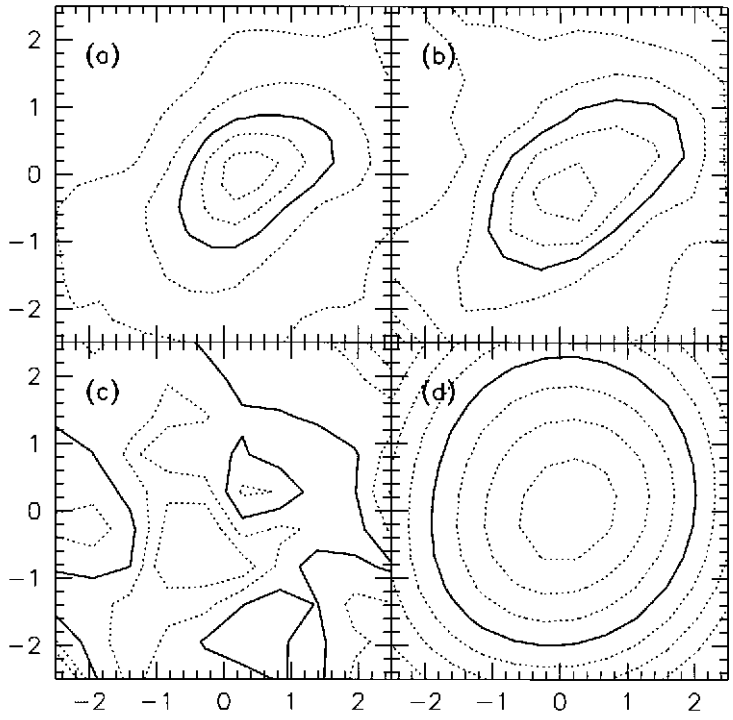

Figure 2.5: *Four contour plots showing (a) the original cluster model, (b) the reconstruction, (c) the difference between the two, and (d) the dimensionless, two-dimensional potential. Contours in (a) and (b) are spaced by 0.1, and the heavy contour follows* $\kappa = 0.5$*. In (c), contours are spaced by 0.05 and the heavy contour follows* $\Delta\kappa = 0$*. The potential is kept fixed at* $\psi = 0$ *at three corners. The heavy line in (d) follows the arbitrary contour* $\psi = -5$*, and the contours are spaced by 1.5. The side length of the fields is 5' (Bartelmann et al. 1996).*

2.2 Parameterization

In this section the parameterization used in this work will be described. The more complete set of possible models may be found in the work of Keeton (2001a). We have already seen the deflection angle for the simplest parametric model - the point mass. This model has one parameter (the mass) and can by used to model small compact objects like starts or black holes.

Before discussing specific models let us first notice some general properties that arise from symmetries. If the mass distribution is axial symmetric then the deflection angle vector is radial and has an amplitude given by

$$\alpha(r) = \frac{2}{r}\int_0^r u\kappa(u)du = \frac{1}{\pi\Sigma_{cr}}\frac{M_{enc}(r)}{r} \ , \tag{2.2}$$

where $r = \sqrt{\boldsymbol{x}^2}$ and $M_{enc}(r)$ is the total mass enclosed within the radius r. More generaly, in the case of elliptical symmetry, when the surface mass density can be written in the form of

$$\kappa = \kappa(\xi) \qquad \xi^2 = x_1^2 + x_2^2/q^2 \ , \tag{2.3}$$

where q is the projected axis ratio (the ellipticity), the lensing properties may be expressed as a set of equations (Schramm 1990)

$$\psi(\boldsymbol{x}) = \frac{q}{2}I(\boldsymbol{x}) \tag{2.4}$$

$$\alpha_1(\boldsymbol{x}) = \frac{\partial\psi}{\partial x_1} = qx_1J_0(\boldsymbol{x}) \tag{2.5}$$

$$\alpha_2(\boldsymbol{x}) = \frac{\partial\psi}{\partial x_2} = qx_2J_1(\boldsymbol{x}) \tag{2.6}$$

$$\kappa(\boldsymbol{x}) + \gamma_1(\boldsymbol{x}) = \frac{\partial^2\psi}{\partial^2 x_1} = 2qx_1^2K_0(\boldsymbol{x}) + qJ_0(\boldsymbol{x}) \tag{2.7}$$

$$\kappa(\boldsymbol{x}) - \gamma_1(\boldsymbol{x}) = \frac{\partial^2\psi}{\partial^2 x_2} = 2qx_2^2K_0(\boldsymbol{x}) + qJ_1(\boldsymbol{x}) \tag{2.8}$$

$$\gamma_2(\boldsymbol{x}) = \frac{\partial^2\psi}{\partial x_1\partial x_2} = 2qx_1x_2K_1(\boldsymbol{x}) \ , \tag{2.9}$$

where the integrals are given by

$$I(\boldsymbol{x}) = \int_0^1 \frac{\xi(u)}{u}\frac{\alpha(\xi(u))}{|1-(1-q^2)u|^2}du \tag{2.10}$$

$$J_n(\boldsymbol{x}) = \int_0^1 \frac{\kappa(\xi(u)^2)}{|1-(1-q^2)u|^{n+\frac{1}{2}}}du \tag{2.11}$$

$$K_n(\boldsymbol{x}) = \int_0^1 \frac{u\kappa'(\xi(u)^2)}{|1-(1-q^2)u|^{n+\frac{1}{2}}}du \tag{2.12}$$

$$\text{where} \quad \xi(u)^2 = u\left(x_1^2 + \frac{x_2^2}{1-(1-q^2)u}\right) \ , \qquad \kappa'(\xi^2) = \frac{d\kappa(\xi^2)}{d\xi^2} \tag{2.13}$$

Now we discuss the two parametric models I used in this work.

2.2.1 Generalized Isothermal Ellipsoid

This model has a projected surface mass density

$$\kappa(\xi) = \frac{1}{2}\frac{b^{2-a}}{(s^2+\xi^2)^{1-\alpha/2}} \ , \tag{2.14}$$

which represents a profile with a core with a scale size s, and a power low decline with an exponent α for $\xi \gg s$ defined in such a way so that the $M_{encl}(r) \propto r^{\alpha}$. In the case of spherical symmetry, the lensing properties of this model are found to be

$$\begin{aligned}\psi(r) &= \frac{1}{\alpha^2}b^{2-\alpha}r^{\alpha}F_1^2\left[-\frac{\alpha}{2},-\frac{\alpha}{2};1-\frac{\alpha}{2};-\frac{s^2}{r^2}\right] - \frac{1}{\alpha}b^{2-\alpha}s^{\alpha}\ln\left(\frac{r}{s}\right) \qquad (2.15)\\ &\quad -\frac{1}{2\alpha}b^{2-\alpha}s^{\alpha}\left[\mathrm{e}-\Psi\left(-\frac{\alpha}{2}\right)\right]\\ &= \frac{1}{\alpha^2}b^{2-\alpha}r^{\alpha} \qquad (\alpha>0, \quad s=0) \qquad (2.16)\\ \alpha(r) &= \frac{\partial\psi(r)}{\partial r} = \frac{b^{2-\alpha}}{\alpha r}\left[(s^2+r^2)^{\alpha/2}-s^{\alpha}\right] \qquad (\alpha\neq 0) \qquad (2.17)\\ &= \frac{b^2}{r}\ln\left(1+\frac{r^2}{s^2}\right) \qquad (\alpha=0)\ , \qquad (2.18)\end{aligned}$$

where the $F_1^2[a,b;c;x]$ is a hypergeometrical function (see e. g. Gradshteyn and Ryzhik 1994), e is the Euler constant, and $\Psi(x) = d[\ln\Gamma(x)]/dx$ is the logarithmic derivative of the gamma function $\Gamma(x)$. Analytical solutions for the elliptical transformations Eq. (2.4) exist for $\alpha = -1, 0, 1$. The model with $\alpha = 1$ represents an isothermal sphere, which is very often used to describe gravitationally relaxed systems like globular clusters. It is however, often used to model the smooth dark matter particles distribution inside the galaxy clusters based on the assumption that those particles are relaxed. The lensing potential $\psi(r)$ and the radial amplitude of the deflection angle $\alpha(r)$ for this model are given by

$$\psi(r) = r\frac{\partial\psi(r)}{\partial r} - bs\ln\left(\frac{s+\sqrt{s^2+r^2}}{2s}\right) \ , \tag{2.19}$$

$$\alpha(r) = \frac{\partial\psi(r)}{\partial r} = \frac{b}{r}\left(\sqrt{s^2+r^2}-s\right) \ . \tag{2.20}$$

The elliptical generalization given by Keeton and Kochanek (1998) has the form of

$$\psi(\boldsymbol{x}) = x_1\frac{\partial\psi(\boldsymbol{x})}{\partial x_1} + x_2\frac{\partial\psi(\boldsymbol{x})}{\partial x_2} - bqs\ln\left[(\Omega+s)^2 + (1-q^2)x_1^2\right]^{1/2} + bqs\ln\left[(1+q)s\right]\ , \tag{2.21}$$

$$\alpha_1(\boldsymbol{x}) = \frac{\partial\psi(\boldsymbol{x})}{\partial x_1} = \frac{bq}{\sqrt{1-q^2}}\arctan\left[\frac{\sqrt{1-q^2}x_1}{\Omega+s}\right]\ , \tag{2.22}$$

$$\alpha_2(\boldsymbol{x}) = \frac{\partial\psi(\boldsymbol{x})}{\partial x_2} = \frac{bq}{\sqrt{1-q^2}}\operatorname{arctanh}\left[\frac{\sqrt{1-q^2}x_2}{\Omega+q^2s}\right]\ , \tag{2.23}$$

$$\Omega = \sqrt{q^2(s^2+x_1^2)+x_2^2}\ . \tag{2.24}$$

In the particular case of singular sphere ($s=0,\ q=1$), the scale factor b has the physical interpretation of the Einstein radius of the mass distribution, and is related to the velocity dispersion σ of the isothermal sphere by

$$b = 4\pi\left(\frac{\sigma}{c}\right)^2\frac{D_{ls}}{D_s}\ . \tag{2.25}$$

2.2.2 Generalized NFW

Another important model used in this work is the generalized NFW profile (Moore et al. 1998). It has the 3D density in the form of

$$\rho(r) = \frac{\rho_s}{(r/r_s)^\beta(1+r/r_s)^{3-\beta}}\ , \tag{2.26}$$

where r_s is the length scale and the ρ_s is a characteristic density. The projected 2D mass density of this model can not be expressed analytically even for the spherical model, in which case it has a form of

$$\kappa(r) = 2\kappa_s u^{1-\beta}\left[(1+u)^{\beta-3} + (3-\beta)\int_0^1 (y+u)^{\beta-4}(1-\sqrt{1-y^2})dy\right]\ . \tag{2.27}$$

where $u = r/r_s$ and $\kappa_s = \rho_s r_s/\Sigma_{cr}$. The radial amplitude of the deflection angle also can not be calculated explicitly and is equal to

$$\alpha(r) = 4\kappa_s r_s u^{2-\beta}\left\{\frac{1}{3-\beta}F_1^2[3-\beta, 3-\beta; 4-\beta; -u] + \int_0^1 (y+u)^{\beta-3}\frac{1-\sqrt{1-y^2}}{y}dy\right\}. \tag{2.28}$$

For $\beta = 1$ generalized NFW reduces to the generic NFW profile introduced by Navarro et al. (1996) as a result of N-body simulations of the cosmic structure growth. In this case the spherical projected 2D mass density, lensing potential, and deflection angle can be expressed as (Bartelmann 1996)

$$\kappa(r) = 2\kappa_s\frac{1-\mathcal{F}(u)}{u^2-1}\,, \tag{2.29}$$

$$\psi(r) = 2\kappa_s r_s^2\left[\ln^2\frac{u}{2} - \operatorname{arctanh}^2\sqrt{1-u^2}\right]\,, \tag{2.30}$$

$$\alpha(r) = 4\kappa_s r_s\frac{\ln\frac{u}{2}+\mathcal{F}(u)}{u}\,, \tag{2.31}$$

where $\mathcal{F}$ is defined as

$$\mathcal{F}(u) = \begin{cases} \frac{1}{\sqrt{u^2-1}}\arctan\sqrt{u^2-1} & (u>1) \\ \frac{1}{\sqrt{1-u^2}}\operatorname{arctanh}\sqrt{1-u^2} & (u<1) \\ 1 & (u=1) \end{cases} \tag{2.32}$$

Due to the need of numerical integration, the elliptical NFW models are slow to compute.

2.3 Goodness of fit

After we have chosen the components of the lens model and their parameterization, we need to define a "goodness of fit" function that will describe how good our model reconstructs the observed lensing features.

2.3.1 Distance in the image plane

In the case of point-like images (e. g. multiple lensed quasars) the most straight forward function is the chi-square

$$\chi^2_{img} = \sum_i \delta \boldsymbol{x}_i^T \cdot S_i^{-1} \cdot \delta \boldsymbol{x}_i \ , \tag{2.33}$$

$$\delta \boldsymbol{x_i} = \boldsymbol{x}_{obs,i} - \boldsymbol{x}_{mod,i} \ , \tag{2.34}$$

where we sum over all images, the $\boldsymbol{x}_{obs,i}$ is the observed position of the image i and the $\boldsymbol{x}_{mod,i}$ is the corresponding image position as predicted by the model. The uncertainties in the measurement of the image positions are described by the covariance matrix (assuming that the measurement errors in x_1 and x_2 directions are the same, ant that there is no correlation between them)

$$S_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2, \end{bmatrix} = \sigma_i^2 I \ , \tag{2.35}$$

where, σ_i is the error on the i-th image position. Now, we can find the model that reproduces best the observed images position by minimizing the Eq. (2.33). This approach, requires however, solving the Eq. (1.45) for the image position(s) $\boldsymbol{x}$. This equation is a two dimensional and non-linear, therefore it needs to be solved numerically and it is not guaranteed that the numerical algorithm will find all the solutions. The common method (used for example by *gravlens* package Keeton 2001b) of inverting the lensing equation is based on the tiling concept. The image plane is divided into triangular tiles I_i. The lensing equation (1.45) maps every image tile I_i into a corresponding tile S_i in the source plane and thus defines a tiling in the source plane. After the tiling is performed, finding all the image positions of a given source is straight forward. We need to find all the source plane tiles S_j that cover the source position, and the corresponding tiles in the image plane I_j will give us the positions off all the images of this source. The size of the tiles gives the upper bounds on the accuracy with which we are able to predict the positions of all the images. To improve this accuracy the adaptive size of the tiles is often introduced.

2.3.2 Minimal source size

The computationally less demanding alternative to the goodness of fit presented in the previous section is based on the assumption that the size of the source producing observed images should be as compact as possible (for point like images the source needs to be point like as well). If we have N images at positions $\{\boldsymbol{x}_\mathrm{i}\}$ corresponding to one source, then we define the χ^2 as

$$\chi^2_{\mathrm{src}} = \sum_i \delta\boldsymbol{u}_i^{\mathrm{T}} \mu_i^{\mathrm{T}} S_i^{-1} \mu_i \delta\boldsymbol{u}_i \ , \tag{2.36}$$

where

$$\delta\boldsymbol{u}_i = \boldsymbol{u}_{\mathrm{obs},i} - \boldsymbol{u}_{\mathrm{mod}} \ , \tag{2.37}$$

$$\boldsymbol{u}_{\mathrm{obs},i} = \boldsymbol{x}_{\mathrm{obs},i} - w\nabla\phi(\boldsymbol{x}_{\mathrm{obs},i}) \ . \tag{2.38}$$

In the previous equations, $\boldsymbol{u}_{\mathrm{obs},i}$ is the source position (as predicted by the model) corresponding to the image $\boldsymbol{x}_{\mathrm{obs},i}$, $\phi(\boldsymbol{x}_{\mathrm{obs},i})$ is the lensing potential at image i, w is the cosmological weight of the source (see e.g. Lombardi and Bertin 1999), and μ_i is the magnification matrix at the image i. The magnification matrix μ_i is included because $\mu_i\delta\boldsymbol{u}_i \approx \delta\boldsymbol{x}_i$, so that χ^2_{src} is an approximation of χ^2_{img} in the image plane. However, this also introduces a weight in the χ^2 term, as images for which $\mu_i\delta\boldsymbol{u}_i$ are small do not contribute significantly to the sum. It is possible to write an analytical expression for the source position that minimizes χ^2_{src}:

$$\boldsymbol{u}_{\mathrm{mod}} = A^{-1}\boldsymbol{b}; \ , \tag{2.39}$$

$$A = \sum_i \mu_i^T S_i^{-1} \mu_i \ , \tag{2.40}$$

$$\boldsymbol{b} = \sum_i \mu_i^T S_i^{-1} \mu_i \boldsymbol{u}_{\mathrm{obs},i} \ . \tag{2.41}$$

The calculation of this goodness of fit function is much faster than the one presented in the previous section, because it requires the computation of the deflection angle only

at the points where the observed images are located. The resulting model however, obtained by minimizing the Eq. (2.36), usually produces more images then there are observed.

2.3.3 Modified Hausdorff Distance

Since many observed multiple images caused by gravitational lensing are not point like but have well defined shapes, it is important to define a goodness of fit between observed extended images and the ones produced by the model. If we call all the pixels that reassemble the observed arc system the set A, and all the pixels of the same arc system reproduced by the model the set B, then we can define a distance between those two sets:

$$\mathrm{HD} = \max(h'_{\mathrm{ab}}, h'_{\mathrm{ba}}) \ , \tag{2.42}$$

$$h'_{\mathrm{ab}} = \max_{a\in A} \min_{b\in B} \|a-b\| \ , \tag{2.43}$$

$$h'_{\mathrm{ba}} = \max_{b\in B} \min_{a\in A} \|a-b\| \ . \tag{2.44}$$

this is the Hausdorff definition of a distance between sets. If the observed and predicted arcs overlap perfectly, then the HD is 0, and tends to infinity with the misalignment of the two sets - this is the behavior we want from the goodness of fit function. However, in the situation when the model predicted arcs overlap the observed ones perfectly, and there is one small (few pixels) image predicted by the model but not included in the observed arcs set (because it was too faint to be detected) then the HD will be far from zero and model that reproduced this image configuration will be rejected. This is not what we expect from our goodness of fit function, since we would like to be able to predict the positions of the faint images. To solve this problem, we introduce the Modified Hausdorff Distance (MHD, Dubuisson and Jain 1994) between the modeled and observed image sets

$$\mathrm{MHD} = \max(h_{\mathrm{ab}}, h_{\mathrm{ba}}) \,, \tag{2.45}$$

$$h_{\mathrm{ab}} = \frac{1}{\|A\|} \sum_{a \in A} \min_{b \in B} \|a - b\| \,, \tag{2.46}$$

$$h_{\mathrm{ba}} = \frac{1}{\|B\|} \sum_{b \in B} \min_{a \in A} \|a - b\| \,. \tag{2.47}$$

By summing over all elements of the first set (instead of taking the maximum) and then normalizing with the power of this set, we assure that the small additional images do not have a significant impact on the total MHD, therefore the hypothetical model presented in previous paragraph would be accepted and we would be able to predict a faint image.

It is worth noticing that the Hausdorff distance Eq. (2.42) is a metric, e.g. it fulfills the four requirements:

$$\mathrm{HD}(A, B) \geq 0 \,, \tag{2.48}$$

$$\mathrm{HD}(A, B) = 0 \quad \text{iff} \quad A = B \,, \tag{2.49}$$

$$\mathrm{HD}(A, B) = \mathrm{HD}(B, A) \,, \tag{2.50}$$

$$\mathrm{HD}(A, B) \leq \mathrm{HD}(A, C) + \mathrm{HD}(C, B) \,. \tag{2.51}$$

The modified Hausdorff distance Eq. (2.45), fails however to fulfill the last condition (triangle inequality). This does not impact the model fitting process, and the advantage of possibility to predict faint images is not questionable.

Modified Hausdorff Distance with luminosity

The possible way of including the luminosity information into the (modified) Hausdorff distance is to threat it as a third dimension. That is, each point in sets A and B would have three coordinates – x_1, x_2 and l. Where x_1, x_2 are the spatial coordinates of a given pixel, and l its luminosity. This approach would require careful estimation of the errors in measurement of position and luminosity, to weight properly the contribution to the distance from spatial misplacement and luminosity mismatch for pixels. This problem requires further attention.

Chapter 3

Minimization methods

We must believe in luck. For how else can we explain the success of those we don't like.

Jean Cocteau

Abstract

In this chapter the minimization algorithms, used by the accompanying software, are introduced. This includes the Powell algorithms, the basic theory of Monte Carlo Markov Chains, together with the Metropolis algorithm, and an introduction to the Genetic Algorithms.

3.1 Introduction

After defining the lens model with a set of parameters, the goal is to find a minimum of the goodness of fit function in the space spanned by those parameters. Since the dimension of the parameters space is often high, and the goodness of fit function is not linear in those parameters, one need to use one of numerical minimization algorithms. Here three algorithms used in this thesis will be presented. Each of those methods are its strengths and weaknesses and is most suitable for a different stage of the model finding process.

3.2 Powell algorithm

The Powell algorithm is the improved "cab driver" algorithm, which is the simplest, non gradient, method for finding a minimum of a multidimensional function. Let $\boldsymbol{X_0}$ be the initial guess for the minimum of the function f, where $f = f(\boldsymbol{X}) =$

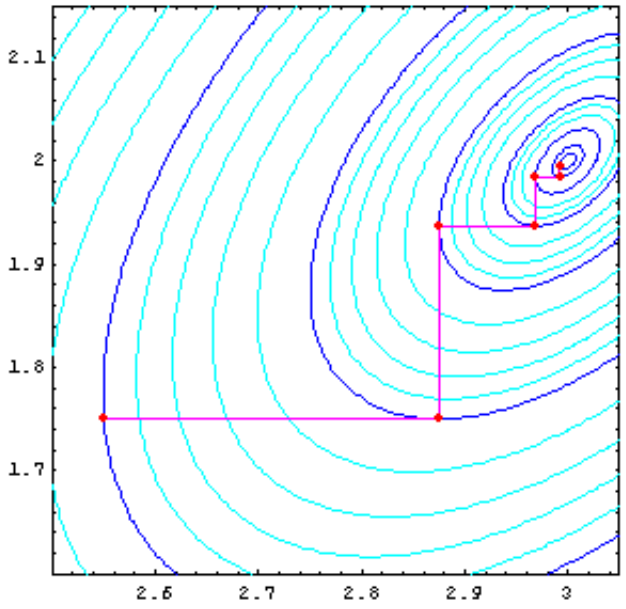

Figure 3.1: *The progress of the "cab driver" method in 2D*

$f(x_1, x_2, ..., x_n)$. The next approximation $\boldsymbol{X_1}$ to the minimum point may be constructed by finding the minimum $\boldsymbol{P_i}$ of the function f along each standard base direction sequentially e.g.

$$\boldsymbol{X_0} = \boldsymbol{P_0}, \boldsymbol{P_1}, \boldsymbol{P_2}, ..., \boldsymbol{P_n} = \boldsymbol{X_1} \ . \tag{3.1}$$

Since along each base vector the function F is a function of only one variable, it is easy and fast to find the sequence of $\boldsymbol{P_i}$ vectors. The iteration is then repeated to generate a sequence of points $\{\boldsymbol{X_k}\}_{k=0}^{\inf}$ which is assumed to converge to the minimum of the function f. Unfortunately, this method is in general not efficient and performs especially badly (in the sense of the convergence of $\{\boldsymbol{X_k}\}_{k=0}^{\inf}$ to the global minimum) when the function f has steep gradients in a direction not corresponding to any of the base directions (Fig. 3.1).

The Powell algorithm is an improved version of the idea just presented that assures a faster convergence to the minimum of the function f. We notice that in the previous method the vector $\boldsymbol{P_n} - \boldsymbol{P_0}$ represents the average direction in which the minimization proceeds in the given step. Therefore now we define $\boldsymbol{X_1}$ as vector that minimizes the

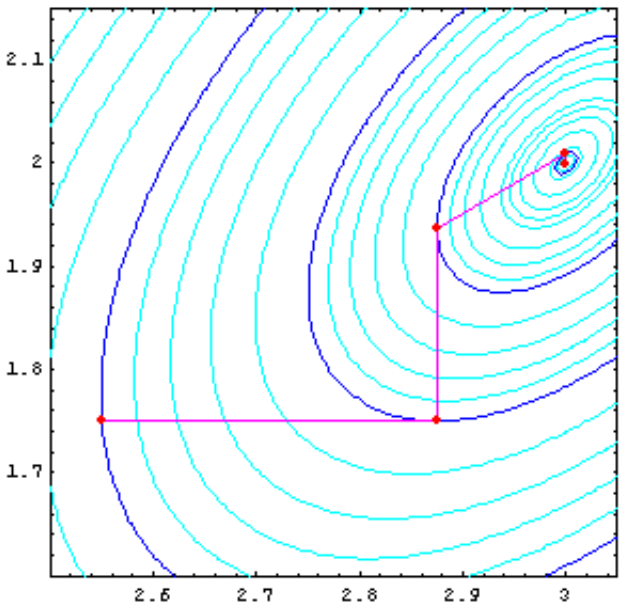

Figure 3.2: *The progress of the Powell method in 2D*

function f along the direction $\boldsymbol{P_n} - \boldsymbol{P_0}$. Since this direction was the direction in which the minimization was most efficient, we also replace one of the base vectors (which define the direction of the minimization for the next step) with $\boldsymbol{P_n} - \boldsymbol{P_0}$ (see Fig. 3.2).

Powell algorithm is a fast method of finding a minimum in a multidimensional search space, since it does not require the calculation of the partial derivatives. However, even if the minimum is found, there is no guarantee that it is the global minimum of the function in question. This, together with the freedom to parameterize the lens model in many different ways, gives rise to an important problem of the "solution uniqueness" that all the studies based on the strong lensing face. An attempt to solve this problem involves the usage of Monte Carlo Markov Chains.

3.3 MCMC

Markov Chain Monte Carlo (MCMC) methods are a class of algorithms for generating a probability distributions based on constructing a Markov chain that has the desired distribution as its stationary distribution. The MCMC methods find applications in the situations where the problem at hand is computationally too demanding

to calculate the probability distribution on a regular grid in the parameters space.

3.3.1 Basic theory of Markov chains

The theory of Markov Chains is well established. For detailed reviews on the subject please see for example Feller (1968) or Kemeny and Snell (1960) and references within. Here only the essential parts needed to understand the *Metropolis* algorithm for MCMC sampling, will be presented. In this section we will follow the theorems and proofs presented in §3.3 and §4.2 of Neal (1993). Before we proceed, we need to define what a Markov Chain is.

Markov Chain is a series of random variables, $X^{(0)}, X^{(1)}, X^{(2)}, ...$, in which the influence of the variables $X^{(0)}, ..., X^{(n)}$ on the distribution of $X^{(n+1)}$ is mediated entirely by the value of $X^{(n)}$. This can be expressed more formally,

$$P(\boldsymbol{x}^{(n+1)}|\boldsymbol{x}^{(n)}, \{\boldsymbol{x}^{(t)} : t \in \Omega\}) = P(\boldsymbol{x}^{(n+1)}|\boldsymbol{x}^{(n)}) \ , \tag{3.2}$$

where Ω is any subset of $\{0, ..., n-1\}$. The indexes $t = 0, 1, 2, 3, ...$ are often viewed as successive "times". If the allowed values of $X^{(t)}$ are finite then the chains is said to operate in finite *state space*.

Since the next link of the chain depends only on the previous one, then the full chain can be specified by giving the marginal distribution for $X^{(0)}$ – the *initial probabilities* of the various states – and the conditional distribution for $X^{(n+1)}$, given the possible values for $X^{(n)}$ – the *transition probabilities* for one state to follow another state. Let us denote the initial probability of the state $\boldsymbol{x}$ as $p_0(\boldsymbol{x})$, and the transition probability for state $\boldsymbol{x}'$ at time $n+1$ to follow state $\boldsymbol{x}$ at time n as $T_n(\boldsymbol{x}, \boldsymbol{x}')$. The Markov Chain is said to be *stationary* or *homogeneous* if the transition probabilities do not depend on time, in which case the transition probability between states $\boldsymbol{x}$ and $\boldsymbol{x}'$ is denoted as $T(\boldsymbol{x}, \boldsymbol{x}')$. Using the transition probabilities one can find the probability of state $\boldsymbol{x}$ occurring at time $n+1$, denoted as $p_{n+1}(\boldsymbol{x})$

$$p_{n+1}(\boldsymbol{x}) = \sum_{\boldsymbol{x}'} p_n(\boldsymbol{x}')T_n(\boldsymbol{x}', \boldsymbol{x}) \ . \tag{3.3}$$

This, together with the initial probabilities p_0 determines the behavior of the chain at all times.

An *Invariant* distribution over the states of a Markov chain is a one that persists for ever once it is reached. Formally, the distribution given by the probabilities $\pi(\boldsymbol{x})$ is *invariant* with the respect of the Markov chain given by transition probabilities $T_n(\boldsymbol{x}, \boldsymbol{x}')$, if for all n,

$$\pi(\boldsymbol{x}) = \sum_{\boldsymbol{x}'} \pi(\boldsymbol{x}')T_n(\boldsymbol{x}, \boldsymbol{x}') \ , \tag{3.4}$$

Often, we will use *time invariant* homogeneous Markov chains that satisfy the more restrictive condition of *detailed balance* – that is if the transition occurs from a state picked according to the probabilities given by π, then the probability of that transition being from state $\boldsymbol{x}$ to $\boldsymbol{x}'$ is the same as the probability of it being from state $\boldsymbol{x}'$ to $\boldsymbol{x}$,

$$\pi(\boldsymbol{x})T(\boldsymbol{x}, \boldsymbol{x}') = \pi(\boldsymbol{x}')T(\boldsymbol{x}', \boldsymbol{x}); . \tag{3.5}$$

The detailed balance implies that π is an invariant distribution, since

$$\sum_{\boldsymbol{x}'} \pi(\boldsymbol{x}')T(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\boldsymbol{x}'} \pi(\boldsymbol{x})T(\boldsymbol{x}', \boldsymbol{x}) = \pi(\boldsymbol{x}) \sum_{\boldsymbol{x}'} T(\boldsymbol{x}', \boldsymbol{x}) = \pi(\boldsymbol{x}) \ . \tag{3.6}$$

Note that it is possible that the distribution is invariant without the detailed balance holding. For example, the uniform distribution on the state space $\{0, 1, 2\}$ is invariant with respect to the homogeneous Markov chain with transition probabilities $T(0, 1) = T(1, 2) = T(2, 0) = 1$ and all other zero, but the detailed balance does not hold.

An *Ergodic Markov chain* is a chain from which the probabilities at the time n, $p_n(\boldsymbol{x})$, converge to the invariant distribution as $n \to \infty$, regardless of the choice of the initial probabilities $p_0(\boldsymbol{x})$. Obviously, an ergodic Markov chain can have only one invariant distribution, which is often called *equilibrium* distribution. The transition probabilities of an homogeneous Markov chain are often decomposed in to a mixture of base transitions B_k

$$T(\boldsymbol{x}, \boldsymbol{x}') = \sum_k \alpha_k B_k(\boldsymbol{x}, \boldsymbol{x}') \tag{3.7}$$

where $\alpha_k > 0$, $\sum_k \alpha_k = 1$ and each of the B_k transitions leaves the distribution invariant, but may not be ergodic individually.

3.3.1. THEOREM. (Fundamental theorem) *If a homogeneous Markov chain on a finite state space with transition probabilities $T(\boldsymbol{x}, \boldsymbol{x}')$ has π as an invariant distribution and*

$$\nu = \min_{\boldsymbol{x}} \min_{\boldsymbol{x}':\pi(\boldsymbol{x})>0} T(\boldsymbol{x}, \boldsymbol{x}')/\pi(\boldsymbol{x}') > 0 \ , \tag{3.8}$$

then the Markov chain is ergodic, i. e. regardless of the initial probabilities, $p_0(\boldsymbol{x})$

$$\lim_{n\to\infty} p_n(\boldsymbol{x}) = \pi(\boldsymbol{x}) \ , \tag{3.9}$$

for all $\boldsymbol{x}$. A bound on the rate of convergence is given by

$$|\pi(\boldsymbol{x}) - p_n(\boldsymbol{x})| \leq (1-\nu)^n \ . \tag{3.10}$$

Furthermore, if $a(\boldsymbol{x})$ is any real-valued function of the state, then the expectation of a, with respect to the distribution p_n, written $E_n[a]$, converges to its expectation with respect to π, written $< a >$, with

$$| < a > - E_n[a]| \leq (1-\nu)^n \max |a(\boldsymbol{x}) - a(\boldsymbol{x}')| \ . \tag{3.11}$$

Proof The basis of the proof is the demonstration that the distribution at time n can be expressed as a "mixture" of the invariant distribution and another arbitrary distribution, with the invariant distribution's proportion of the mixture approaching one as n approaches infinity. This growth occurs because the invariant portion can never shrink, while the non-invariant portion keeps producing extra invariant bits, due to the condition (3.8).

Specifically, we will see that the distribution at time n can be written as

$$p_n(\boldsymbol{x}) = [1-(1-\nu)^n]\pi(\boldsymbol{x}) + (1-\nu)^n r_n(\boldsymbol{x}) \ , \tag{3.12}$$

with r_n being a valid probability distribution. Note that $\nu < 1$, since $\pi(\boldsymbol{x}')$ can not be smaller that $T(\boldsymbol{x}, \boldsymbol{x}')$ for all $\boldsymbol{x}'$. The proof of the (3.12) is based on induction. The can

formula obviously hold for $n = 0$ – just put $r_0(\boldsymbol{x}) = p_o(\boldsymbol{x})$. If it holds for $n = n'$ then,

$$p_{n'+1}(\boldsymbol{x}) = \sum_{\boldsymbol{x}'} p_n(\boldsymbol{x})T(\boldsymbol{x}', \boldsymbol{x}) \tag{3.13}$$

$$= [1 - (1-\nu)^{n'} \sum_{\boldsymbol{x}'} \pi(\boldsymbol{x}')T(\boldsymbol{x}', \boldsymbol{x}) + (1-\nu)^{n'} \sum_{\boldsymbol{x}'} r_{n'}(\boldsymbol{x}')T(\boldsymbol{x}', \boldsymbol{x}) \tag{3.14}$$

$$= [1 - (1-\nu)^{n'}]\pi(\boldsymbol{x}) + (1-\nu)^{n'} \sum_{\boldsymbol{x}'} r_{n'}(\boldsymbol{x}')[T(\boldsymbol{x}', \boldsymbol{x}) - \nu\pi(\boldsymbol{x}) + \nu\pi(\boldsymbol{x})] \tag{3.15}$$

$$= [1 - (1-\nu)^{n'}]\pi(\boldsymbol{x}) + (1-\nu)^{n'}\nu\pi(\boldsymbol{x}) + (1-\nu)^{n'} \sum_{\boldsymbol{x}'} r_{n'}(\boldsymbol{x}')[T(\boldsymbol{x}', \boldsymbol{x}) - \nu\pi(\boldsymbol{x})] \tag{3.16}$$

$$= [1 - (1-\nu)^{n'+1}]\pi(\boldsymbol{x}) + (1-\nu)^{n'+1} \sum_{\boldsymbol{x}'} r_{n'}(\boldsymbol{x}') \frac{T(\boldsymbol{x}', \boldsymbol{x}) - \nu\pi(\boldsymbol{x})}{1-\nu} \tag{3.17}$$

$$= [1 - (1-\nu)^{n'+1}]\pi(\boldsymbol{x}) + (1-\nu)^{n'+1} r_{n'+1}(\boldsymbol{x}') \ , \tag{3.18}$$

where $r_{n'+1}(\boldsymbol{x}') = \sum_{\boldsymbol{x}'} r_{n'}(\boldsymbol{x}') \frac{T(\boldsymbol{x}', \boldsymbol{x}) - \nu\pi(\boldsymbol{x})}{1-\nu}$. From (3.8), we find that $r_{n'+1}(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$. One can also easily show that $\sum_{\boldsymbol{x}} r_{n'+1}(\boldsymbol{x}) = 1$. The $r_{n'+1}(\boldsymbol{x})$ is therefore a valid probability distribution, establishing (3.12) for $n = n' + 1$, and, by induction, for all n.

Using (3.12), we can now show that (3.10) holds,

$$|\pi(\boldsymbol{x}) - p_n(\boldsymbol{x})| = |\pi(\boldsymbol{x}) - [1 - (1-\nu)^n]\pi(\boldsymbol{x}) - (1-\nu)^n r_n(\boldsymbol{x})| \tag{3.19}$$

$$= |(1-\nu)^n \pi(\boldsymbol{x}) - (1-\nu)^n r_n(\boldsymbol{x})| \tag{3.20}$$

$$= (1-\nu)^n |\pi(\boldsymbol{x}) - r_n(\boldsymbol{x})| \tag{3.21}$$

$$\leq (1-\nu)^n \ . \tag{3.22}$$

Similarly we can prove (3.11),

$$| < a > - E_n[a]| = | \sum_{\boldsymbol{x}'} a(\boldsymbol{x})\pi(\boldsymbol{x}') - \sum_{\boldsymbol{x}'} a(\boldsymbol{x}) p_n(\boldsymbol{x}')| \tag{3.23}$$

$$= | \sum_{\boldsymbol{x}'} a(\boldsymbol{x}')[(1-\nu)^n \pi(\boldsymbol{x}') - (1-\nu)^n r_n(\boldsymbol{x}')]| \tag{3.24}$$

$$= (1-\nu)^n | \sum_{\boldsymbol{x}'} a(\boldsymbol{x}')\pi(\boldsymbol{x}') - \sum_{\boldsymbol{x}'} a(\boldsymbol{x}') r_n(\boldsymbol{x}')| \tag{3.25}$$

$$= \ \leq (1-\nu)^n \max_{\boldsymbol{x}, \boldsymbol{x}'} |a(\boldsymbol{x}) - a(\boldsymbol{x}')| \ . \tag{3.26}$$

This completes the whole proof of the *fundamental theorem*.

3.3.2 The Metropolis algorithm

The *Metropolis* algorithm was first introduced by Metropolis et al. (1953). Suppose that we wish to sample from a joint distribution for $X = \{X_1, X_2, ..., X_n\}$. The Metropolis algorithm does this by repeatedly considering randomly generated changes to the components of X, accepting or rejecting these changes based on how probability of the state. This process can be seen as the operation of a Markov chain built from a set of base transition probabilities, B_k, for $k = 1, 2, ..., n$. The way transition B_k operates to generate a new state, $\boldsymbol{x}'$, from the current state, $\boldsymbol{x}$, can be described as follows:

1. Select a *candidate state*, $\boldsymbol{x}'$, in which all the components other than k-th are the same as in the current state $\boldsymbol{x}$, while x'_k is picked at random from a *proposal distribution*, which may depend on $\boldsymbol{x}$, given the probabilities $S_k(\boldsymbol{x}, \boldsymbol{x}'_k)$.

2. *Accept* this candidate state with probability $A(\boldsymbol{x}, \boldsymbol{x}')$; otherwise, *reject* it, and retain the current state. In detail, this can be done by generating a random number, u, from the uniform distribution on $[0, 1)$, and then setting the next state as follows:

$$\boldsymbol{x}' = \begin{cases} \boldsymbol{x}' & u < A(\boldsymbol{x}', \boldsymbol{x}) \\ \boldsymbol{x} & \text{otherwise} \end{cases} \tag{3.27}$$

For simplicity let us assume that the proposal distribution is symmetric e. g.

$$S_k(\boldsymbol{x}, \boldsymbol{x}'_k) = S_k(\boldsymbol{x}', \boldsymbol{x}_k) \ . \tag{3.28}$$

Note that when the candidate state is rejected, the current state becomes the new state, and is included again in any averages that are being computed. Formally, the transition probabilities can be written as

$$B_k(\boldsymbol{x}, \boldsymbol{x}') = S_k(\boldsymbol{x}, \boldsymbol{x}')A(\boldsymbol{x}, \boldsymbol{x}') \prod_{i \neq k} \delta(\boldsymbol{x}_i, \boldsymbol{x}'_i) \tag{3.29}$$

$$+\delta(\boldsymbol{x}, \boldsymbol{x}')[1 - \sum_{\tilde{\boldsymbol{x}}} S_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}_k)A(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \prod_{i \neq k} \delta(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i)] \ . \tag{3.30}$$

The first term is the probability of proposing a change in component k from $\boldsymbol{x}_k$ to $vecx'_k$, and then accepting the proposed change. The second term accounts for the possibility of rejecting the candidate state, and therefore remaining in the current state.

The *Metropolis* acceptance probability has a for of

$$A(\boldsymbol{x}, \boldsymbol{x}') = min(1, P(\boldsymbol{x}')/P(\boldsymbol{x})) \; . \tag{3.31}$$

We can prove that $P(\boldsymbol{x})$ is an invariant distribution for the Markov chain used in the Metropolis algorithm by showing that the detailed balance (3.5) holds for each of the B_k, with respect to any two states $\boldsymbol{x}$, and $\boldsymbol{x}'$. If $\boldsymbol{x}_i \neq \boldsymbol{x}'_i$ for some $i \neq k$, then the detailed balance certainly holds, since both transition probabilities are zero. If $\boldsymbol{x} = \boldsymbol{x}'$, the detailed balance also holds. Otherwise, detailed balance can be verified as follows, for symmetric S_k and the Metropolis acceptance function (3.31):

$$\begin{aligned} P(\boldsymbol{x})B_k(\boldsymbol{x}, \boldsymbol{x}') &= P(\boldsymbol{x})S_k(\boldsymbol{x}, \boldsymbol{x}')A(\boldsymbol{x}, \boldsymbol{x}') && (3.32)\\ &= S_k(\boldsymbol{x}, \boldsymbol{x}') \min(P(\boldsymbol{x}), P(\boldsymbol{x}')) && (3.33)\\ &= S_k(\boldsymbol{x}, \boldsymbol{x}') \min(P(\boldsymbol{x}'), P(\boldsymbol{x})) && (3.34)\\ &= P(\boldsymbol{x}')S_k(\boldsymbol{x}', \boldsymbol{x})A(\boldsymbol{x}', \boldsymbol{x}) && (3.35)\\ &= P(\boldsymbol{x}')B_k(\boldsymbol{x}', \boldsymbol{x}) && (3.36) \end{aligned}$$

The Markov chain will be ergodic as long as $S_k(\boldsymbol{x}, \boldsymbol{x}')$ is non-zero for all $\boldsymbol{x}'_k$ and $P(\boldsymbol{x})$ is non-zero for all $\boldsymbol{x}$. This guarantees that new value for X_k has a non-zero probability of being proposed and then accepted. In n steps , there is thus a non-zero probability of moving from any state to any other. The Metropolis algorithm is often used when these criteria are not satisfied, however. Ergodicity must then be shown by somewhat more specific arguments.

3.3.3 Finite Markov Chains

MCMC chains calculated numerically are obviously finite. This causes some problems that are not present in the case of infinite chains. Two most distinct difficulties are the choice of a scale defining the average distance to probe in each step of the chain construction, and the problem of chain falling into a local minimum and not probing the whole parameter space. The scale must be chosen empirically so that the resulting chain points do not have a multiplicity higher then a few, and the parameters space should be probed smoothly and should not bare any sings on a random walk (see Fig. 3.3).

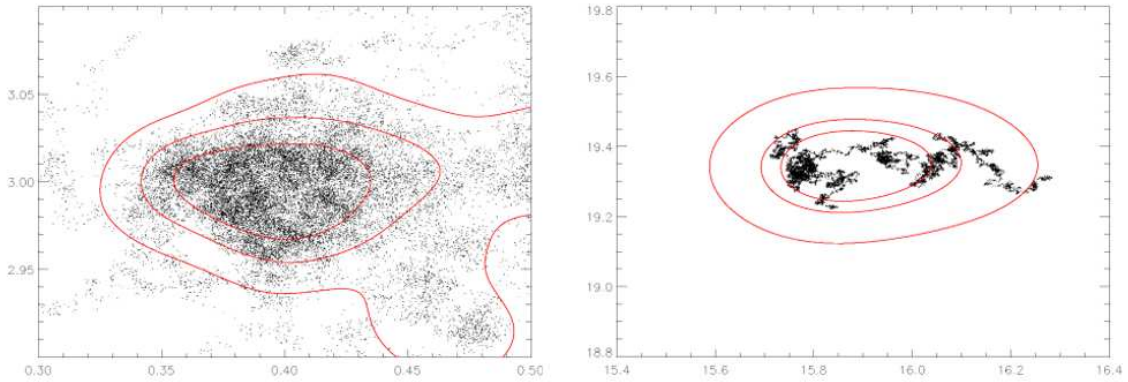

Figure 3.3: *Two examples of 2D MCMCs. The one on the left has a proper scale, while the one on the right has a scale that is too small, which results in a new point in almost every step – the MCM "crawls" through parameters space.*

The problem of local minimums is partially solved by starting multiple chains from different starting points and then combining them into the final chain.

3.4 Genetic Algorithms

Genetic algorithms (GA) are the search techniques used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (recombination). The basis of genetic algorithms is the parallelism with biology, natural selection, and evolution (Holland 1975).

Most organisms living on Earth are characterized by their *chromosomes*, where each chromosome is a sequence of *genes*. Genes are responsible for particular features of the organism, like for example color of the eyes, or height. When two organisms mate, the child organism contains genes from both their parents, which are occasionally perturbed by mutations. Then "life" evaluates how well the child organism is prepared to the environment. If the genes the child has, give it some particular advantage to survive, its chances of living long enough to mate are higher then the ones of another organism with poorer genotype. Therefore the "good" genes (or genes combinations)

have higher chances of being transfered to the next population. This leads to the whole population of organism being better adopted to the environment its living in.

A similar concept has been adopted in the genetic algorithms. Let us assume we want to find a minimum of a function $f(X)$, where X might be a set of parameters. We threat each point in the parameters space as an "organism" described by a string of genes (chromosome) – the binary coding is the most common one. In this coding each gene can have only value of 0 or 1, therefore each point in the parameter space is described with a binary string (a proficient way of converting a set of real valued parameters in to binary strings is described in the next section). The genetic algorithm proceeds as follows:

- A large random population of organisms is generated - each is represented by a binary string.
- The fitness of each organism is evaluated on the basis how well it minimizes the function f.
- The parents for the next generation of organisms are chosen - with the probability of being chosen depending on the fitness function.
- The children generation is created from the parents by combining the genes of the parents, and applying *crossover* and *mutation*.
- The worst N organisms of the old population are replaced by the children.
- The process continues until the minimum of the function f is found or the maximum number of populations is created.

3.4.1 crossover

Crossover is performed by randomly selection a position along the parents chromosomes, and then swapping all the genes before and after that position. For example if the chosen parents are:

01010101110010101010 (3.37)

11010010011101001001 01 (3.38)

and the crossover position was chosen to be 10, then the children would have the following chromosomes (before the mutation occurs)

$$0101010111010100100101 \tag{3.39}$$

$$1101001001101010101010 \tag{3.40}$$

3.4.2 mutations

After the crossover is performed, random mutations in the genotype can occur. The probability of a gene mutation, which is flipping a given gene from 0 to 1 or vice versa, should be set to a low value – typically it is around 0.0001%. Mutations are important for the overall performance of the genetic algorithm since they allows exploration of otherwise not accessible parts of the parameters space. However, if the mutation rate is too high then the information encoded in the parents chromosomes is lost and the genetic algorithm converges poorly to the optimal solution.

3.4.3 Real-valued parameters encoding

We want to find a set of (real-valued) parameters defining a lens, that minimizes the Hausdorff distance between observed and model reproduced arcs. The usual way of applying genetic algorithms to real-parameters problems is to encode each parameter as a bit string using a standard binary coding. The bit strings for the parameters are concatenated together to give a single bit string – a chromosome – which represents the entire vector of parameters. If $\boldsymbol{x}$ is the parameter vector, we will denote the corresponding bit string by corresponding uppercase letter $\boldsymbol{X}$. In this section we follow the arguments presented in Wright (1991).

If a single parameter x_i has a lower and upper bounds a_i and b_i respectively, then the standard way of binary coding x_i using n bits is to let the real values between $a_i + k\frac{b_i - a_i}{2^n}$ and $a_i + (k+1)\frac{b_i - a_i}{2^n}$, correspond to the standard binary code for the integer k for $0 \leq k \leq 2^n$. For example if $a_i = 0$ and $b_i = 8$ and $n = 6$, then the real values between 5/8 and 6/8 would be encoded as 000101. To avoid talking about intervals, we will refer to the binary code for integer k above as corresponding to the left end of the interval, namely

$a_i + k\frac{b_i - a_i}{2^n}$. Thus, in the above example we would assign the binary code 000101 to the real number $5/8$.

crossover

Now we will investigate the impact of a binary crossover on the real value of the parameters. First let us consider a special case in which the crossover point falls between the codes for two parameters. In this case, one child gets some of its parameters from one parent, and some of its parameters from the other one. For example, if the parents are represented by binary strings X and Y, corresponding to the real-valued vectors $\boldsymbol{x} = (x_1, x_2, ...x_m)$ and $\boldsymbol{y} = (y_1, y_2, ..., y_m)$, and the crossover point is between x_i and x_{i+1}., then one child corresponds to the parameter vector $(x_1, x_2, ...x_i, y_{i+1}, ...y_m)$, and the other one $(y_1, y_2, ...y_i, x_{i+1}, ..., x_m)$. Thus in this special case, the binary crossover is exactly the same as *real crossover* – the crossover applied at the real parameter level.

Next, suppose that a crossover point is chosen within the code for a parameter. Then the part of the binary code of the parameter to the left of the crossover point will correspond to the more significant bits, and the part to the right will correspond to the less significant bits. Thus, a child gets the more significant part of the parameter from one parent, and the less significant part from another parent. The child might be viewed as a "perturbation" of the first parent, where the size of the perturbation is determined by the difference in the less significant bits of the parents. If the crossover point is between bits k and $k+1$, then the perturbation corresponds to changing some of the k bits on one parent to the corresponding values of the other parent. If $R_i = b_i - a_i$ is the size of the range for this parameter, then the maximum perturbation is $R_i 2^{-k}$. This observation might be put in a form of a theorem, which has a significant practical use.

3.4.1. THEOREM. *Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be the bit strings corresponding to real parameter vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Let $\boldsymbol{Z}$ be obtained from $\boldsymbol{X}$ and $\boldsymbol{Y}$ by one-point crossover where the crossover point lies between bits k and $k+1$ of the parameter x_i and y_i. We assume that $\boldsymbol{Z}$ gets the bits to the left of the crossover point from $\boldsymbol{X}$, and those to the right from $\boldsymbol{Y}$. Then the real parameter vector $\boldsymbol{z}$ corresponding to $\boldsymbol{Z}$ can also be obtained from $\boldsymbol{x}$ and $\boldsymbol{y}$ by real one-point crossover, where the crossover point is between x_i and x_{i+1}, followed by a perturbation of parameter x_i of size at most $R_i 2^{-k}$.*

This theorem allows us to perform the crossover in the real parameter space without a need to convert parameters to binary strings.

mutation

Mutation can be viewed in the same way, with the difference that we know exact value of the perturbation. Mutating the kth bit (flipping it from 0 to 1) will produce the perturbation of the x_i parameter of the size $R_i 2^{-k}$, where R_i is again the size of the range for the parameter x_i.

Published as: J. Rzepecki, M. Lombardi, P. Rosati, A. Bignamini, P. Tozzi – *"Strong Lensing Analysis of the Cluster RCS0224-0002,"* 2007A&A...471..743R.

Chapter 4

Strong Lensing Analysis of the cluster RCS0224-0002

If scientific reasoning were limited to the logical processes of arithmetic, we should not get very far in our understanding of the physical world. One might as well attempt to grasp the game of poker entirely by the use of the mathematics of probability.

Vannevar Bush

Abstract

We present a detailed mass reconstruction of the cluster RCS0224-0002 at $z = 0.773$ from the strong lensing features observed with HST/WFPC2. The mass profile is obtained using a parametric approach. We introduce a novel technique to fit extended multiple images based on the Modified Hausdorff Distance between observed arcs and the arcs reproduced by the model. We perform a detailed error analysis of the lens parameters using the MCMC method. Our model reproduces all the observed strong lensing features of the RCS0224-0002 and predicts the redshift of one of the arcs systems to be $z \approx 2.65$ (the other system has a spectroscopic redshift of $z = 4.87$). The reconstructed inner mass profile is well fitted by a non-singular isothermal sphere, rather than with an NFW model. Dark matter substructure, derived from the light distribution of the most luminous cluster members, is crucial for reproducing the complexity of the quadrupole image system, which could not be achieved otherwise. The reconstructed mass distribution closely follows the light, however it has a significant shift from the X-ray emission of the gas. The mass of RCS0224-0002 derived from the lensing model, $\approx 2 \times 10^{14}\, M_{\odot}$ is in a very good agreement with the one obtained from the X-ray temperature measured with deep Chandra observations.

4.1 Introduction

In this chapter we present a strong lensing study of the cluster RCS0224-0002 at $z = 0.773$ which was discovered as a part of the Red-Sequence Cluster Survey (RCS, Gladders et al. 2002). After the identification of the main strong lensing features of this cluster with VLT imaging, follow-up observations were carried out with HST-WFPC2 by Gladders et al. (2002), in X-rays with the Chandra observatory (Hicks et al. 2005), and in sub-mm using SCUBA on the JCMT (Webb et al. 2005).

We construct a parametric model of the projected mass density distribution of RCS0224-0002 based on its strong lensing features, one of which with secure redshift. The method used to construct the best mass model is based on the methodology described in the previous chapters.

When I was finalizing this work, a lensing model of the same cluster has been independently presented by Swinbank et al. (2007). However, these authors focus their work on the properties of a highly magnified $z = 4.87$ galaxy observed in the field; moreover, their lensing model, which is based only on the constraints provided by a single arc system (the giant arc labeled A in Fig. 4.3), is significantly different from the one presented here.

In this chapter we use a standard cosmological model with $\Omega_{\rm m} = 0.3$, $\Omega_\Lambda = 0.7$, and $\rm H_0 = 72\ km\ s^{-1}\ Mpc^{-1}$. We give all the magnitudes in the AB system, if not otherwise specified.

4.2 Observations

The HST observations of the RCS0224-0002 were taken on the 2001/08/20 in two filters, F606W and F814W using the WFPC2 camera (PI: Gladders, Proposal ID: 9135). The target coordinates were RA: 02:24:30.82, DEC: $-$00:02:27.8 and the exposure time for each filter was 1100 seconds. The WFPC2 data reduction was performed by Associations Science Products Pipeline.[1]

The X-ray data were taken on the 2002/11/15 with the ACIS-S instrument on the

[1] *http://archive.eso.org/archive/hst/wfpc2_asn/wfpc2_products.html*

Chandra observatory (PI: Gladders, Proposal No: 03800013). The target coordinates were RA: 02:24:34.10, DEC: −00:02:30.90 and the exposure time was 14560 seconds. On the 2004/12/09, RCS0224-0002 was observed with the ACIS-S again (PI: Ellingson, Proposal ID: 05800899) with exposure time of 90150 seconds. The two ACIS-S observations were combined with CIAO 3.3, using CALDB 3.2.1, leading to 100.8 ksec of effective exposure time. Details on the reduction and spectral analysis, whose results are given below, can be found in Balestra et al. (2007).

4.3 Arc identification and cluster members

RCS0224-0002 has seven prominent luminous arcs and arclets marked as A1, A2, A3, B1, B2, B3, and B4 in Fig. 4.3. Unfortunately, out of those seven arcs, only one arc system (A) has a confirmed spectroscopic redshift of 4.87 (Gladders et al. 2002), based on a spectacular Lymannα emission seen in Fig. 4.1 and 4.2. The same authors estimated the redshift of system B within the range 1.4 to 2.7 based on the lack of emission lines in their spectra. Since the redshifts of arcs B1, B2, B3, and B4 are not known, an assumption needs to be made of whether all those arcs are images of one source or more sources. Based on very similar color, structure and distance from the center of the cluster we suppose that arcs B1, B2, B3, and B4 are images of one source and we call it system B. This conjecture is supported by the lensing model described below, since by assuming the existence of two separate systems (B1–B2, B3–B4) our model predicts relatively bright multiple images which are not observed. We excluded that the feature D is a radial arc, despite its elongated morphology, since no tangential counter images are visible and because its position and morphology makes this hypothesis unlikely. Our model suggests that feature C is a central demagnified image, which is clearly visible in Fig. 4.6 showing the F606W image after subtracting the two cD galaxies. There is also a very faint red arc, labeled E, which was not included in our analysis.

Since mass is known to follow light in galaxy clusters (see e.g. Sand et al. 2002), the distribution of color selected cluster members is often used to model substructure of the underlying dark matter. Besides to the two brightest central galaxies (BCGs), there is no public spectroscopic information available in the field, we then used the red sequence

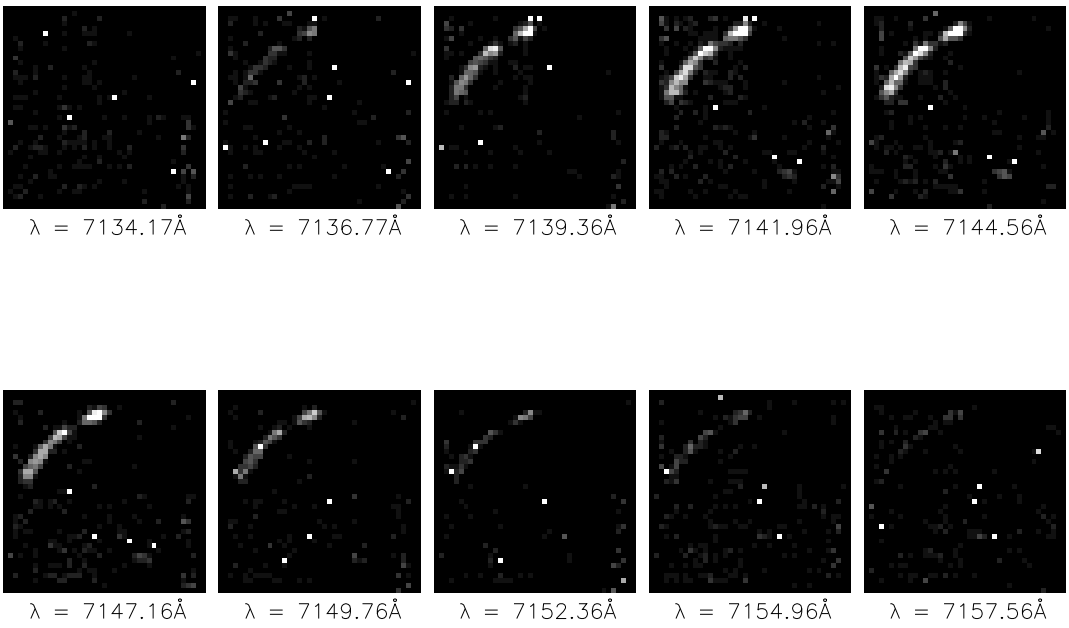

Figure 4.1: *VIMOS IFU data on RCS0224, the arc system A is clearly visible as a strong $L\alpha$ emission at the redshift of 4.8786*

to identify likely cluster members. In Fig. 4.4 we show the color-magnitude diagram over the whole WFPC2 field, highlighting red sequence objects lying withing 15 arcsec from the cluster core. Photometry was performed using SExtractor software (Bertin and Arnouts 1996), by detecting sources in the F814W band and measuring F606W $-$ F814W colors with aperture of 1″ diameter[2]. The solid and dot-dashed lines represent our best fit to the red sequence and the best fit found by Best et al. (2002) for the cluster MS1054 at $z = 083$ for the same filters, after applying a K-correction of 0.07 mag. Red sequence objects were defined as those within ± 0.25 mag of the best fit line.

[2]The WFPC2 zero points were calculated according to: $\mathrm{ZP_{AB}} = -2.5\log(PHOTFLEM) - 21.1 - 5\log(PHOTPLAM) + 18.6921$

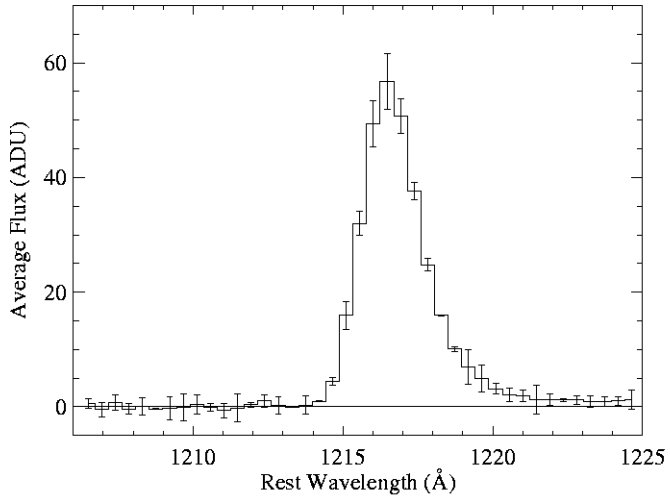

Figure 4.2: *Final average rest-frame Lyα line profile. The nominal zero flux value was established as the median of the profile in the 1000-1100Å region, rather than attempting to interpolate sky values across the extremely wide slit spectrum. Effectively, all flux blueward of Lyα appears suppressed, and the region or method selected makes a minimal difference on, for example, the continuum level seen redward of Lyα (Gladders et al. 2002).*

4.4 X-ray emission

The X-ray emission traces the hot gas trapped in the cluster potential well. The gas itself contributes about 15% to the total mass of the cluster and for relaxed systems traces closely the total mass density distribution. We overlay the X-ray contours of RCS0224-0002 from the 100 ksec Chandra observations in the 0.5–2 keV band onto the WFPC2 image in Fig. 4.7. The overall X-ray emission is not symmetric, with a plume extending NW, and its peak shifted ~ 5 arc seconds north from the two central BCGs. To measure the X-ray temperature, we used and extraction region of 36.7 arcsec (or 265 kpc), which encompasses most of the X-ray emission by maximizing the signal-to-noise. The background subtracted, unfolded spectrum is shown in Fig. 4.8. We used Xspec v.12.3.0 Arnaud (1996) to fit the data with a single temperature Mekal model (Kaastra 1992; Liedahl et al. 1995) and model the Galactic absorption with tbabs (Wilms et al. 2000), fixing the Galactic neutral Hydrogen columns density to the Galactic value obtained with radio data (Dickey and Lockman 1990). Since the signal–to–noise ratio in each

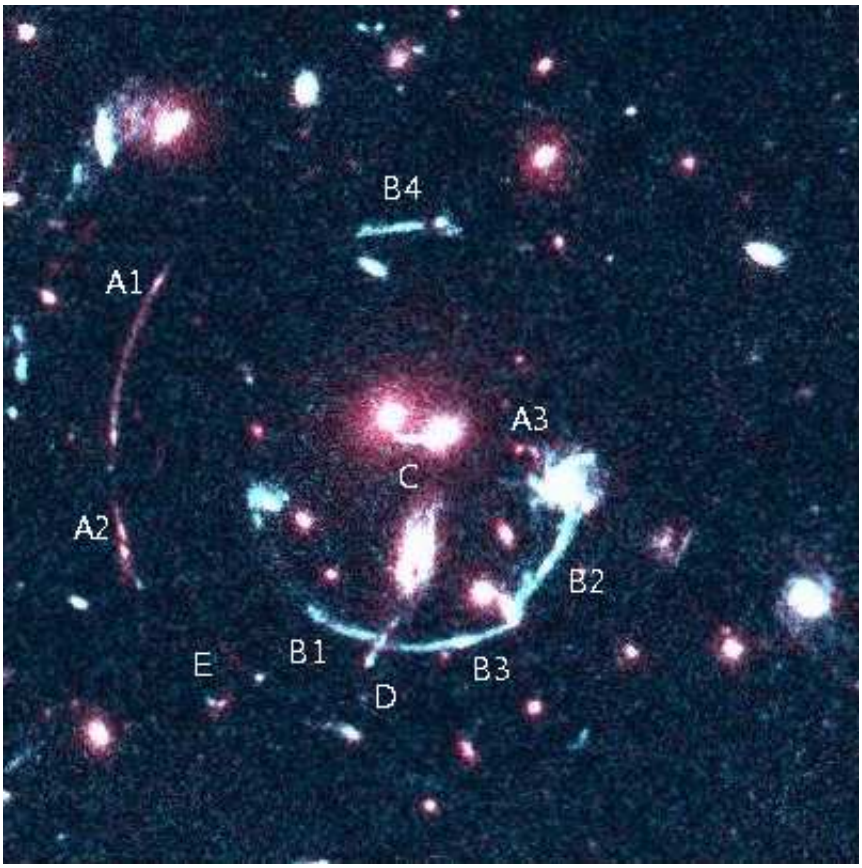

Figure 4.3: *The RCS0224-0002 cluster with labeled arcs. Color image composed from F814W and F606W WFPC2 HST images. The image is 40 arcsec across.*

energy bin is low, we used the C-statistics for the best fit model, over the energy range 0.6-8.0 keV (excluding low energy photons due to uncertainties of ACIS calibration). We used 742 ± 35 total net counts in the fit (514 ± 23 in the soft 0.5-2 keV band) and found a best fit temperature of $kT = 5.26^{+1.14}_{-1.07}$ keV (1-sigma error). The de-absorbed flux within the extraction aperture, in the (0.5 - 2.0) keV band, is 1.84×10^{-14}erg cm^{-2} s^{-1} and the rest-frame X-ray luminosity $L_X(0.5 - 2\mathrm{keV}) = (0.38 \pm 0.02) \times 10^{44}$ erg s^{-1}. The bolometric luminosity returned by the best fit model is $L_{BOL} = (1.28 \pm 0.06) \times 10^{44}$. With these values of X-ray luminosity and temperature, we note that RCS0224-0002, which is an optically selected cluster, lies on the $L_X - T$ relation determined from large samples of X-ray selected clusters (e. g. Rosati et al. 2002) We can use the measured cluster temperature to estimate the cluster mass assuming the hydrostatic equilibrium and isothermal distribution of the gas, with a polytropic index $\gamma = 1$. Using the standard β-model for

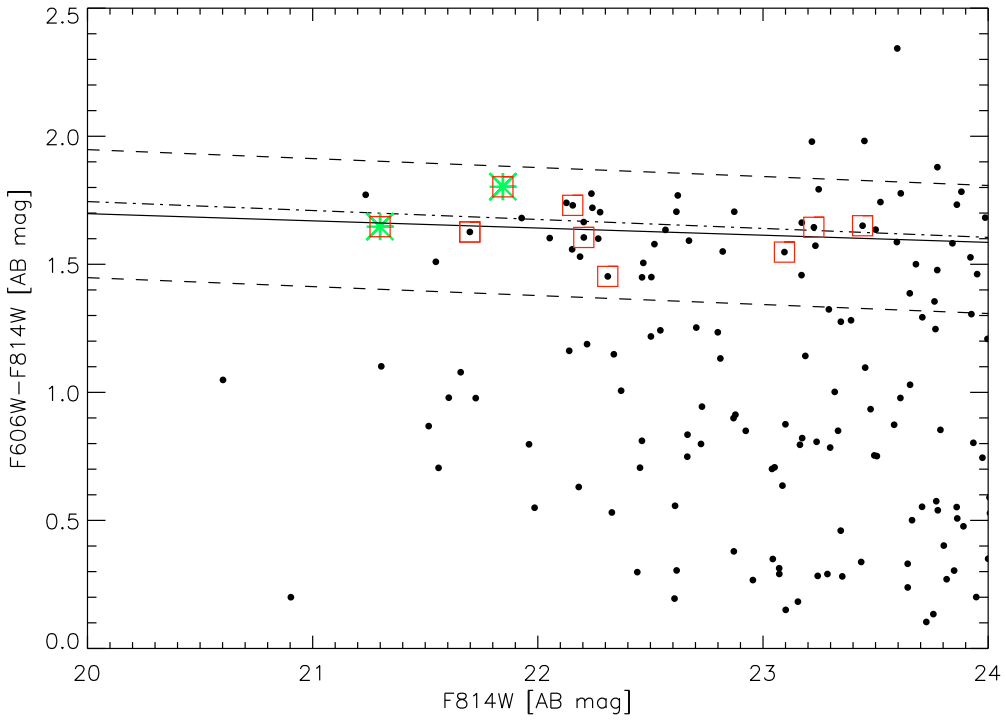

Figure 4.4: *The color-magnitude diagram of RCS0224 with the WFPC2 F606W/F814W filters. The dots represent all objects in the field. The squares represent the cluster red sequence (galaxies within 15 arcsec from the cluster center), the stars mark two central galaxies. The solid and dot-dashed lines are our best fit to the red sequence and the one of MS1054 at similar redshift.*

the gas density profile, $\rho_{\rm gas}(r) = \rho_0/[1+(r/r_c)^2]^{3\beta/2}$, the mass within the radius r can be written as (Sarazin 1988):

$$M(<r) \simeq 1.11\times10^{14}\beta\gamma\frac{T(r)}{\rm keV}\frac{r}{h^{-1}{\rm Mpc}}\frac{(r/r_c)^2}{1+(r/r_c)^2}\,h^{-1}M_\odot\,, \tag{4.1}$$

A fit to the X-ray surface brightness profile with the corresponding β-model ${\rm SB}(r) \propto [1+(r/r_c)^2]^{-3\beta+1/2}$ yields a core radius $r_c = (253\pm72){\rm kpc}$ and $\beta = 0.97\pm0.3$. Therefore the mass within $R_{200} = 0.4$ Mpc is $(1.7\pm1.1)\times10^{14}M_\odot$.

4.5 Model

We constructed the mass model of RCS0224-0002 by fitting the position and shapes of the multiple image systems A, B, and C. Based on the light distribution of most lu-

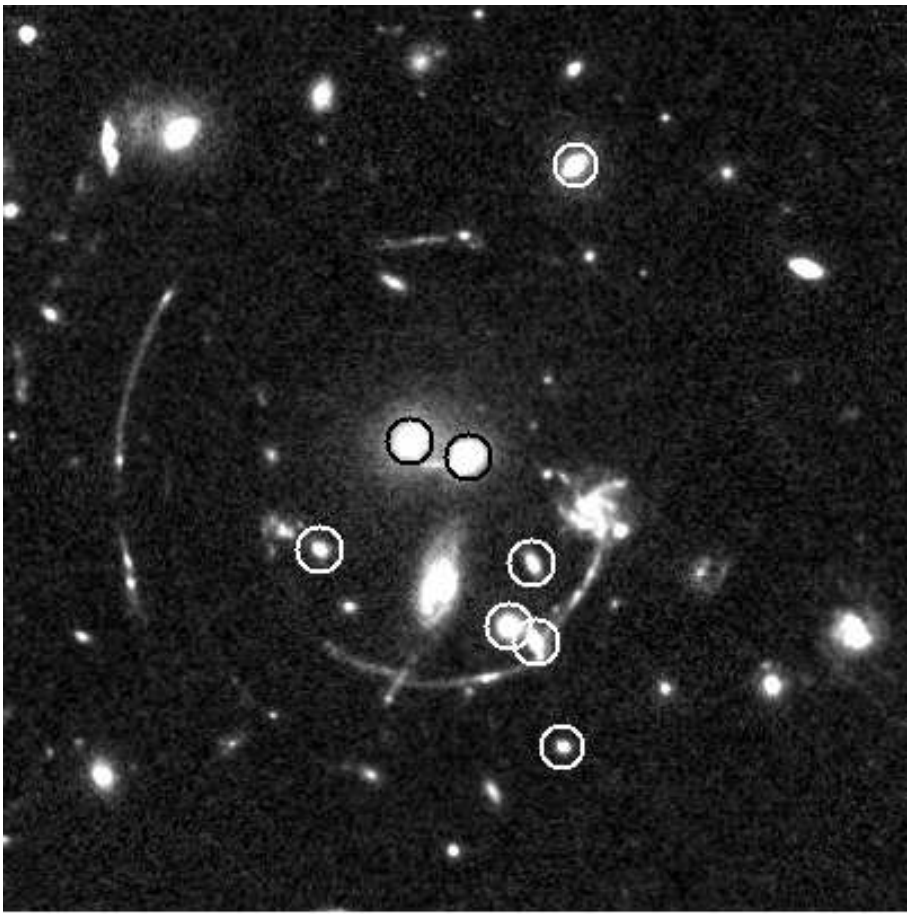

Figure 4.5: *The red sequence galaxies visible on the F814W filter WFPC2 HST image (objects that are marked here correspond to the squares in Fig 4.4).*

minous red-sequence galaxies, our model consists of several mass components: two isothermal non-singular ellipsoids to reproduce global cluster properties (NIE1, NIE2); eight isothermal non-singular spheres fixed at the position of cluster members (NIS1..8) - refereed to as the substructure; one non-singular ellipsoid, corresponding to the elongated object marked D in Fig. 4.3 (NIE3). In order to reduce the number of free parameters, we fixed the positions and the *relative* masses of the galaxy cluster clumps using the optical data available. In summary, we have 17 adjustable parameters in our model, including sources positions and unknown redshifts. All parameters are listed in Tables 4.1 and 4.2. The seven observed extended images are enough to constrain those 17 parameters due to the fact that we base our goodness of fit function not only on the position of the images but on the full information encoded in their shapes. Models in-

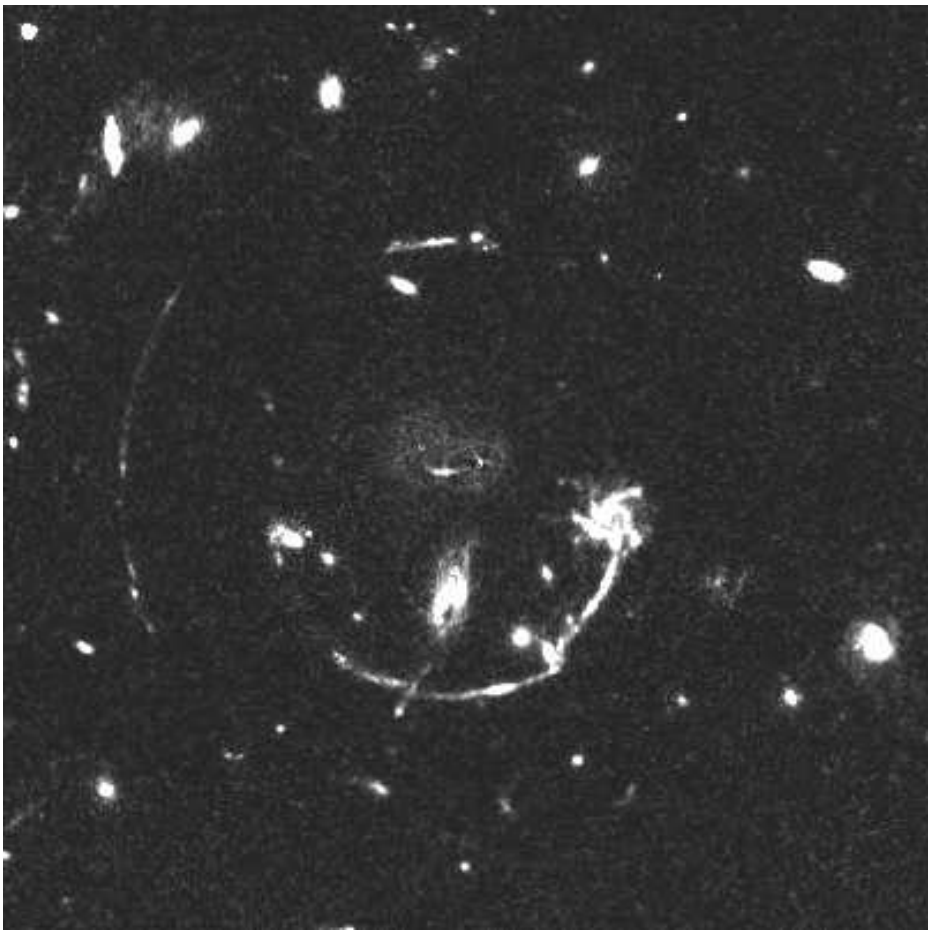

Figure 4.6: *RCS0224-0002 in the F606W filter with subtracted cD galaxies. The central radial feature C is clearly visible.*

cluding the radial feature D as a counter-image give the worst results, but as mentioned in Sect. 4.3, it is probably an foreground edge-on galaxy. Arc E was not used in the model since its redshift is unknown and it is too faint to provide any further constraint. We would like to emphasize that we do not assign any physical meaning to the two distinct smooth components (NIE1&NIE2), and we are interested in the properties of the overall, combined profile. We have also tried to fit the data with only one smooth component (NIE1) and the substructure, however in that case we were not able to fit the arcs system B accurately.

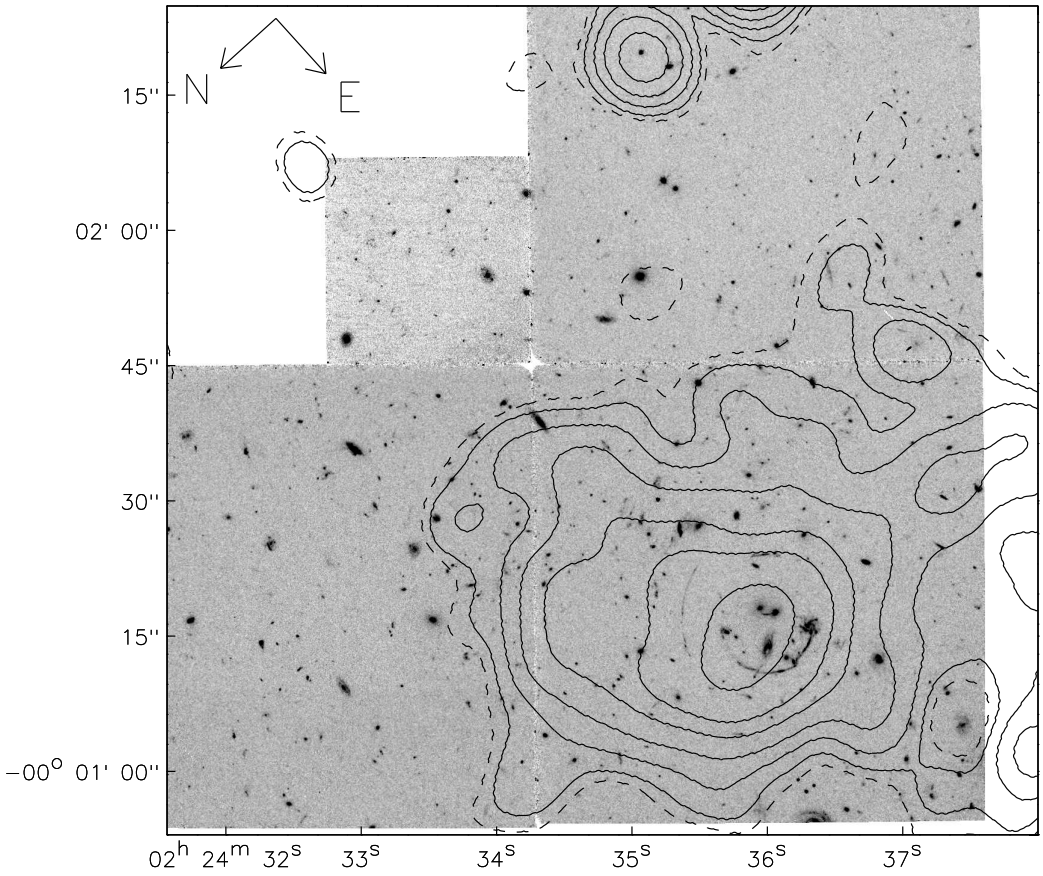

Figure 4.7: *The X-ray emission contours of RCS0224-0002 (smoothed with a Gaussian with* $\sigma = 5\mathrm{arcsec}$*) over-plotted on the F606W WFPC2 HST image.*

4.5.1 Mass profiles

Although the N-body simulations of dark matter halo formation suggest NFW profiles rather than isothermal ones, recent strong lensing studies do not exclude and in some cases even prefer isothermal profile over NFW Halkola et al. (2006); Gavazzi et al. (2003). We model here all mass components as non-singular isothermal ellipsoids, simple generalizations of non-singular isothermal spheres often used as a physical representation of a gravitationally relaxed system. The use of isothermal profiles has also the advan-

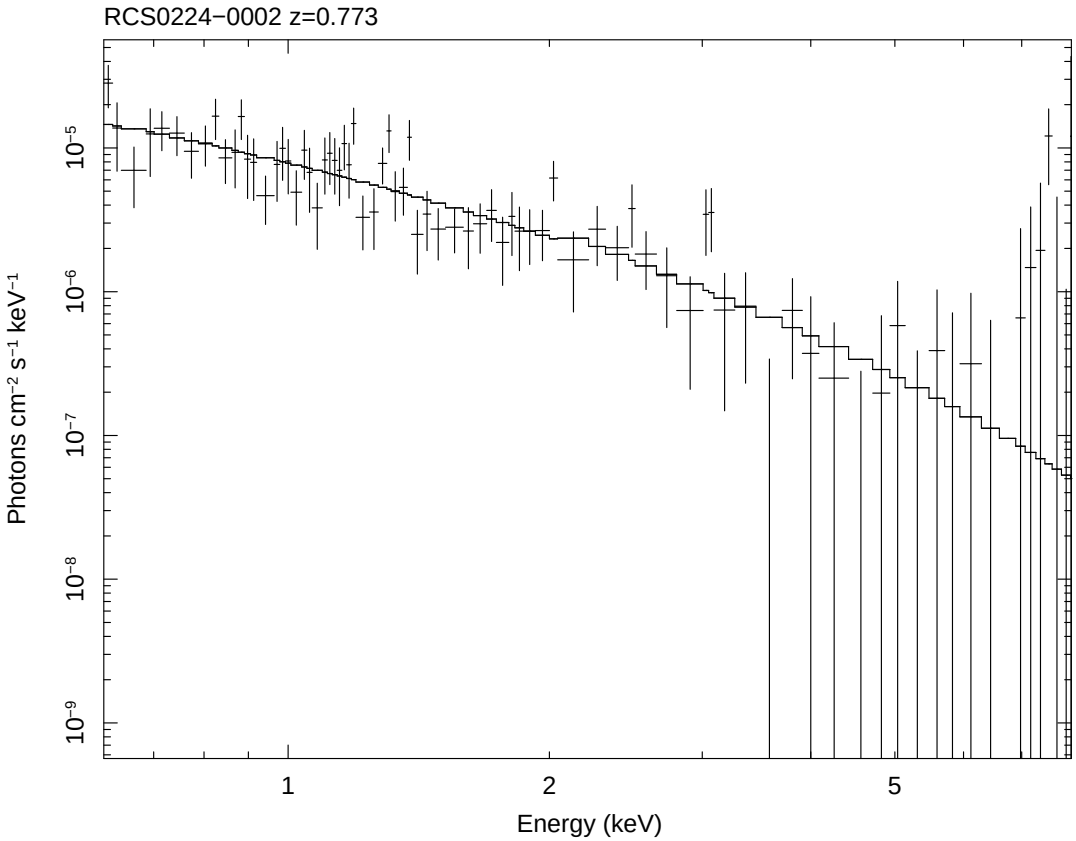

Figure 4.8: *X-ray spectrum of RCS0224-0002 from 100ksec Chandra observations, with the best fit Mekal model, for* $kT = 5.26^{+1.14}_{-1.07}$ *keV.*

tage of being computationally less demanding. The associated gravitational potential ϕ, projected mass density ρ, and deflection angle $\boldsymbol{\alpha}$ are given by Eq. (2.21) in Sect. 2.2.1.

4.5.2 Minimization method

Source plane minimization

In order the get a first, approximated solution, we perform model fitting minimization on the source plane. As described in Sect. 2.3.2 this technique is computationally very efficient, since there is no need to solve the inverse problem of the lensing equation and the deflection angle is only computed at the position of the images. We also assume that sources are small compared to the scale of variations of the lensing potential. If we have

N images at positions $\{\boldsymbol{x}_\mathrm{i}\}$ corresponding to one source, then we define the χ^2 as

$$\chi^2_{\mathrm{src}} = \sum_i \delta\boldsymbol{u}_i^{\mathrm{T}} \mu_i^{\mathrm{T}} S_i^{-1} \mu_i \delta\boldsymbol{u}_i + P \ , \tag{4.2}$$

The covariance matrix of position measurement is diagonal and takes the form (assuming that the measurement errors in x_1 and x_2 directions are the same, ant that there is no correlation between them)

$$S_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2, \end{bmatrix} = \sigma_i^2 I \ , \tag{4.3}$$

where σ_i is estimated to be $\sim 0.05''$. In the definition of our χ^2 [Eq. (4.2)] we introduced also a "penalty" function P, which was not used in the Eq. (2.36) in Sect. 2.3.2. This function, is used to bound some of the free parameters to certain intervals, and is chosen to have the functional form

$$\begin{aligned} P = \mathcal{P} \sum_{p=0}^{N} [&\mathrm{atan}\big(10^7 (b_{\mathrm{down},i} - p_i)\big) \\ &+ \mathrm{atan}(10^7 (p_i - b_{\mathrm{up},i}))] + \mathcal{P}\pi \ , \end{aligned} \tag{4.4}$$

where N is the number of bounded parameters in our model, p_i is the i-th bounded parameter, which is required to be in the range $[b_{\mathrm{down},i}, b_{\mathrm{up},i}]$. Note that the penalty function P behaves similarly to a "square potential well", i.e. the sum of two Heaviside functions; however, the use of analytic functions ensures that P is differentiable and makes our minimization numerically stable. In order to effectively bound our parameters, we used a large number for the coefficient $\mathcal{P} \approx 10^5$.

4.5.3 Extended images

The best fit model provided by Eq. (4.2) is used as starting point for the image plane analysis. This step is based on a new χ^2 minimization, with a χ^2 composed of two terms: the Modified Hausdorff Distance between the modeled and observed image sets introduced in the Sect. 2.3.3 and the "plain difference" between the same sets. For computational speed, we modify the MHD presented in the Sect. 2.3.3 – we do not sum the

Euclidean distances between points in sets A and B, but their squares. Therefore the MHD between two sets A and B we use has the form

$$\mathrm{MHD} = \max(h_{\mathrm{ab}}, h_{\mathrm{ba}}) \ , \tag{4.5}$$

$$h_{\mathrm{ab}} = \frac{1}{\|A\|} \sum_{a \in A} \min_{b \in B} \|a - b\|^2 \ , \tag{4.6}$$

$$h_{\mathrm{ba}} = \frac{1}{\|B\|} \sum_{b \in B} \min_{a \in A} \|a - b\|^2 \ . \tag{4.7}$$

In addition, the "plain difference" between the observed and modeled arcs is computed as follows. All pixels[3] in each observed arc system, generically called O, are assigned a value of 1; other pixels are assigned a value of -1. The same procedure is applied to the corresponding modeled arcs (M) and the difference $\mathrm{diff}(O, M) = |O - M|$ is calculated. In summary, the expression to minimize in the image plane is

$$X^2 = \mathrm{MDH}(D, M) + \omega \, \mathrm{diff}(M, D) + P \ . \tag{4.8}$$

The factor ω was chosen to be ~ 0.1, since this value resulted in the fastest convergence to the minimum. The penalty function P is used to bound some of the model parameters and it is defined in Sect. 4.5.2. By using two distance components, we ensure an efficient convergence of the minimization since when the modeled and the observed images start to overlap, the MHD becomes less sensitive to small variations then the plain difference. The *Powell* algorithm (Powell 1964) is used for all the minimization procedures.

4.6 Results

The best fit model (with MHD as defined by Eq. (4.5) equal to 30.3) is presented in Fig. 4.9. The values of corresponding parameters are given in Tables 4.1 and 4.2. The model reproduces fairly well all the observed strong lensing features. The giant arc A includes a counter-image 7 arcsec to the west of the BCGs (A3). The model also reproduces the quadrupole system B (B1,..B4). The central feature C is also predicted fairly close to the observed one, although with different morphology. None of the models we

[3] the selection of pixels belonging to the system is performed using a graphic program like *GIMP*

analyzed could reproduce the radial feature D, a fact that further supports the hypothesis that it is probably a foreground edge-on galaxy. In addition, inclusion of D to the lens model (NIE3) significantly improved our fits and allowed us to "break" the arcs system B into two arcs B1 and B3. The best fit redshift of the source for the system B is 2.65 ± 0.08; a spectroscopic redshift of these blue arcs, as well as object D, would provide a strong validation of our lensing model and could also be used to better constrain the mass distribution. Estimates of the statistical errors are discussed in the following section. Figure 4.10 and Tab. 4.3 show the results of some tests performed to assess how well the best fit model is able to reproduce the morphology of the multiple image systems A and B. For this purpose, we ray-traced a given image for each system (A2 and B1, marked with green boxes in Fig. 4.10) into the source plane by using its HST color image. This gave us the reconstructed source image. We then ray-traced back all the pixels from the source plane into the image plane, thus finding all counter-images of the given image. These reconstructed counter-images were finally compared with the observed ones (A1,3 and B2,3,4). In general, we found a good agreement, especially the knots in the A1 arc are very well reconstructed. The overall shapes of all the arcs in the system B are also accurately predicted. The mass of the cluster within $R_{200} = 0.4\mathrm{Mpc}$ obtained from the model is $1.9 \pm 0.1 \times 10^{14}\ \mathrm{M}_{\odot}$ and its distribution is shown in Fig. 4.11. This is in a good agreement with an mass derived above from the X-ray temperature. Since we do not know all the cluster member galaxies, we cannot reliably estimate the mass-to-light ratio of the whole cluster. For the substructure (the mass associated with the luminous cluster component - NIS1..8), we find an average mass-to-light ratio $\mathrm{M/L_{B,}}_{vega} \approx 3.6\ \mathrm{M}_{\odot}/\mathrm{L}_{\odot,\mathrm{B}}$. We converted the observed F814W filter flux to the rest frame B filter flux, by calculating a k-correction for a template elliptical galaxy from Kinney et al. (1996). The center of the mass of the best fit model follows the light distribution. NIE1 is found to be a diffuse (core radius $\approx 15\,\mathrm{arcsec}$) mass component close to the peak of the X-ray emission. The latter is shifted ≈ 5" from the NIE2 component, which corresponds to the center of the potential well and the position of the BCGs. This may indicate the presence of a merger. The radial average profile of the best fit surface mass density is shown in Fig. 4.12. This can be well approximated by a power law profile with a slope $\gamma = 0.74^{+0.03}_{-0.04}$, which is closer to the isothermal profile ($\gamma = 1$) than results obtained in other clusters. For ex-

ample, the analysis of the cluster J1004-4112 yielded $\gamma \approx 0.5$ (Sharon et al. 2005) and $0.3 < \gamma < 0.5$ (Williams and Saha 2004), whereas Broadhurst, Benítez, Coe, Sharon, Zekser, White, Ford, Bouwens, Blakeslee, Clampin, Cross, Franx, Frye, Hartig, Illingworth, Infante, Menanteau, Meurer, Postman, Ardila, Bartko, Brown, Burrows, Cheng, Feldman, Golimowski, Goto, Gronwall, Herranz, Holden, Homeier, Krist, Lesser, Martel, Miley, Rosati, Sirianni, Sparks, Steindling, Tran, Tsvetanov and Zheng (2005) found $\gamma = 0.5$ in A1689 using a large number of identified multiple images. Note that the flat core of the mass profile we have found, being a result of a hight value of the r_c of the NIE1 component, is well constrained by the position of the central arc C. The change of the r_c by 50% causes the shift in the C arc position of ≈ 1 arc sec.

By approximating the mass density distribution with NFW-like profile of the form

$$\rho(r) = \frac{\rho_0}{(r/r_c)^{\beta}(1 + r/r_c)^{(1-\beta)}} \ , \tag{4.9}$$

we find a slope $0.69^{+0.09}_{-0.13}$, flatter then the canonical NFW model ($\beta = 1$), however in good agreement with other studies which obtained $\beta < 1$. For example, Sand et al. (2002) finds $\beta = 0.35$ for the galaxy cluster MS1237-23, and $\beta < 0.57$ (at 99% confidence level) from the analysis of a large sample of clusters (Sand et al. 2004).

In addition, we have tried to fit a model based the universal NFW profile rather than NIE. The result, presented in the Fig. 4.13, shows that an NFW model performs significantly worse then the NIE one. The arcs A1 and A2 are reproduced fairly well, but the counter image A3 is found much too far from the cluster center. Moreover, in the NFW model feature B4 is split into two arcs (the second of which is not observed) and the reproduced arc B2 is shifted with respect to the observed one. This is reflected by the value of MHD, which is ten times bigger then the corresponding value for the best-fit NIE model. We note, however, that this bad performance might be due to the approximated NFW elliptical model used in our code, where the ellipticity is achieved by perturbing the potential of the spherical NFW profile instead of its density. This approximation holds for potentials close to spherical, and therefore we need to impose additional restrictions on the ellipticity of the NFW components.

4.7 Error analysis

Our method involves the minimization of the MHD whose expression (Eq.18) is not a formal χ^2 and includes a number of penalty functions (weights) to limit the range of some parameters. As a result, it is difficult to obtain reliable errors on the best fit parameters. In the presence of many parameters, the Monte Carlo Markov Chain (MCMC, see for example Neal 1993) method is an efficient way to estimate the likelihood associated to our best fit model. MCMC is used as a third step of our minimization process by reconstructing the probability distribution function of our model parameters. We start the construction of Markov chain using the *Metropolis* algorithm (Metropolis et al. 1953) from the best fit solution of the MHD minimization. We use a number of chains with seeds randomly distributed around the best fit point. We discard the first 10 points from each series to give the chain the time to reach the equilibrium. The resulting chain being the composition of all those partial chains provides an approximate probability distribution function for our parameters, from which we estimate the confidence levels shown in the Fig. 4.14. Also by randomly probing the parameters space, the MCMC algorithm helps to fine tune our best fit parameters returned by the previous step of minimization. Most of the parameters are well constrained (within 10 - 20 percent). The unknown redshift of the arc system B appears to be well constrained, $z_B = 2.65 \pm 0.08$. The mass to light ratio of the substructure is however poorly constrained to be $3.6^{+3.3}_{-1.8}$.

We estimated the errors of a single power law and NFW-like profile parameters by drawing a random sample of models from our Markov Chain, and then fitting a single power law and NFW-like profile to that sample. The resulting error estimates are presented in Fig. 4.15. This shows that isothermal and NFW profiles are excluded with 99% confidence level.

4.8 Conclusions

We have performed a strong lensing analysis of the cluster RCS0224-0002 using HST/WFPC2 images in F814W and F606W bands. We used two arc systems: a red giant tangential arc 14 arcsec , from the center, with measured redshift of 4.87, for which we identified

an inner counter image, and a system of blue arcs at smaller radii with no spectroscopic information.

We have modeled the mass distribution with with three mass components: isothermal spheres associated with the most luminous cluster members to model the substructure, and two isothermal ellipsoids to model the underlying smooth mass component. Since spectroscopic information is available in the literature only for two cD galaxies, we identified likely member galaxies in the cluster core from the red sequence, which is clearly detected in the F606W-F814W color distribution. To infer the mass distribution from the position and shapes of the strong lensing features we used a three-step approach: (i) minimization of the size of the two sources on the source plane, (ii) minimization of the difference between the observed and modeled arcs on the image plane, based on the Modified Hausdorff Distance, and (iii) a refined estimate of the best fit parameters and errors analysis with the Monte Carlo Markov Chain. The resulting mass density reproduces all the strong features fairly well. The redshift of the blue arc system is predicted to be 2.65 ± 0.08.

We find that the substructure made of nine isothermal components centered on the brightest cluster members, with $\mathrm{M/L_{B},}_{vega} \approx 3.6 \mathrm{M_{\odot}/L_{\odot,B}}$ is crucial to exactly reproduce the shapes and positions of all the arcs.

By fitting a single power-law or NFW-like halo to the radial average mass density distribution we have found that both profiles are far from canonical isothermal and standard NFW: we have found the power-law parameter γ to be $0.74^{+0.03}_{-0.04}$ ($\gamma = 1$ for an isothermal profile) and steepness parameter for NFW-like profile β to be $0.69^{+0.09}_{-0.13}$ ($\beta = 1$ for a NFW profile), with the upper boundary very well constrained. Both those values are consistent with the results obtained by studying the strong lensing properties of other clusters (see Sand et al. 2002, 2004). The best fit NIS has $\sigma_v = 925\,\mathrm{km/s}$ and $r_c = 11\,\mathrm{kpc}$; the best fit NFW has $R_{200} = 0.4\mathrm{Mpc}$ and concentration parameter $c = 3.4^{+0.4}_{-0.5}$, similarly to other massive clusters ($c \approx 4$ for a z = 0.18 cluster Halkola et al. 2006, $c \approx 5$ for z = 0.68 cluster Williams and Saha 2004). However, a wide range of concentration parameters are found (e.g. for $c > 10$ see Broadhurst, Takada, Umetsu, Kong, Arimoto, Chiba and Futamase 2005). We have measured the total mass of the cluster within R_{200} to be $1.9 \pm 0.1 \times 10^{14}\,\mathrm{M_{\odot}}$ and its main component may be well described by a two NISs

with a $\sigma_{v1} = 945^{+30}_{-23}\,\mathrm{km/s}$, a $r_{c1} = 112^{+13}_{-14}\,\mathrm{kpc}$, a $\sigma_{v2} = 702^{+31}_{-28}\,\mathrm{km/s}$, and a $r_{c2} = 12^{+4}_{-2}\,\mathrm{kpc}$. The mass of RCS0224-0002 derived from the lensing model is in a very good agreement with the one obtained from the X-ray temperature measured with deep Chandra observations ($M_{200} = (1.7 \pm 1.1) \times 10^{14} M_{\odot}$).

This analysis shows that even with a limited number of identified multiple images we could constrain the mass distribution fairly accurately. This was possible, in the case of RCS0224-0002, because the two arcs systems are at very different angular diameter distances and probe significant fraction ($\approx 20\%$ for the arcs system A, and $\approx 60\%$ for the system B) of the Einstein rings. Further spectroscopic observations of the system B, as well as cluster members, will allow a very robust constraint of the mass density profile of the inner core of this cluster and its substructure.

Table 4.1: *Parameters defining our model (see equation **??**–**??**) after minimization. Parameters in parenthesis were allowed to change during minimization*

	NIE1	NIE2	NIS1	NIS2	NIS3	NIS4	NIE3	NIS5	NIS6	NIS7	NIS8
x_1	(16.834)	19.413	18.039	20.578	24.799	23.494	17.621	22.389	23.614	25.296	14.097
x_2	(18.502)	20.307	20.834	20.147	7.336	15.330	10.253	12.612	12.005	32.969	16.076
z	0.782	0.782	0.782	0.782	0.782	0.782	0.782	0.782	0.782	0.782	0.782
b	(19.196)	(10.086)	(0.116)*	(0.191)*	(0.027)*	(0.032)*	(0.417)	(0.081)*	(0.075)*	(0.087)*	(0.037)*
q	(0.396)	(0.597)					0.3				
θ	(2.994)	(1.409)					0.873				
s	(15.539)	(1.778)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

x_1, x_2 : central position in arc seconds in the coordinate system of the Fig. 4.11, z : redshift, b : scale factor in arc seconds, q : ellipticity, θ : position angle in radians, s : core radius in arc seconds

* – for the substructure the M/L ratio has been used as the variable for the minimization

Table 4.2: *Parameters defining sources after minimization. Parameters in parenthesis were allowed to change during minimization*

	SOURCE1	SOURCE2
u_1	(15.979)	(18.892)
u_2	(19.204)	(19.412)
z	4.878	(2.648)

U_1, U_2 : source position in arc seconds, z : redshift

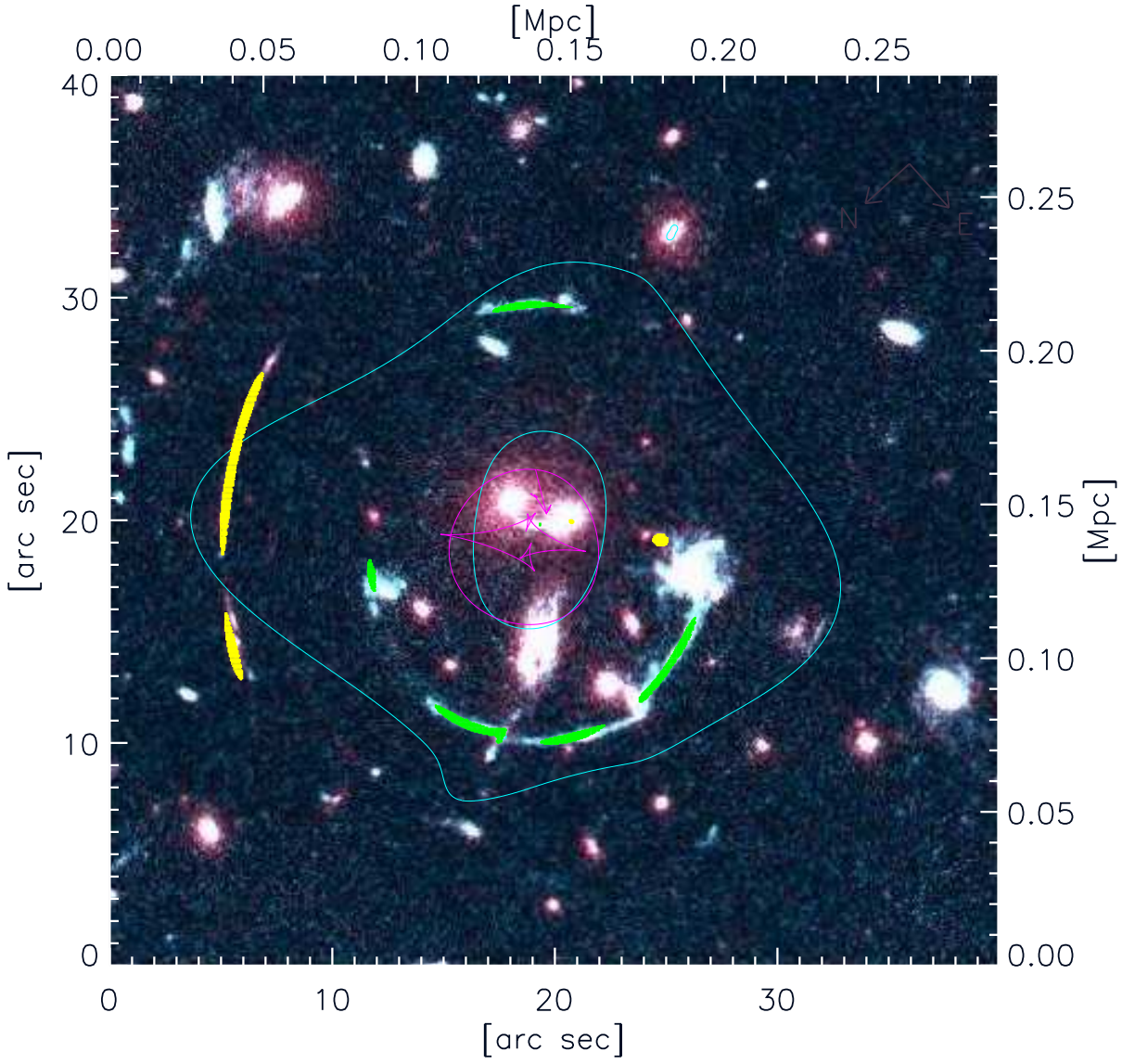

Figure 4.9: *Images reproduced by our best fit model over-plotted on the combined F606W/F814W WFPC2 HST image. The closed lines show the critical curves and caustics for a source at* $z = 4.87$. *The center of the image is at RA 02:24:34.218, Dec -00:02:31.64.*

4.9 The Way to Perfection

Before reaching the best fit model described in the previous section, the ensemble of different models have been tried out, varying in the number of components and con-

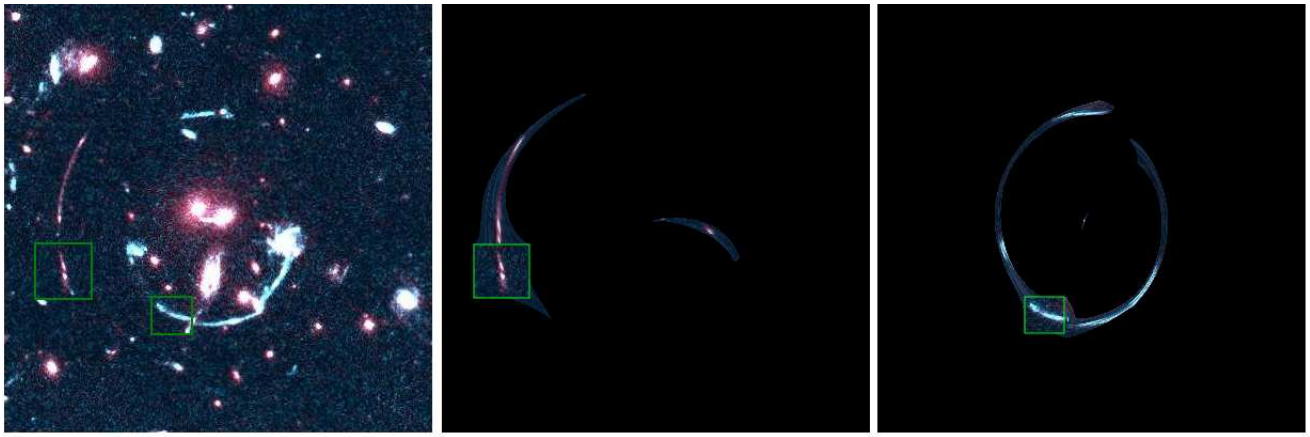

Figure 4.10: *Result of image plane – source plane – image plane mapping. Panel to the left shows the arcs (marked by boxes) used to reproduce the arc systems. Middle and right panels show the arc systems as reproduced by the best fit mass model (the box marks the original image).*

strains. In this section we will present some of those models, often they have interesting properties not seen in the model presented in the previous section.

4.9.1 Single Isothermal Ellipsoid

The simples model that have chances of reproducing the general strong lensing properties of the cluster is a single isothermal ellipsoid. Therefore that parameterization has been tried out at the beginning. The lens has 6 free parameters (position of the NIE, position angle, ellipticity, and core radius s). The result of the minimization is presented in the Fig. 4.16.

This model has the mass of $1.4\times10^{14}\mathrm{M}_{\odot}$, the $R_{200} = 0.38\mathrm{Mpc}$, and the redshift of the B system is found to be 2.79. It has difficulties with reconstructing accurately the observed strong lensing features. In particular the A1 and A2 arcs merge, and the position of arc A3 is shifted 5 arc seconds in respect to the observed position. The B arcs system also does not fit the observation well. The B1 and B3 arcs merge into one arc positioned in between the observed arcs B1 and B3, the arc B2 is too small, and finally the B4 is shifted by 4 arcsec in respect to the observed position.

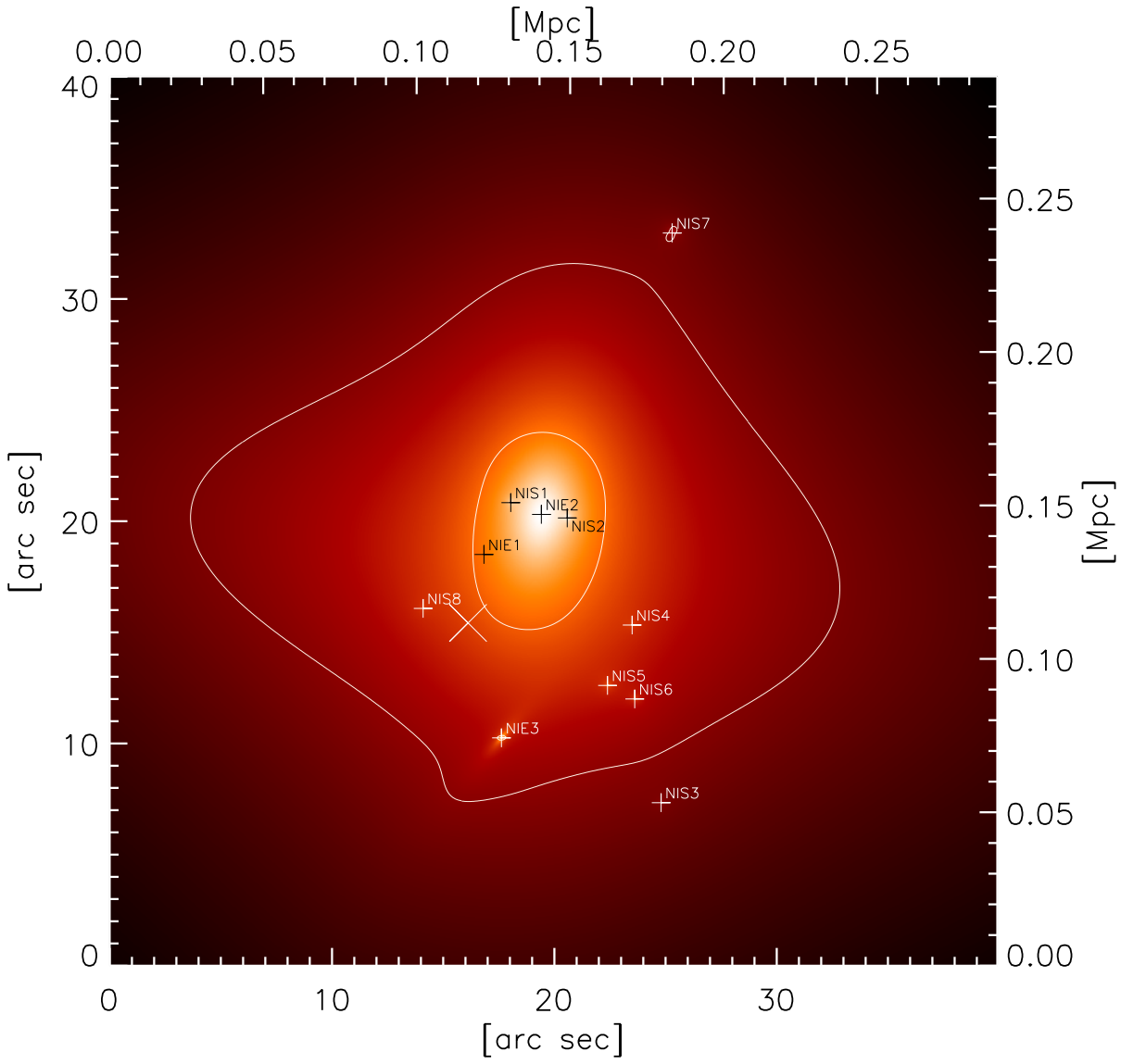

Figure 4.11: *Mass density produced by our best fit model. The closed lines are the critical curves for a source at $z = 4.87$. The crosses (+) mark the positions of our model components. The big cross (X) gives the position of the peak of the X-ray emission. The center of the image is at RA: 02:24:34.218 Dec: -00:02:31.64, the orientation as in Fig. 4.7*

4.9.2 Single Isothermal Ellipsoid with Substructure

To improve the ability of the model to reproduce accurately the observed strong lensing features, we add the substructure – isothermal spheres centered on the bright galaxy members selected using the red sequence. We keep the mass to light ratio for the sub-

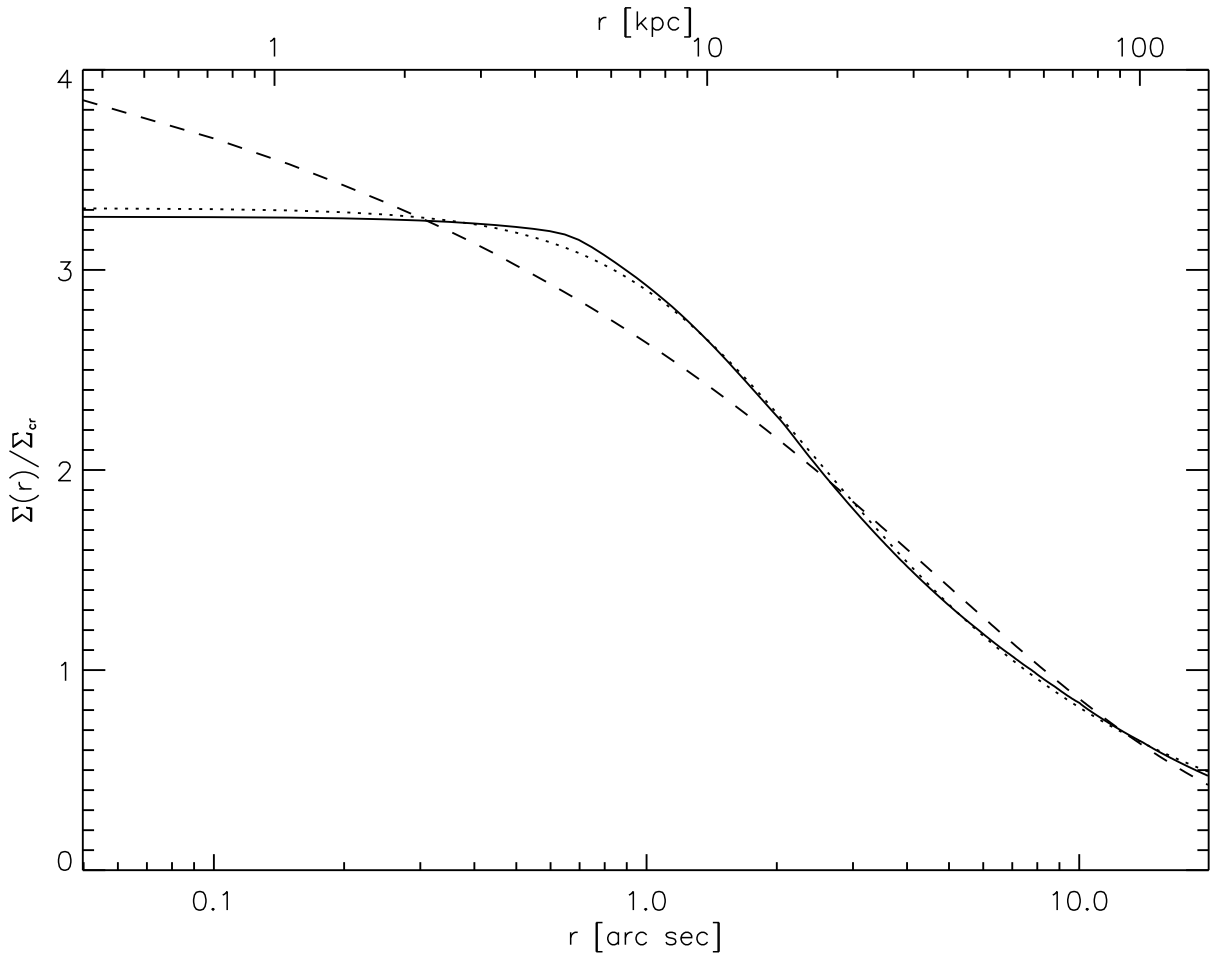

Figure 4.12: *Radial average profile of the surface mass density of our best fit model (solid line) versus power law profile with* $\gamma = 0.74$ *(dotted line) and NFW-like profile with* $\beta = 0.69$ *(dashed line).*

structure constant, therefore this lens has only one more parameter in respect to the previous one. The resulting model is presented in Fig. 4.17. Much of the improvement can be noticed. In particular the positions of arcs A3 and B4 are now accurately reproduced. The whole arc system B is reconstructed much more precisely, together with the break between arcs B1 and B3. Unfortunately the arcs A1 and A2 are still merged. The mass predicted by this model is $3.5 \times 10^{14}\mathrm{M}_{\odot}$, the $R_{200} = 0.54\mathrm{Mpc}$, and the redshift of the B system is found to be 3.14. Note that the central arc C is much more extended here than in the best fit from Sect. 4.6, its shape however does not correspond to the observation.

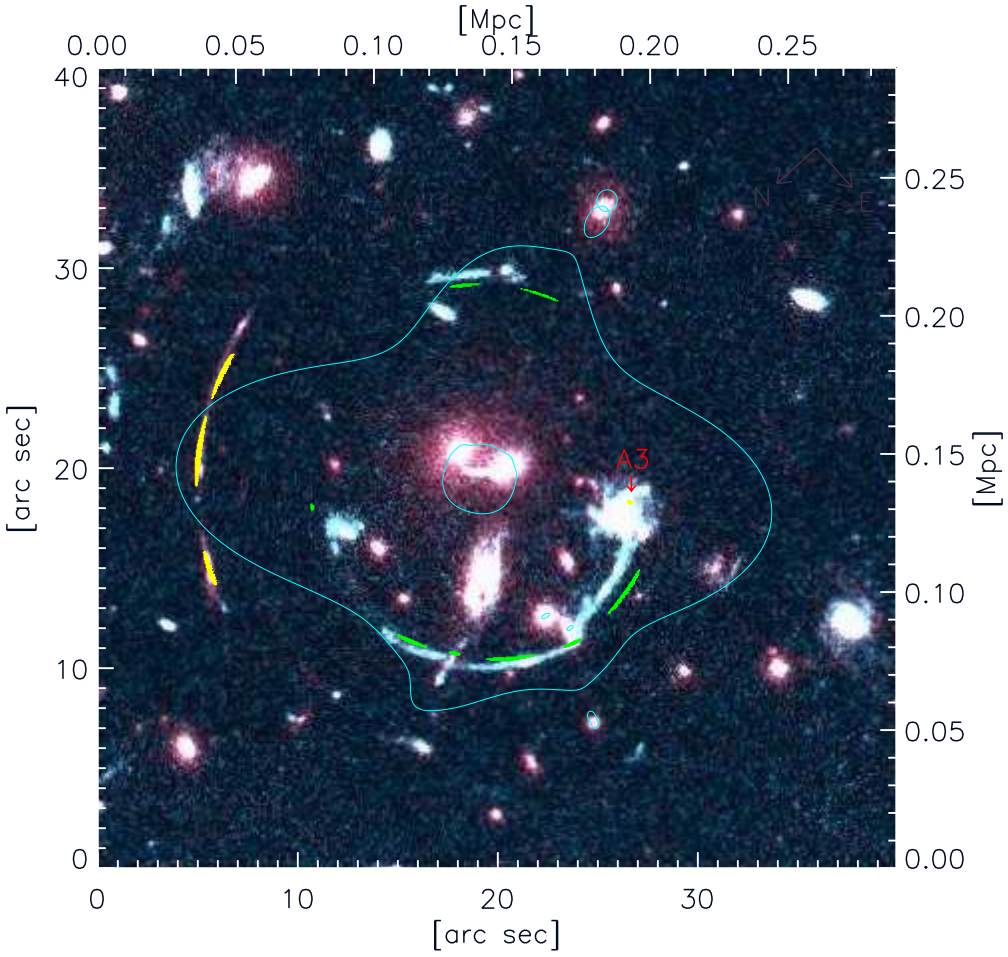

Figure 4.13: *Images reproduced by our best fit NFW model over-plotted on the combined F606W/F814W WFPC2 HST image. The closed lines are the critical curves for a source at* $z = 4.87$. *The center of the image is RA: 02:24:34.218 Dec: -00:02:31.64*

4.9.3 Two Isothermal Ellipsoids with Four Sources

Since the redshifts of arcs B1, B2, B3, B4 and central feature C is not known it is possible, that those features belong to separate arcs systems. The best fit model based on two isothermal ellipsoids and the substructure centered on bright galaxy members is presented in Fig. 4.18. This lens has been however, obtained using different constrains than the previous ones. The arcs B1 and B2 were assumed to be one system, B3 and B4 the second, A1, A2, and A3 – the third, and finally the feature C – the fourth.

The model reproduces the arc system A very well. The arc B2 is also precisely recre-

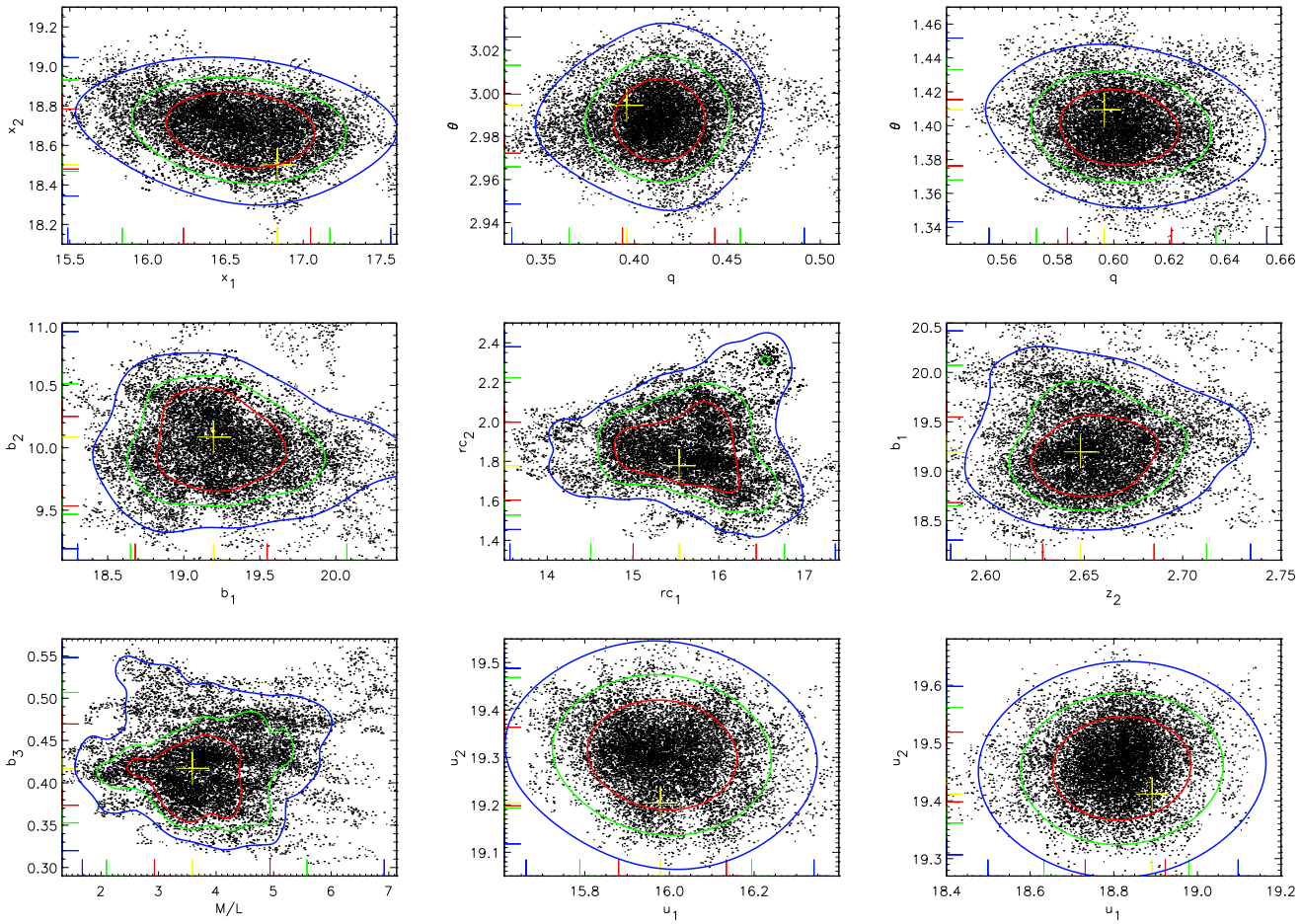

Figure 4.14: *MCMC error estimates. The contours correspond to 68%, 90% and 99% confidence levels. Marks on vertical and horizontal axis give the same confidence levels for 1D projected variables. The cross marks the position of the best fit point.*

ated, B1 however is shifted by ≈ 1 arc second with respect to the observed one. The system B3-B4 is accurately reproduced, together with an additional image that falls within arc B1. The central arc C is recreated, as well as three not(clearly) visible images. In summary this parameterization suffers from creation of additional images that are not observed. The total mass of the cluster predicted by this model is $5.9 \times 10^{13} \mathrm{M}_{\odot}$, the virial radius $R_{200} = 0.35 \mathrm{Mpc}$, and the redshifts of arcs systems B1-B2: 2.24, B3-B4: 2.59 and C: 2.41.

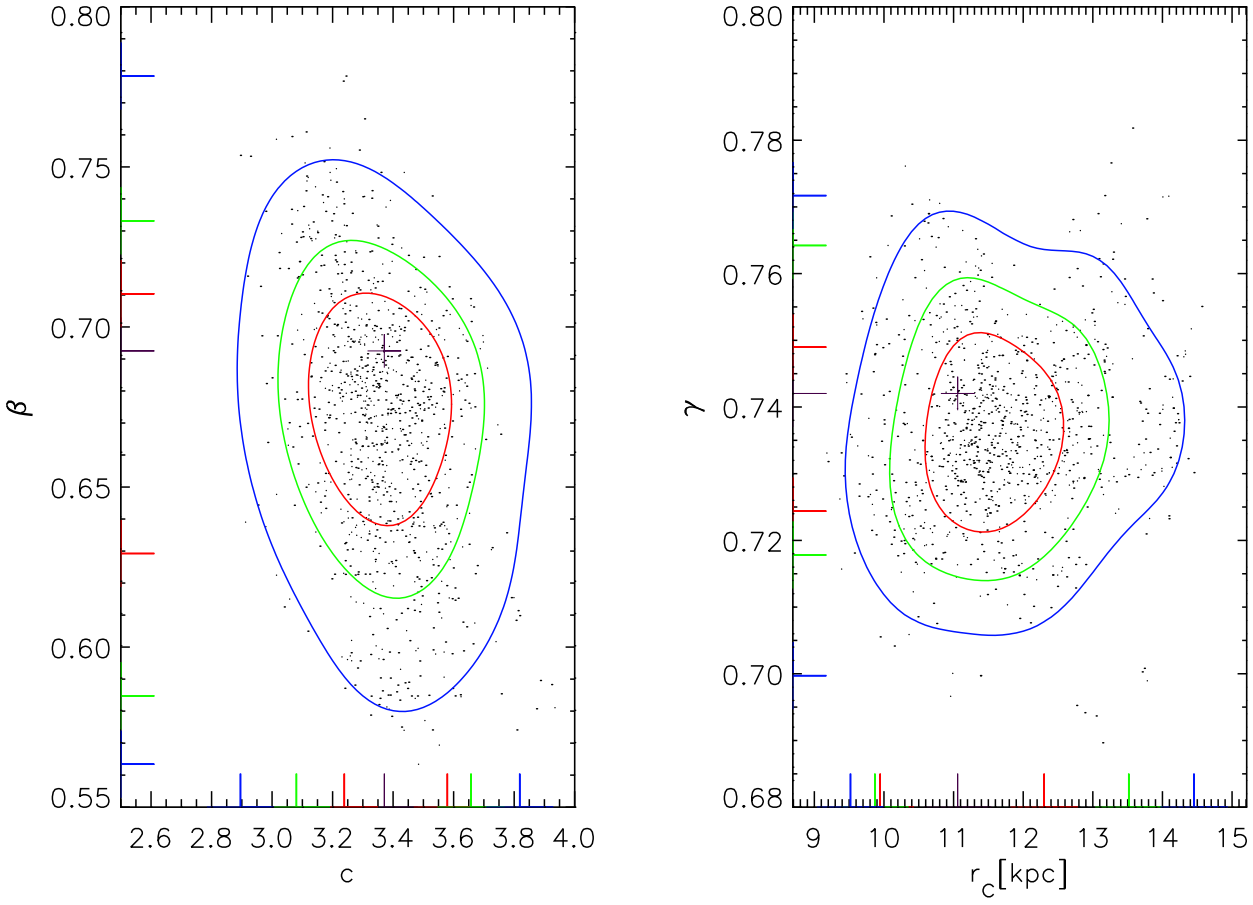

Figure 4.15: *MCMC error estimates of the parameters of the single power law and NFW-like profiles fit. The contours correspond to 68%, 90% and 99% confidence levels. Marks on vertical and horizontal axis give the same confidence levels for 1D projected variables. The cross marks the position of the best fit point.*

4.10 Preliminary Results from Genetic Algorithm Minimization

This section presents preliminary results obtained, when genetic algorithm (Sect. 3.4) was used as a minimization method. Since I have implemented this scheme recently, I did not gain much experience in using it yet. The first results presented in Tab. 4.4 are however very promising, especially that the cpu time needed to run genetic algorithm minimization was about one third of the time needed by the Powell algorithm. The

Table 4.3: *Images reproduced under image plane – source plane – image plane mapping*

Image	Counter Images Reproduced Images
A2 	A1 A3
	  x2
	  x2
B1 	B2 B3 B4
	  
	  

First column shows images used to construct sources. Second column shows both original and model reproduced images.

average Hausdorff distance between model produced, and data arcs seen in Tab. 4.4 is 350. This is ten times more then the result of the best model obtained using Powell algorithm in Sect **??**. The performance of the genetic method will be however improved by better adjustment of various parameters (number of generations, mutation probability, crossover place etc...). It is also worth to investigate the possibility of combining the genetic algorithms with MCMC method.

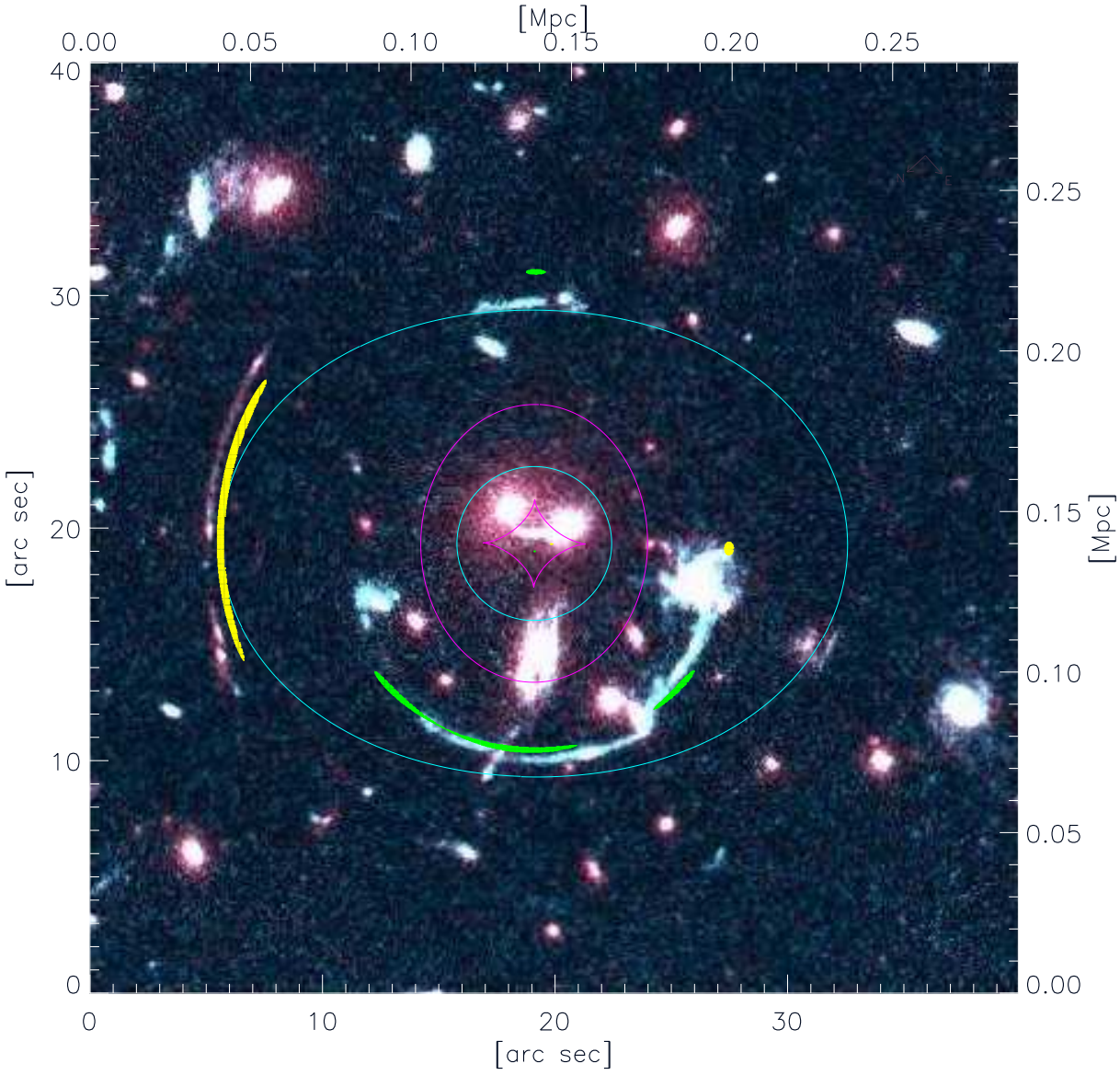

Figure 4.16: *The best fit model based on only one isothermal ellipsoid and no substructure*

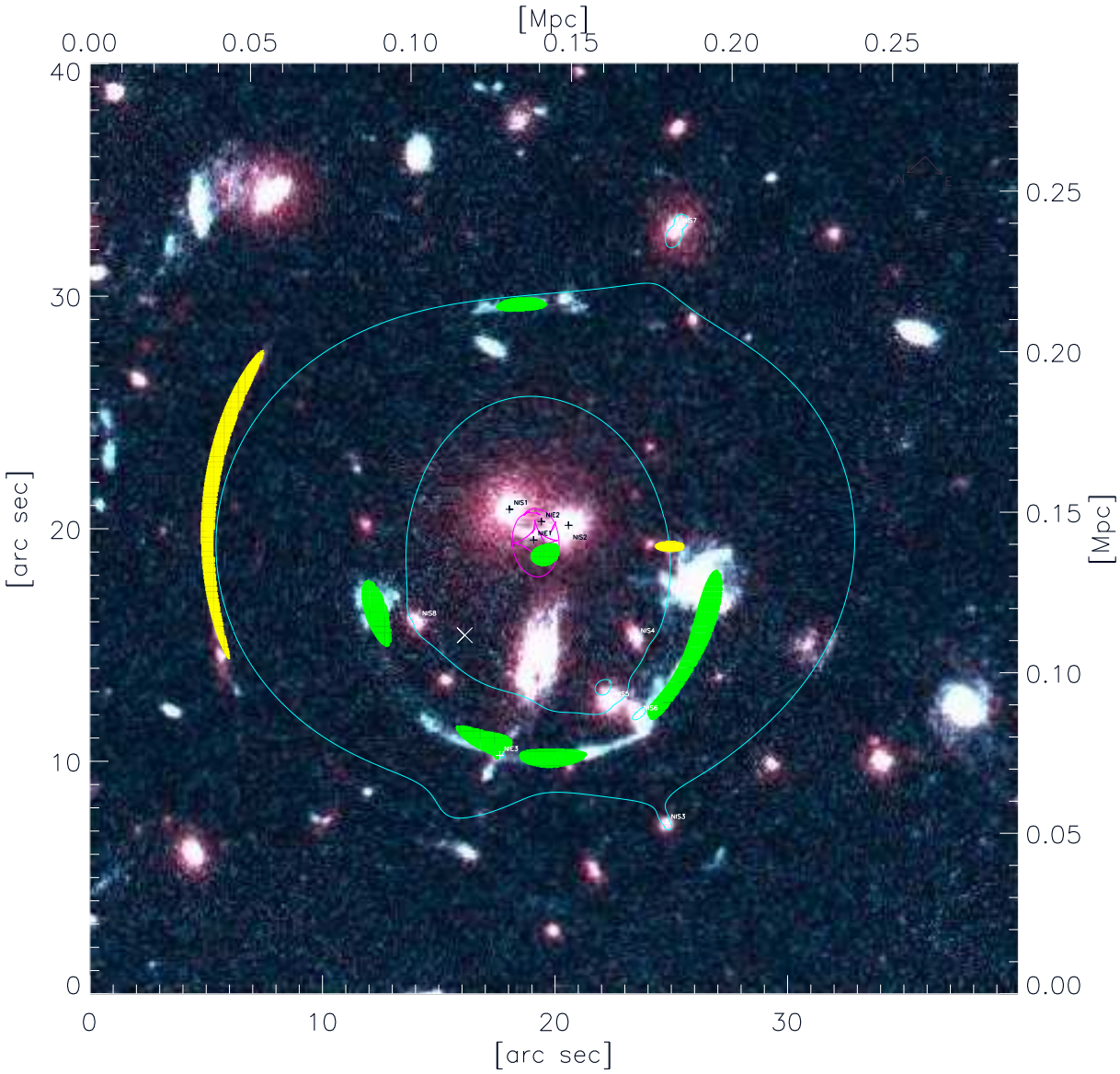

Figure 4.17: *The best fit model based on only one isothermal ellipsoid and substructure centered on bright cluster members*

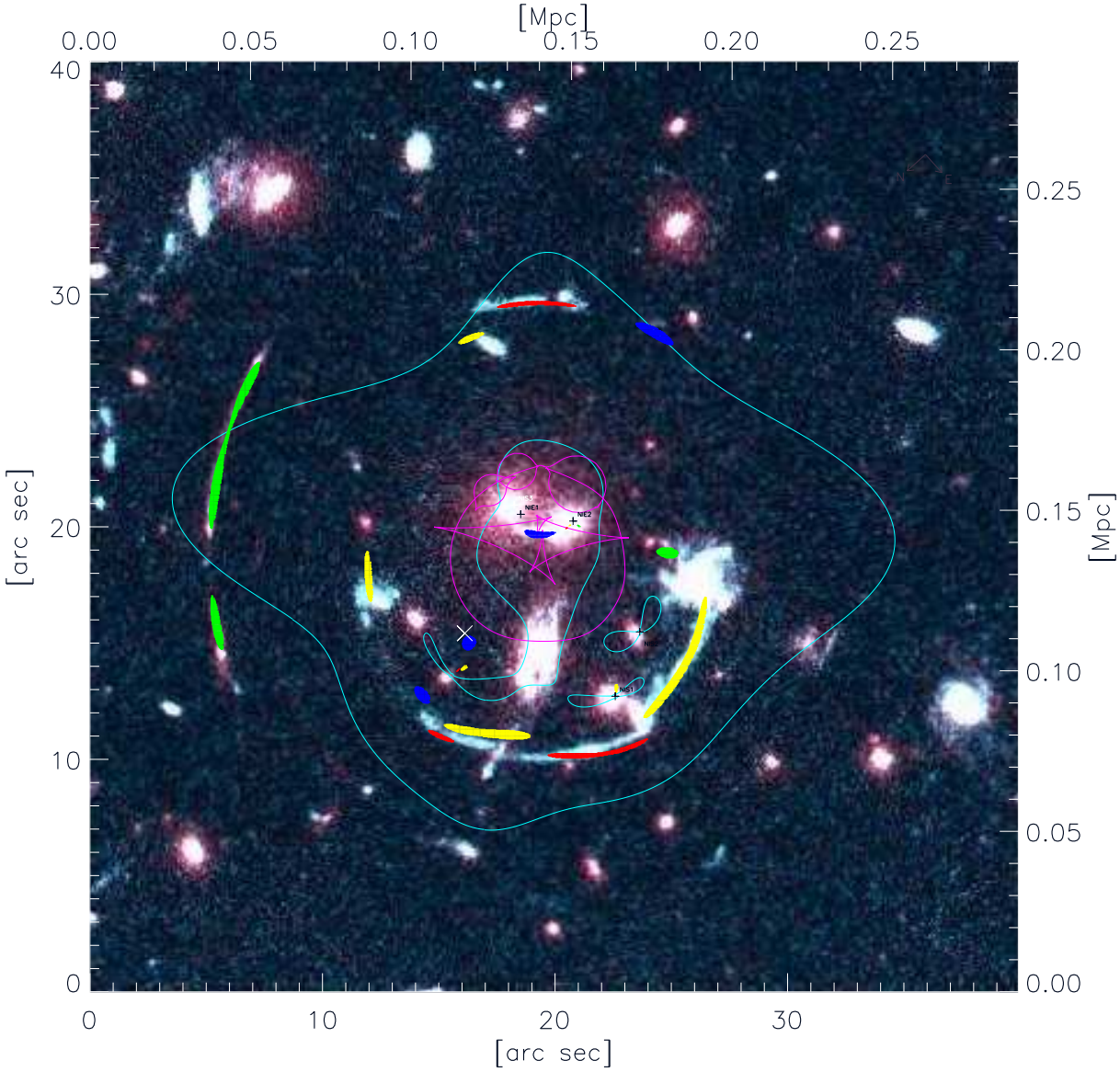

Figure 4.18: *The best fit model based on two isothermal ellipsoids, substructure, and four distinct arc systems*

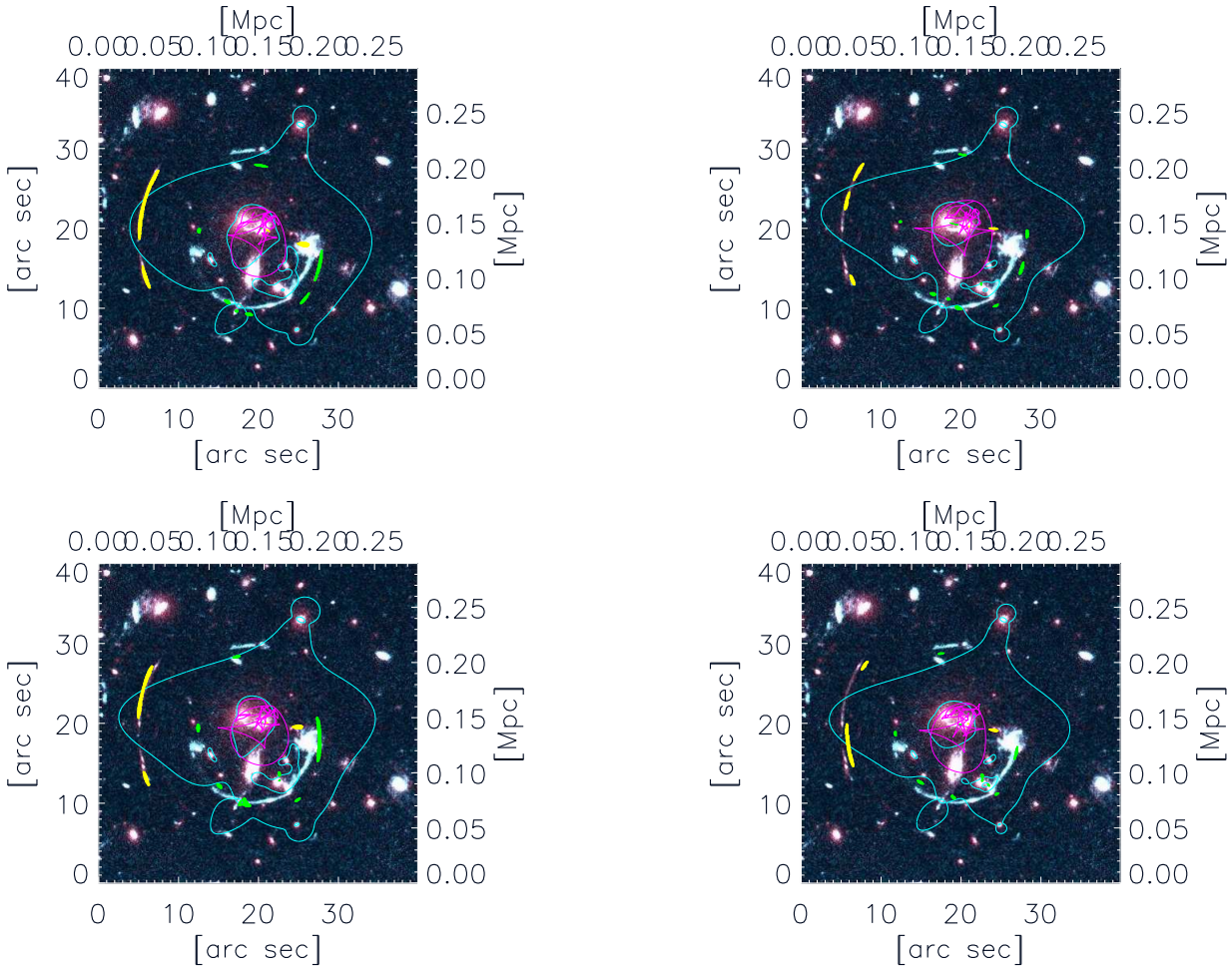

Table 4.4: *Four examples of the genetic algorithm based minimization. The model produced arcs, caustics and critical curves are over-plotted on the RCS0224-0002 cluster data.*

Published as: J. Rzepecki, M. Lombardi, P. Rosati, A. Bignamini, P. Tozzi – *"Strong Lensing Analysis of the Cluster RCS0224-0002,"* 2007A&A...471..743R.

Chapter 5

The Summary

The goal of my PhD project was to develop a general framework for modeling of a matter distributions in cosmological objects (especially galaxy clusters) based on the observed strong lensing features. The created software package has been build around the parametric approach described in Sect. 2.2, and makes use of advanced minimization algorithms presented in Chap. 3. The three step model construction, described in Chap. 4, has proved to be a very efficient and robust way of inverting the lensing problem. The first step – source size minimization (Sect. 2.3.2) gives the good first approximation of the solution, which is then a starting point for the second step – minimization of the Hausdorff distance between observed and model produced arcs in the image plane. Finally the solution space is explored by the Monte Carlo Markov Chains to give an estimated likelihood of the best solution (Sect. 3.3).

The novel method of using a Modified Hausdorff Distance as a goodness-of-fit function (Sect. 2.3.3) has proved to be a very elegant and successful method to make use of the information about matter distribution encoded in the shapes of giant arcs. This metric is general enough to be easily used by other strong/weak lensing reconstruction algorithms, that would greatly benefit from it.

The developed method was successfully applied to the galaxy cluster RCS0224-0002. The modeled mass distribution was capable of reproducing in great details, all the strong lensing features observed in this high-redshift cluster (see Chap. 4). The shapes and morphologies of all the arcs has been accurately reconstructed, and the prediction about the unknown redshift of one of the arcs systems has been given. The properties of the reconstructed mass profile show interesting features, including a very flat core (see Sect. 4.6). The substructure centered on the bright galaxy members has been crucial for accurate reconstruction of observed strong lensing features. Obtained radial average

profile is significantly different from both NFW and Isothermal profiles, proving that the galaxy clusters can not, in general, be modeled by those simple density distributions. The MCMC analysis (see Sect. 4.7) showed that most of the model parameters are well constrained. The analysis has also showed a significant (5 arc seconds) shift between X-ray emission and the peak of mass profile – this should be further investigated.

Published as: J. Rzepecki, M. Lombardi, P. Rosati, A. Bignamini, P. Tozzi – *"Strong Lensing Analysis of the Cluster RCS0224-0002,"* 2007A&A...471..743R.

Appendix A

Software

Power does not corrupt. Fear corrupts... perhaps the fear of a loss of power.

John Steinbeck

Abstract

This appendix presents the documentation for the software package that implements the strong lensing modeling methods presented in this thesis. The package consists of two versions, one written in IDL and one written in C++. Both versions have similar functionality, the IDL version however also provides a GUI. For simplicity all the classes diagrams in this appendix refer to the C++ version.

A.1 Introduction

The software package implements all the methods presented in this thesis. The package consists of two versions, one written in IDL and one written in C++. Both versions have similar functionality, the IDL version however also provides a GUI. For simplicity all the classes diagrams in this appendix refer to the C++ version. The package has four main components:

- Classes to support model construction
- Classes to support data holding
- Classes to support the minimization methods
- Helper functions and utility functions

- GUI[1]

A.2 Classes to support model construction

The classes described here are meant to support the creation of the composite lensing model. The most top object is the *LensesContainer*, which has a collection of *LensesSystem* objects and an interface calculate the properties of the model (deflection angle, mass, viral radius etc...). The *LensesContainer* is presented in Fig. A.1, and it has

LensesContainer
- mLensesSystemVector : std::vector< LensesSystem * > - mpPointsX : std::vector< double >* - mpPointsY : std::vector< double >* - mMapAnglesXtoSystemId : std::map< int, std :: vector < double > > - mMapAnglesYtoSystemId : std::map< int, std :: vector < double > >
+ LensesContainer() + ~ LensesContainer() + AddLensesSystem(pLensesSystem : LensesSystem*) + ComputeDeflectionAngle() + SetPoints(new_PointsX : std::vector< double >*, new_PointsY : std::vector< double >*) + GetAnglesX(LensesSystemId : int) : std::vector< double >* + GetAnglesY(LensesSystemId : int) : std::vector< double >* + GetLensesSystemVector() : std::vector< LensesSystem * >*

Figure A.1: *The LensesContainer class.*

following methods

- *AddLensesSystem* – adds a new *LensesSystem* object to the container
- *ComputeDelflectionAngle* – calculates the total deflection angle of all the componenets of the container
- *SetPoints* – sets the points (x, y) at which the deflection angle should be calculated
- *GetAnglesX* – returns the x component of the calculated deflection angles
- *GetAnglesY* – returns the y component of the calculated deflection angles

[1] implemented only in the IDL version

- *GetLensesSystemVector* – returns all the *LensesSystem* objects of the container

Since, *LensesSystem* can (and should) have different redshifts then all the methods that calculate the physical properties of the lens work on the *LensesSystem* level and not on the *LensesContainer* level.

The *LensesSystem* object is the container for all the lenses belonging to one "system" – all those lenses should have the same redshift. Therefore usualy we will have only one system, however if we have for example two clusters in the line of sight, then we will define one *LensesSystem* for the first cluster and the second *LensesSystem* for the second cluster. The *LensesSystem* is presented in Fig. A.2,

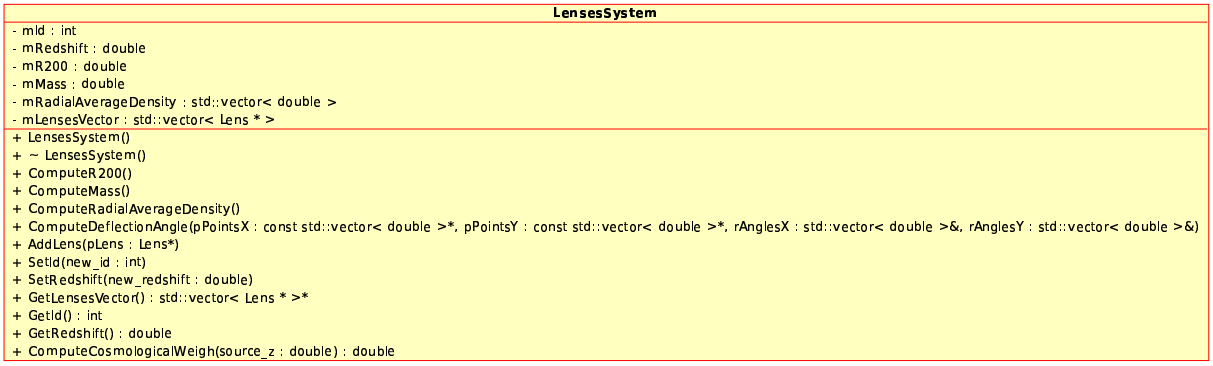

Figure A.2: *The LensesSystem class.*

and has following methods

- *ComputeR200* – computes the R_{200} of the system
- *ComputeMass* – computes the mass of the system within R_{200}
- *ComputeRadialAverageDesnity* – computes the radial average density profile of the system
- *ComputeDeflectionAngle* – calculates the total deflection angle caused by this system. The deflection angle is calculated at points *PointsX* and *PointsY* and it is stored in *AnglesX* and *AnglesY*
- *AddLens* – adds a *Lens* object to the system

- *SetId* – sets a unique Id of this system
- *SetRedshift* – sets the redshift of this system
- *GetLensesVector* – returns the vector of all the *Lenses* objects belonging to this system
- *GetId* – returns the Id of this system
- *GetRedshift* – returns the redshift of this system
- *ComputeCosmologicalWeight* – calculates the cosmological weight of a given *Source* in respect to this system

The *Lens* class is the parent for all the specific realizations of the parametric models. It provides a common interface to interact with all the models. Therefore all the parametric models should inherit from this base call and implement all its virtual methods. The *Lens* class is presented in Fig. A.3 and provides the following interface:

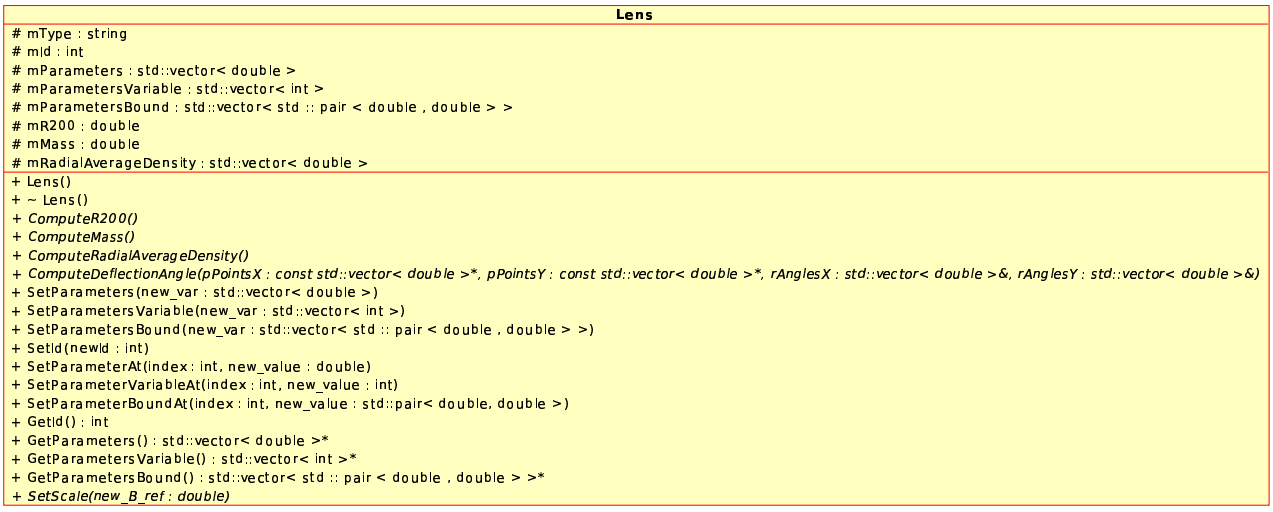

Figure A.3: *The Lens class.*

- *ComputeR200* – computes the R_{200} of the model
- *ComputeMass* – computes the mass of the model within R_{200}

- *ComputeRadialAverageDesnity* – computes the radial average density profile of the model
- *ComputeDeflectionAngle* – calculates the deflection angle caused by this model. The deflection angle is calculated at points *PointsX* and *PointsY* and it is stored in *AnglesX* and *AnglesY*
- *SetParameters* – sets the vector of parameters defining the model (ellipticity, position angle etc...)
- *SetParametersVariable* – defines which of the parameters should be allowed to change during the minimization process (0 – fixed, 1 – changeable)
- *SetParametersBound* – defines the bounding limits for the changeable parameters
- *SetId* – sets an unique id for this lens
- *SetParametersAt* – changes the parameter at a given position in parameters vector
- *SetParametersVariableAt* – changes the variability of a parameter at a given position in *mParametersVariable*
- *SetParametersBoundAt* – sets the bounding limits for a parameter at a given position in *mParametersBound*
- *GetId* – returns the id of this lens
- *GetParameters* – returns a vector of parameters describing this lens
- *GetParametersVariable* – returns a vector which defines which parameters are allowed to change during the minimization process
- *GetParametersBound* – returns the vector defining the bounding limits for this lens parameters
- *SetScale* – sets the new "scale" for the lens. It is equivalent with *SetParametersAt(2,scale)*

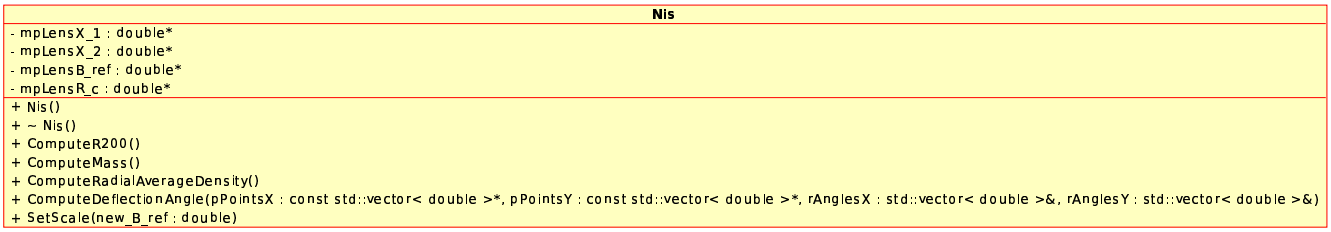

Figure A.4: *The NIS class.*

The *NIS* class implements the Non-Singular Isothermal Sphere lens model. It is derived from the basic *Lens* class and is presented in the Fig. A.4. It has the following methods

- *ComputeR200* – computes the R_{200} of the NIS model
- *ComputeMass* – computes the mass of the NIS model within R_{200}
- *ComputeRadialAverageDesnity* – computes the radial average density profile of the NIS model
- *ComputeDeflectionAngle* – calculates the deflection angle in the NIS model. The deflection angle is evaluated at points *PointsX* and *PointsY* and it is stored in *AnglesX* and *AnglesY*
- *SetScale* – sets a new "scale" for the lens. It is equivalent to *SetParametersAt(2,scale)*

The *NIE* class implements the Non-Singular Isothermal Ellipsoid lens model. It is derived from the basic *Lens* class and is presented in the Fig. A.5. It has the following

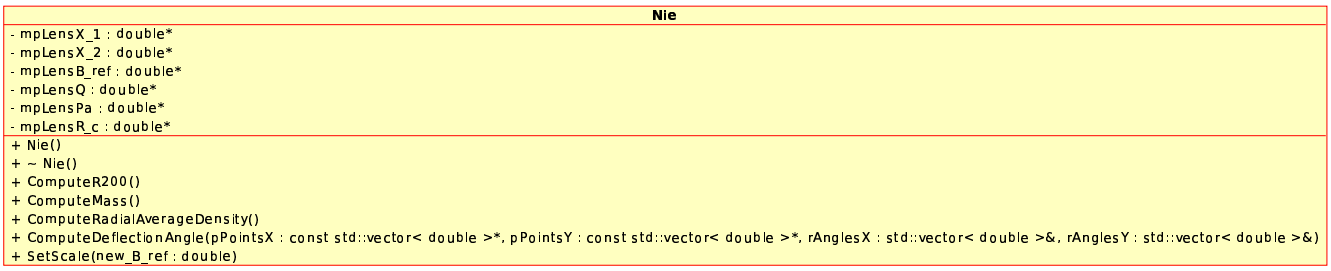

Figure A.5: *The NIE class.*

methods

- *ComputeR200* – computes the R_{200} of the NIE model
- *ComputeMass* – computes the mass of the NIE model within R_{200}
- *ComputeRadialAverageDesnity* – computes the radial average density profile of the NIE model
- *ComputeDeflectionAngle* – calculates the deflection angle in the NIE model. The deflection angle is evaluated at points *PointsX* and *PointsY* and it is stored in *AnglesX* and *AnglesY*
- *SetScale* – sets a new "scale" for the lens. It is equivalent to *SetParametersAt(2,scale)*

The *FixedScaleRatioGroup* class supports the substructure realized by a set of isothermal spheres with positions fixed on the cluster galaxy members. The scale of each isothermal sphere is proportional to the brightness of the galaxy it is centered on. Therefore the whole substructure is parameterized by one global parameter corresponding to the M/L ratio. The class is presented in Fig. A.6 and implements the following methods

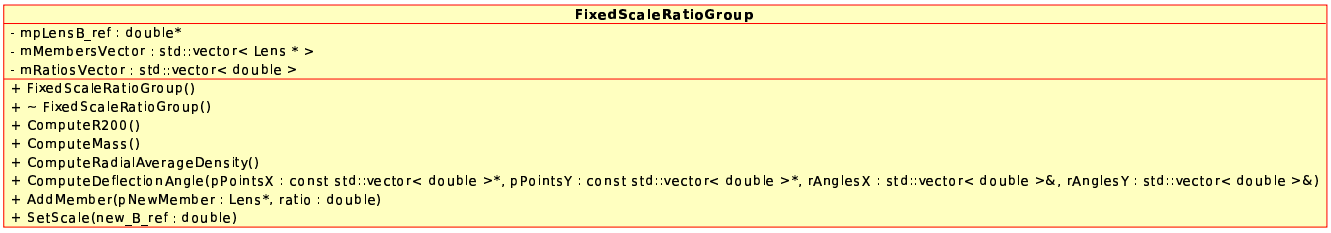

Figure A.6: *The FixedScaleRatioGroup class.*

- *ComputeR200* – computes the R_{200} of the model
- *ComputeMass* – computes the mass of the model within R_{200}
- *ComputeRadialAverageDesnity* – computes the radial average density profile of the model
- *ComputeDeflectionAngle* – calculates the deflection angle in the model. The deflection angle is evaluated at points *PointsX* and *PointsY* and it is stored in *AnglesX* and *AnglesY*

- *SetScale* – sets the new "scale" for the whole group. It is equivalent to *SetParametersAt(2,scale)*
- *AddMember* – adds a new isothermal sphere member to the group

The above methods make use of the NIS implementation.

Other models including NFW and King Model are implemented in the IDL version of the software. Implementation of a new model in the C++ version requires the creation of a model class derived from the basic *Lens* class and providing all the required methods. One can use the new model by adding it to the *LensesSystem* container.

The sources are organized is a similar fashion as the lenses. The *SourcesContainer* class is the global container that holds all the sources. Each particular source object is derived from the basic *Source* class. The *SourcesContainer* class is presented in Fig. A.7 and has the following methods

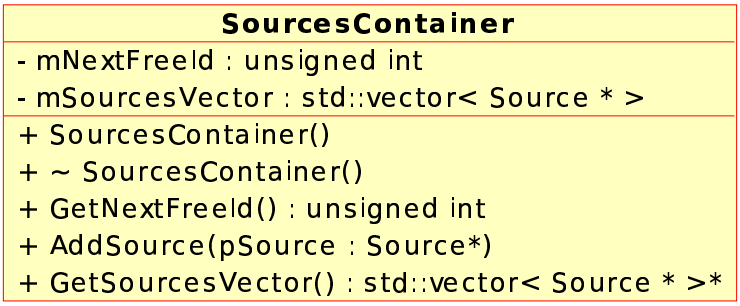

Figure A.7: *The SourcesContainer class.*

- *GetNextFreeIs* – returns the next free id of the source
- *AddSource* – adds a new source to the container
- *GetSourcesVector* – returns a vector of all the sources in the container

The basic *Source* class is used to derive all the realizations of the sources. It is presented in the Fig. A.8 and provides following interface

- *IsWithinSource* – virtual method that need to be implemented by the derived source model. Defines whether a given pixel in the source plane is within the source or not.

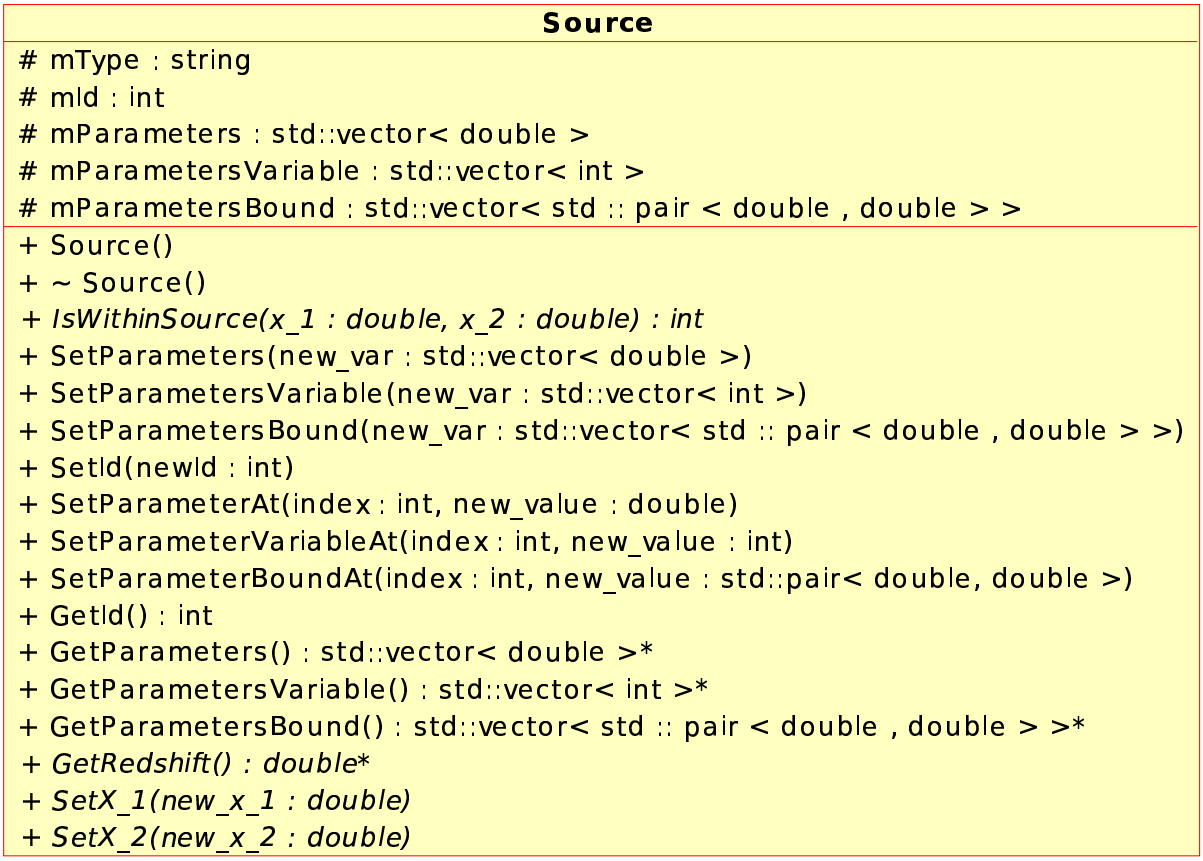

Figure A.8: *The Source class.*

- *SetParameters* – sets the vector of parameters defining the source (position, radius, etc...)
- *SetParametersVariable* – defines which of the parameters should be allowed to change during the minimization process (0 – fixed, 1 – changeable)
- *SetParametersBound* – defines the bounding limits for the changeable parameters
- *SetId* – sets an unique id for this source
- *SetParametersAt* – changes the parameter at a given position in parameters vector
- *SetParametersVariableAt* – changes the variability of a parameter at a given position in *mParametersVariable*

- *SetParametersBoundAt* – sets the bounding limits for a parameter at a given position in *mParametersBound*
- *GetId* – returns the id of this source
- *GetParameters* – returns a vector of parameters describing this source
- *GetParametersVariable* – returns a vector which defines which parameters are allowed to change during the minimization process
- *GetParametersBound* – returns the vector defining the bounding limits for this source parameters
- *GetRedshift* – (virtual) gets the redshift of the source (equivalent to getting the value of the second parameter)
- *SetX_1* – (virtual) sets the x_1 coordinate of the source position (equivalent to getting the value of the zeroth parameter)
- *SetX_2* – (virtual) sets the x_2 coordinate of the source position (equivalent to getting the value of the first parameter)

The *SourceCircular* class is a realization of the source. It represents a disk with a given radius and positioned at a given position – it is the most simple not point-like source. It is presented in Fig. A.9 and implements the virtual methods of the basic *Source* class

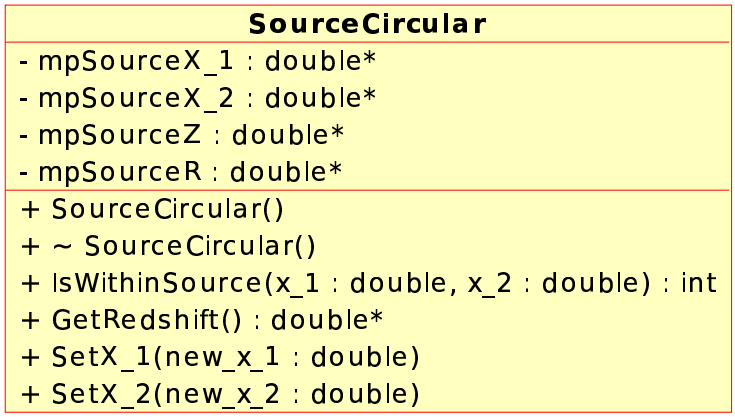

Figure A.9: *The SourceCircular class.*

- *IsWithinSource* – method that need to be implemented by the derived source model. Defines whether a given pixel in the source plane is within the source or not.
- *GetRedshift* – gets the redshift of the source (equivalent to getting the value of the second parameter)
- *SetX_1* – sets the x_1 coordinate of the source position (equivalent to getting the value of the zeroth parameter)
- *SetX_2* – sets the x_2 coordinate of the source position (equivalent to getting the value of the first parameter)

A.3 Classes to support data holding

The *DataContainer* class is used to read in and store data about available arcs – their positions, redshifts, and shapes. It is accompanied by a *ArcsSystem* class used to represent each arc system – all the images of the same source. The *ArcsSystem* class is presented in the Fig. A.10. Its purpose is to hold the x_1 and x_2 coordinates of all the pixels belonging to one arc system. Since each arc system is binded to one source, then the constructor of *ArcsSystem* takes a *Source* as a parameter. *ArcsSystem* class provides the following methods

- *GetImagesX* – returns the vector of x_1 coordinates of all the pixels belonging to the arc system
- *GetImagesY* – returns the vector of x_2 coordinates of all the pixels belonging to the arc system
- *GetRedshift* – returns the redshift of this arc system
- *GetSigma* – gets the error of the position measurement of this arc system
- *GetSource* – returns the source responsible for creation of this arc system
- *GetId* – gets the unique id of this arcs system

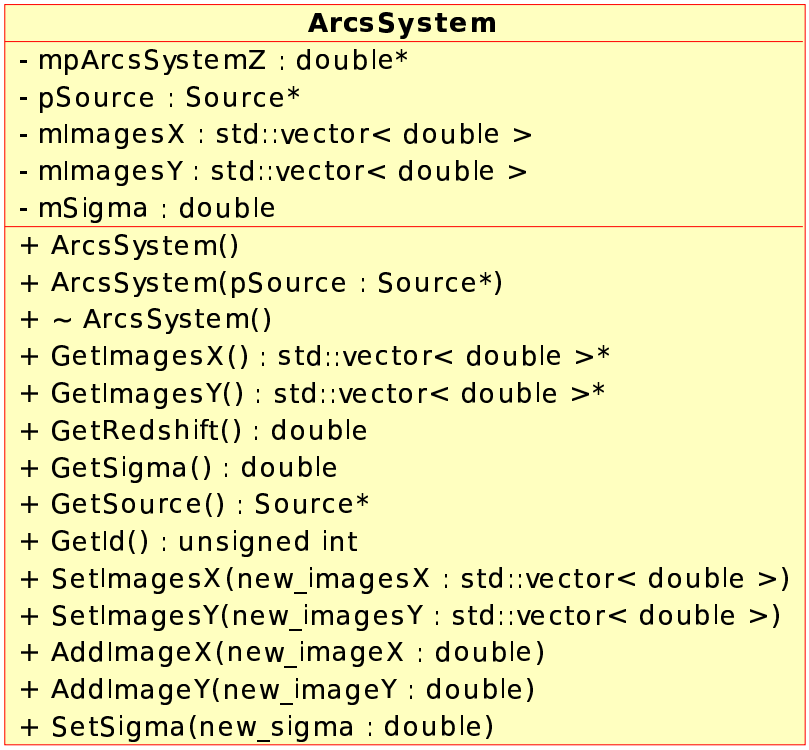

Figure A.10: *The ArcsSystem class.*

- *SetImagesX* – sets the x_1 coordinates of all the pixels belonging to this arc system
- *SetImagesY* – sets the x_2 coordinates of all the pixels belonging to this arc system
- *AddImageX* – adds a new pixel to this arc system (x_1 coordinate)
- *AddImageY* – adds a new pixel to this arc system (x_2 coordinate)
- *SetSigma* – sets the error of position measurement for this arc system

The *DataContainer* class holds information about arc systems – both extended and point like. It also provides a method to read in the information about extended arcs from a png file. The class is presented in the Fig. A.11 and has the following methods

- *GetArcsSystemVector* – returns a vector of all the available arcs systems (point like)
- *GetArcsSystemExtendedVector* – returns a vector off ale extended arcs systems
- *AddArcsSystem* – adds a new point like arc system

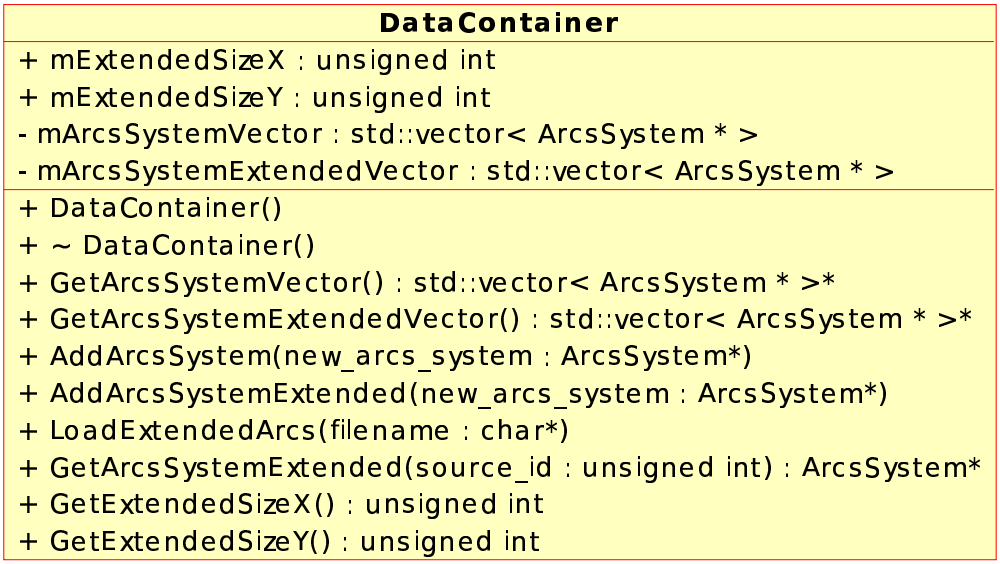

Figure A.11: *The DataContainer class.*

- *AddArcsSystemExtended* – adds a new extended arc system
- *LoadExtendedArcs* – reads in the informations about extended arc systems from a png file. The file should be in indexed mode and each index (color) should correspond to an arc system
- *GetArcsSystemExtended* – returns an arc system binded to a given source
- *GetExtendedSizeX* – returns the x dimension of the read in file
- *GetExtendedSizeY* – returns the y dimension of the read in file

A.4 Classes to support the minimization methods

The classes described here are meant to provide an interface to the model fitting process. In order to work they need an input data provided by the *DataContainer* class, and the model – *LensesContainer* and *SourcesContainer*. There are two minimization classes derived from the base *Minimizer* class. The first one, *MinimizerPointsSourcePlane*, performs the model fitting under the assumption that the sources (and images) are point-like. The second one, *MinimizerExtendedImagePlane*, performs the model fitting based on the

full information about arc shapes and make use of the Modified Hausdorff Distance as a metric for fitting. Each minimizer class can use a different algorithm for finding a minimum. Currently the *Powell*, *MCMC* and *Genetic*[2] methods are implemented.

The basic *Minimizer* class is presented in Fig. A.12 and it has the following methods

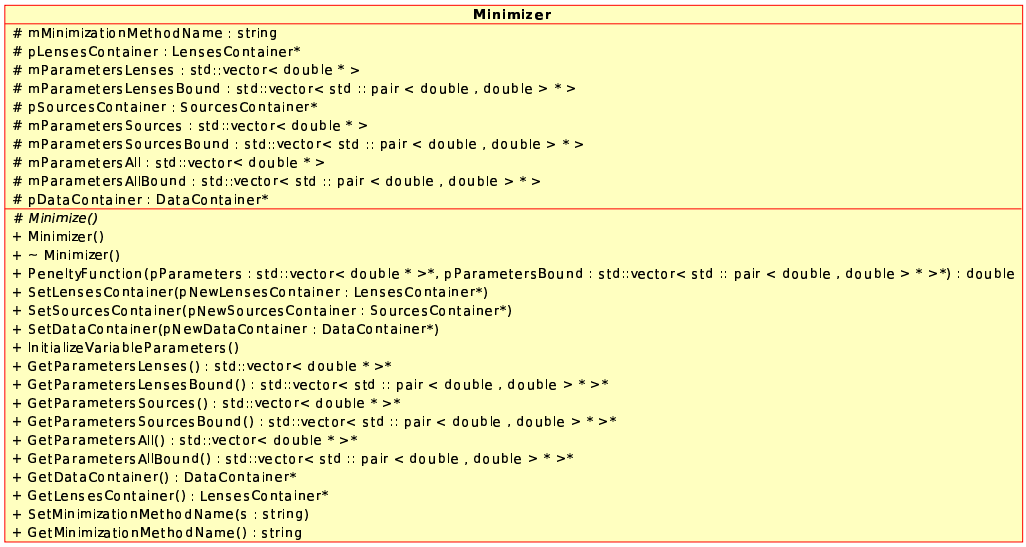

Figure A.12: *The Minimizer class.*

- *Minimize* – a pure virtual method implemented in all derived classes
- *PeneltyFunction* – the function used to bound parameters to certain intervals.
- *SetLensesContainer* – sets the lenses container (the lenses model)
- *SetSourcesContainer* – sets the sources container
- *SetDataContainer* – sets the data container
- *InitializeVariableParameters* – construct a vector of all variable parameters and corresponding bounding vectors
- *GetParametersLenses* – returns a vector of the parameters describing the lens model
- *GetParametersLensesBound* – returns the bounds vector for the lens parameters

[2]implemented only in the IDL version

- *GetParametersSources* – returns a vector of the parameters describing the sources
- *GetParametersSourcesBound* – returns the bounds vector for the sources parameters
- *GetAllParameters* – returns a composite vector of all the parameters (lenses + sources)
- *GetAllParametersBound* – returns a bounds vector of the composite parameters (lenses + sources)
- *GetDataContainer* – returns the data container
- *GetLensesContainer* – returns the lenses container
- *SetMinimizationMethodName* – sets the name of the algorithm to be used for the minimization ("Powell" or "MCMC")
- *GetMinimizationMethodName* – returns the name of the minimization algorithm to be used

The *MinimizerPointsSourcePlane* is a class derived from *Minimizer* that implements the minimization described in Sect. 2.3.2. It is presented in Fig. A.13 and implements

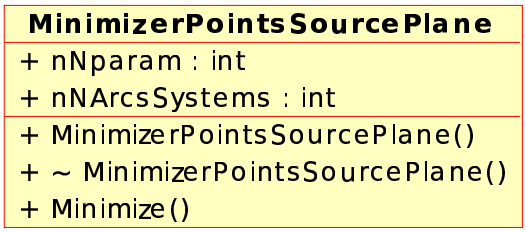

Figure A.13: *The MinimizerPointsSourcePlane class.*

the *minimize* method.

The *MinimizerExtendedImagePlane* is a class derived from *Minimizer* that implements the minimization described in Sect. 4.5.3. It is presented in Fig. A.14 and implements the *minimize* method.

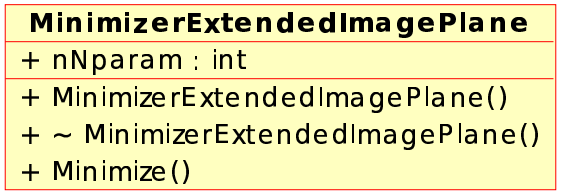

Figure A.14: *The MinimizerExtendedImagePlane class.*

A.5 Helper functions and utility functions

The package requires also some additional functions and tools that will be described in this section. Those are usually small specialized routines dedicated to perform a specific task.

- *mcmc* – contains routines needed to perform MCMC analysis of a function
- *ToolBox* – contains various useful routines
  - *DistanceAngular* – calculates the angular distance to an object at a given redshift
  - *DistanceLuminosity* – calculates the luminosity distance to an object at a given redshift
  - *MagMatrix* – calculates the magnification matrix at a given position in lens plane

A.6 GUI

Before starting the GUI, one needs to define the model as a array of *lens* structures – *lenses*, and array of *source* structures – *sources*. In addition the background image and header file of the fits file corresponding to the background image are needed. Then one can start the GUI by committing an command:

gui_main, lenses, sources, dimx, dimy, resolution, xscale, yscale, background_image, header

In the above statement the *dimx* and *dimy* are the x and y dimensions of the background image in pixels, *resolution* is the resolution in which all the calculation will be carried out (e.g. resolution = 2 means that the positions of arcs will be calculated on the grid that has twice the resolution of the background image). *xscale* and *yscale* define a conversion factor from pixels to arc seconds in x and y direction. The path to the background image is given by *background_image*, and the *header* is the fits header. The Fig. A.15 shows the overview of the GUI. The window is splited into two main areas: the bigger one to

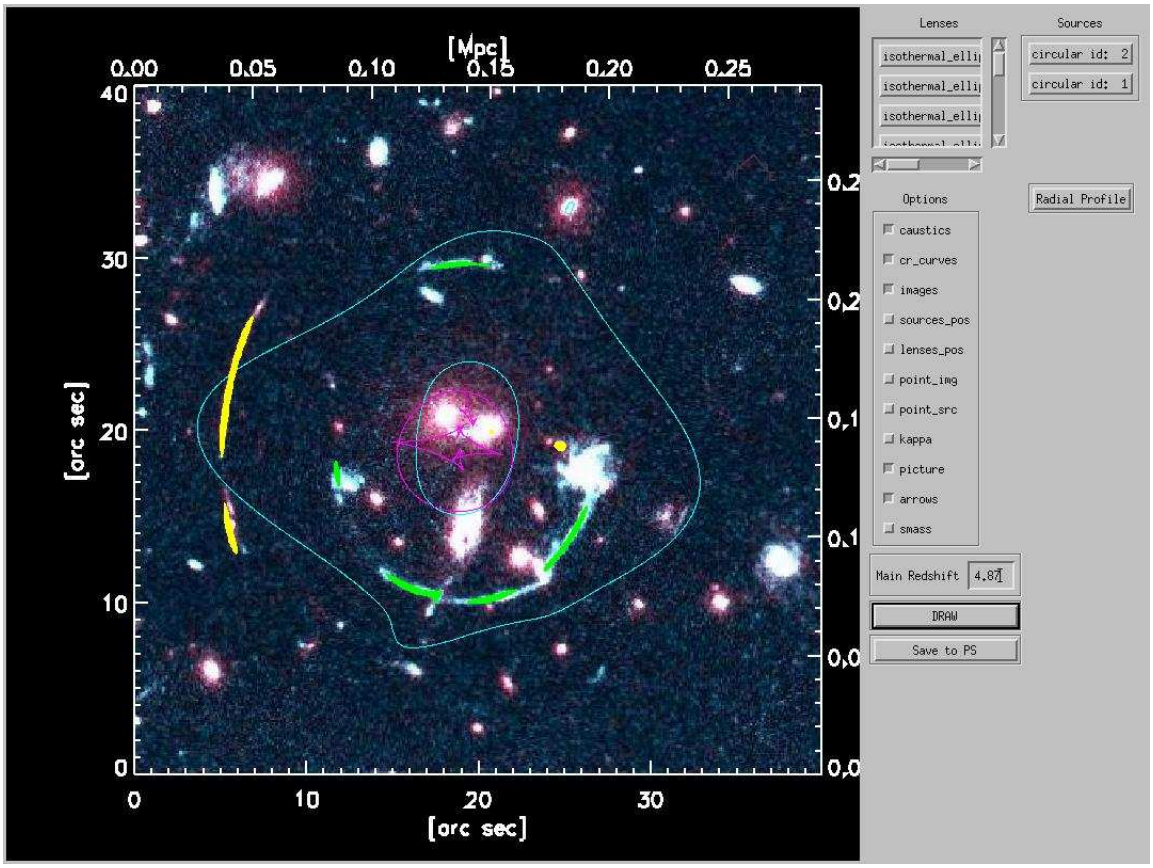

Figure A.15: *The general GUI overview*

the left is the plotting area, and the smaller one to the right is the control area. Most of the interaction is done thought the control area; the plotting area however, allows some degree of interaction as well.

The top of the control area has two lists – one for all the lenses defined for the model,

and one for all the sources. Choosing one of the members of those lists brings the window showed in Fig. A.16. This table allows one to modify of all the parameters of a given lens or source component.

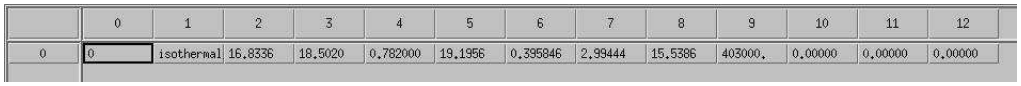

Figure A.16: *The window to modify the model parameters*

Below the lenses list there is an *Options* box. This box is used to enable/disable different plotting options. The following choices are possible

- *coustics*: Enable/disable plotting of caustics
- *cr_curves*: Enable/disable plotting of critical lines
- *images*: Enable/disable plotting of images (arcs)
- *sources_pos*: Enable/disable plotting of sources positions
- *lenses_pos*: Enable/disable plotting of center of lenses model positions
- *point_img*: Enable/disable plotting of point like images
- *point_src*: Enable/disable plotting of point like sources
- *kappa*: Enable/disable plotting of kappa contours
- *picture*: Enable/disable plotting of background image
- *arrows*: Enable/disable plotting of North/East arrows
- *smass*: Enable/disable plotting of the surface mass density

Since many of the components to be plotted depend on the redshift (caustics, kappa, etc...) then one need to define a redshift for which to plot them. This is done thought a *Main Redshift* box, below the *Options* box.

Once all the options are set up one can choose to draw the picture on the screen (*Draw* button) or save it in the PostScript file (*Save to PS* button).

Under the sources list there is a *Radial Profile* button. Pressing it brings a window showed in the Fig. A.17. This window is used to control and plot the radial average

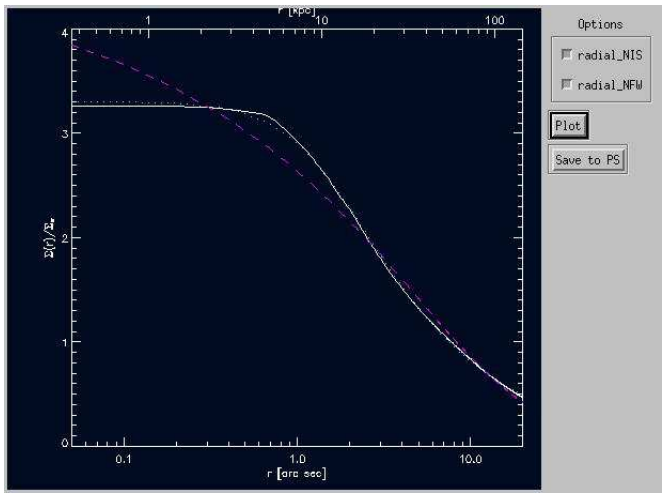

Figure A.17: *The window to plot the radial average mass profile*

profile of the mass distribution. It allows plotting (*Plot* button) or saving to a PostScript file (*Save to PS* button) of the radial average mass profile. One can also choose to over plot the best NIS and/or NFW fit to the profile.

The plotting area also allows a simple interaction. When a mouse click is registered in this area, then the lens that is centered closest to the mouse position is selected, and can be moved by clicking on the new position. This is useful when one investigates the impact of the substructure, or want to fine tune the model a little bit by hand.

A.7 Example program usage

In this section the usage of the package will be presented. The full process of finding a best fitting model to observed strong lensing features of the cluster RCS0224-002 will be discussed. Both the IDL and C++ code will be explained.

A.7.1 IDL Example

First we reset the session, load necessary paths, then we set the error sensitivity and finally we load a nice color table.

```
.Reset_Session
retall
!path=!path+path_to_general_idl_routines
!path=!path+path_to_tools
!path=!path+path_to_psconfig
!EXCEPT=2
loadct,39
```

Now we load the physical, astrophysical and cosmological constantes

```
constants                    ;define constants
COMMON COSMO_CONST
COMMON ASTRO_CONST
COMMON PHYS_CONST
```

Read background image, and fits header

```
cl_image = read_bmp("rcs0224-cut-indexed246.bmp", R,G,B)
h=headfits("rcs0224_red_cut.fits")
```

Set the pixel to arc seconds conversion factors

```
xscale = 0.0995528
yscale = 0.0995431
```

Then the source structure is defined, together with the array of two sources. The sources are given some starting values for the parameters. Also the variability and bound of parameters are set up.

```
;define the source structure
source = {source, id: 0, type: "circular", x_1: 182.132,
  x_2: 153.026, z: 1.255, ellipcity: 0.0, pos_angle: 0.0,
  p0: 0.05, p1:0.0, p2:0.0, p3:0.0, p4:0.0, color: 0}
;allocate a two element array for the two sources
sources = replicate(source,2)
```

```
;the first source
sources[0].id = 2          ;the id
sources[0].x_1=16.3027     ;the position (in arc seconds)
sources[0].x_2=19.3914
sources[0].z = 4.87860     ;the redshift
sources[0].p0=0.15         ;the radius
sources[0].color=251       ;color
var_s0 = [0]               ;set the redshift of this source as
                           ;fixed for minimization
                           ;(known spectroscopic redshift)
bound_s0 = [0.0,0.0]       ;ignore the bounds

;the second source
sources[1].id = 1          ;the id
sources[1].x_1=19.0781     ;the position
sources[1].x_2=19.5253
sources[1].z = 2.35084     ;the starting redshift
sources[1].p0=0.1          ;the radius
sources[1].color=248       ;the color
var_s1 = [1]               ;this redshift should be allowed to
                           ;change during minimization
bound_s1 = [1.0,8.0]       ;reasonable bounds for the redshift
```

Now the lensing model needs to be defined. It consists of 12 parametric lenses (NIEs and NISs). The starting values of the parameters are initialized and the bounds for variable parameters are set up.

```
;definition of the lens structure
lens = {lens, id: 0, type: "isothermal_ellipsoid_elliptical",
  x_1: 1.0, x_2:2.0, z: 0.782, b_ref:4.1857 , ellipcity: 0.75,
  pos_angle: 45.0*!Pi/180.0, p0:0.0, p1:0.0, p2:0.0, p3:0.0,
  p4:0.0}
```

```
;the lenses array
lenses = replicate(lens, 12)

;the first lens
;the id
lenses[0].id = 0
;the type
lenses[0].type="isothermal_ellipsoid_elliptical"
;the velocity dispersion
lenses[0].p1 = 403000.0D
;scale factor
lenses[0].b_ref = 10.0
;position angle
lenses[0].pos_angle = 170.0*!Pi/180.0
;ellipticity
lenses[0].ellipcity = 0.8
;the position
lenses[0].x_1 = 195.0
lenses[0].x_2 = 204.0
;the s parameter
lenses[0].p0 = 1.0
;the variability of parameters
;position, scale factor, position angle, ellipticity, and s
;will change during minimization
var0=[1,1,0,1,1,1,1,0,0,0,0]
;the bounds for variable parameters
bound0=[[15.0,25.0],[15.0,25.0],[0,0],[0,0],[0.3,0.99],[0,!Pi],
       [0,0],[0,0],[0,0],[0,0],[0,0]]

;the second lens
lenses[1].id = 1
```

```
lenses[1].type="isothermal_ellipsoid_elliptical"
lenses[1].p1 = 510000
lenses[1].b_ref = 5.0;
lenses[1].pos_angle = 1.52668
lenses[1].ellipcity = 0.7
lenses[1].x_1 = 195.0;181.2
lenses[1].x_2 = 204.0;209.3
lenses[1].p0 = 1.778;1.0
var1=[0,0,0,1,1,1,1,0,0,0,0]
bound1=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.3,0.99],[0,!Pi],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

;the third , fourth ,... lens
lenses[2].id = 2
lenses[2].type="isothermal_ellipsoid_elliptical"
lenses[2].p1 = 510000
lenses[2].b_ref = 0.0
lenses[2].pos_angle = 1.46156
lenses[2].ellipcity = 0.7
lenses[2].x_1 = 206.7
lenses[2].x_2 = 202.4
lenses[2].p0 = 1.0
var2=[0,0,0,0,0,0,0,0,0,0,0]
bound2=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.9,0.99],[0,!Pi],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[3].id = 193
lenses[3].type="isothermal_ellipsoid_circular"
lenses[3].p1 = 510000
lenses[3].b_ref = 2.0
lenses[3].pos_angle = 50.0*!Pi/180.0;
```

```
lenses[3].ellipcity = 1.0
lenses[3].x_1 = 181.2
lenses[3].x_2 = 209.3
lenses[3].p0 = 0.1
lenses[3].p3 = 2.5
;''2'' in place of variability means that this will be a
;group of constant mass-to-light ratio lenses
;p3 is the starting ratio of M/L
;all the lenses with ''2'' in the place of variability
;of the scale factor belong to one group
;and are therefore described by on parameter
var3=[0,0,0,2,0,0,0,0,0,0,0]
bound3=[[0.0,0.0],[0.0,0.0],[0.0,0.0],[0.1,3.0],[0.0,0.0],
       [0,0],[0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[4].id = 184
lenses[4].type="isothermal_ellipsoid_circular"
lenses[4].p1 = 510000
lenses[4].b_ref = 2.0
lenses[4].pos_angle = 50.0*!Pi/180.0
lenses[4].ellipcity = 1.0
lenses[4].x_1 = 206.7
lenses[4].x_2 = 202.4
lenses[4].p0 = 0.1
var4=[0,0,0,2,0,0,0,0,0,0,0]
bound4=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
       [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[5].id = 182
lenses[5].type="isothermal_ellipsoid_circular"
lenses[5].p1 = 510000
```

```
lenses[5].b_ref = 0.1
lenses[5].pos_angle = 0.0
lenses[5].ellipcity = 1.0
lenses[5].x_1 = 249.1
lenses[5].x_2 = 73.7
lenses[5].p0 = 0.1
var5=[0,0,0,2,0,0,0,0,0,0,0]
bound5=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[6].id = 210
lenses[6].type="isothermal_ellipsoid_circular"
lenses[6].p1 = 510000
lenses[6].b_ref = 0.1
lenses[6].pos_angle = 0.0
lenses[6].ellipcity =1.0
lenses[6].x_1 = 236.0
lenses[6].x_2 = 154.0
lenses[6].p0 = 0.1
var6=[0,0,0,2,0,0,0,0,0,0,0]
bound6=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[7].id = 7
lenses[7].type="isothermal_ellipsoid_elliptical"
lenses[7].p1 = 510000
lenses[7].b_ref = 0.1
lenses[7].pos_angle = 50.0*!Pi/180.0
lenses[7].ellipcity = 0.3
lenses[7].x_1 = 177.0
lenses[7].x_2 = 103.0
```

```
lenses[7].p0 = 0.1
var7=[0,0,0,1,0,0,0,0,0,0,0]
bound7=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[8].id = 205
lenses[8].type="isothermal_ellipsoid_circular"
lenses[8].p1 = 510000
lenses[8].b_ref = 0.1
lenses[8].pos_angle = 50.0*!Pi/180.0
lenses[8].ellipcity = 1.0
lenses[8].x_1 = 224.9
lenses[8].x_2 = 126.7
lenses[8].p0 = 0.1
var8=[0,0,0,2,0,0,0,0,0,0,0]
bound8=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[9].id = 185
lenses[9].type="isothermal_ellipsoid_circular"
lenses[9].p1 = 510000
lenses[9].b_ref = 0.1
lenses[9].pos_angle = 70.0*!Pi/180.0;
lenses[9].ellipcity = 0.6
lenses[9].x_1 = 237.2
lenses[9].x_2 = 120.6
lenses[9].p0 = 0.1
var9=[0,0,0,2,0,0,0,0,0,0,0]
bound9=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]
```

```
lenses[10].id = 346
lenses[10].type="isothermal_ellipsoid_circular"
lenses[10].p1 = 510000
lenses[10].b_ref = 0.1
lenses[10].pos_angle = 50.0*!Pi/180.0
lenses[10].ellipcity = 1.0
lenses[10].x_1 = 254.1
lenses[10].x_2 = 331.2
lenses[10].p0 = 0.1
var10=[0,0,0,2,0,0,0,0,0,0,0]
bound10=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]

lenses[11].id = 243
lenses[11].type="isothermal_ellipsoid_circular"
lenses[11].p1 = 510000
lenses[11].b_ref = 0.1
lenses[11].pos_angle = 50.0*!Pi/180.0
lenses[11].ellipcity = 1.0
lenses[11].x_1 = 141.6
lenses[11].x_2 = 161.5
lenses[11].p0 = 0.1
var11=[0,0,0,2,0,0,0,0,0,0,0]
bound11=[[0.0,0.0],[0.0,0.0],[0,0],[0,0],[0.0,0.0],[0,0],
        [0,0],[0,0],[0,0],[0,0],[0,0]]
;end of the group definition
```

convert the position of lenses from pixels to arcseconds

```
lenses[0:11].x_1 = lenses[0:11].x_1*xscale
lenses[0:11].x_2 = lenses[0:11].x_2*yscale
```

calculate how much dimmer the members of the substructure group are in respect to the brightest member of the group

```
;the lenses ids
substr_id = [182,    184,    185,    193,    205,    210,
             243,    346]
;and corresponding magnitudes of the galaxies
substr_mag = [15.8388, 13.6959, 14.7071, 14.2414, 14.6008 ,
              15.6224, 15.4918, 14.5516]

ratios = 10^(-substr_mag/2.5)/10^(-substr_mag[1]/2.5)
;save the ''light'' of each member in p4 parameter
for i=0, n_elements(substr_id)-1 do lenses[where(lenses[*].id
    eq substr_id[i])].p4 = ratios[i]
```

Not we define the data. First the position of point-like images

```
;prepare the arrays to hold the positions of arcs
arcsec_x = findgen(37)
arcsec_y = findgen(37)

arcsec_x[0] = 74.0
arcsec_y[0] = 273.0
arcsec_x[1] = 66.0
arcsec_y[1] = 258.0
arcsec_x[2] = 59.0
arcsec_y[2] = 240.0
arcsec_x[3] = 54.0
arcsec_y[3] = 220.0
arcsec_x[4] = 53.0
arcsec_y[4] = 210.0
arcsec_x[5] = 53.0
arcsec_y[5] = 200.0
```

```
arcsec_x[6] = 57.0
arcsec_y[6] = 155.0
arcsec_x[7] = 58.0
arcsec_y[7] = 145.0

arcsec_x[8] = 145.0
arcsec_y[8] = 118.0
arcsec_x[9] = 157.0
arcsec_y[9] = 110.0
arcsec_x[10] = 166.0
arcsec_y[10] = 107.0
arcsec_x[11] = 176.0
arcsec_y[11] = 106.0

arcsec_x[12] = 181.0
arcsec_y[12] = 101.0
arcsec_x[13] = 206.0
arcsec_y[13] = 102.0
arcsec_x[14] = 222.0
arcsec_y[14] = 105.0
arcsec_x[15] = 236.0
arcsec_y[15] = 110.0

arcsec_x[16] = 247.0
arcsec_y[16] = 127.0
arcsec_x[17] = 253.0
arcsec_y[17] = 135.0
arcsec_x[18] = 261.0
arcsec_y[18] = 147.0
arcsec_x[19] = 265.0
arcsec_y[19] = 155.0
```

```
arcsec_x[20] = 266.0
arcsec_y[20] = 162.0
arcsec_x[21] = 267.0
arcsec_y[21] = 165.0

arcsec_x[22] = 176.0
arcsec_y[22] = 296.0
arcsec_x[23] = 187.0
arcsec_y[23] = 297.0
arcsec_x[24] = 197.0
arcsec_y[24] = 298.0
arcsec_x[25] = 210.0
arcsec_y[25] = 295.0

;raidal feature D
arcsec_x[26] = 0.0
arcsec_y[26] = 0.0
arcsec_x[27] = 0.0
arcsec_y[27] = 0.0

;A3
arcsec_x[28] = 242.0
arcsec_y[28] = 194.0
arcsec_x[29] = 246.0
arcsec_y[29] = 194.0
arcsec_x[30] = 252.0
arcsec_y[30] = 192.0

;part of B4
arcsec_x[31] = 0.0
arcsec_y[31] = 0.0
```

```
arcsec_x[32] = 117.0
arcsec_y[32] = 169.0
arcsec_x[33] = 121.0
arcsec_y[33] = 160.0

;radial arc in the center
arcsec_x[34] = 189.5
arcsec_y[34] = 199.5
arcsec_x[35] = 195.5
arcsec_y[35] = 199.0
arcsec_x[36] = 201.0
arcsec_y[36] = 199.5

;convert from pixels to arcseconds
arcsec_x = arcsec_x*xscale
arcsec_y = arcsec_y*yscale
```

Now we load the extended arc information from the png file

```
arcs_data = fix(read_png("rcs0224-with-central-B.png"))
```

Before the minimization in the source plane (minimum source size) can start, we have to organize the point images into *multiplets* – each multiplet represents one arc system. The information about the flux of each point like image and the error of the position measurement are also embedded into the multiplet structure.

```
;prepare data structure for minimalization routine
sigma_1 = replicate(0.05,11)
multiplet_1 = {multiplet_1, n_images: ptr_new(11), redshift:
  ptr_new(4.8786), sigma: ptr_new(sigma_1), image_x:
  ptr_new([arcsec_x[0:7],arcsec_x[28:30]]), image_y:
  ptr_new([arcsec_y[0:7],arcsec_y[28:30]]),flux:ptr_new([0]),
```

```
  sigma_flux:ptr_new([0])}

sigma_2 = replicate(0.05,23)
multiplet_2 = {multiplet_2, n_images: ptr_new(23), redshift:
  ptr_new(2.45084), sigma: ptr_new(sigma_2), image_x:
  ptr_new([arcsec_x[8:11],arcsec_x[16:21],arcsec_x[32:33],
  arcsec_x[12:15],arcsec_x[22:25],arcsec_x[34:36]]), image_y:
  ptr_new([arcsec_y[8:11],arcsec_y[16:21],arcsec_y[32:33],
  arcsec_y[12:15],arcsec_y[22:25],arcsec_y[34:36]]),flux:
  ptr_new([0]),sigma_flux:ptr_new([0])}

data=[multiplet_1,multiplet_2]
```

We are now ready to lunch the minimization routine. We need to pass the lenses and sources arrays, the data and the vectors indicating which variables should be allowed to change, and if so to what limits.

```
P = fit_point_sources(lenses,sources,data, [var0,var1,var2,var3,
    var4,var5,var6,var7,var8, var9, var10, var11, var_s0,
    var_s1], [[bound0],[bound1],[bound2],[bound3],[bound4],
    [bound5],[bound6], [bound7],[bound8], [bound9], [bound10],
    [bound11], [bound_s0], [bound_s1]],/violate)
```

the *violate* keyword means that the lenses and sources will be overwritten with the best fit model. The routine returns a structure that has three members

- *minimum* – the parameters that minimize the chi square
- *minimum_value* – the chi square at the minimum
- *penelty* – the penalty (Eq. (4.4))

Now we can proceed to the model fitting based on the exact shapes of arcs. First we need to create a grid of appropriate resolution. The deflection angle will be calculated at each point of the grid, therefore the more points the more memory and CPU time is

needed. Then we change the variability vector of the source position (since it is a free parameter now and was fixed (calculated) in the previous minimization)

```
;prepare grid
dimx = 401L
dimy = 402L
res = 2
dim_x = dimx*res
dim_y = dimy*res
points_x = reform(findgen(dim_x) # replicate(1.0, dim_y),
   dim_x*dim_y)
points_y = reform(replicate(1.0, dim_x) # findgen(dim_y),
   dim_x*dim_y)
points_x = points_x*xscale/res
points_y = points_y*yscale/res

;the same resolution as background (data) image...
points_x_ei = reform(findgen(dimx) # replicate(1.0, dimy),
   dimx*dimy)*xscale
points_y_ei = reform(replicate(1.0, dimx) # findgen(dimy),
   dimx*dimy)*yscale

;change the sources position variability
var_s0 = [1,1,0]
var_s1 = [1,1,1]
bound_s0 = [[0.0,0.0],[0.0,0.0],[0.0,0.0]]
bound_s1=[[0.0,0.0],[0.0,0.0],[1.0,8.0]]
```

The *fit_extended_images* routine takes the lenses, sources, data, and variability and bounds vectors as parameters. It returns a structure that has four members

- *minimum* – the set of parameters that minimizes Eq. (4.8)
- *minimum_value* – the value of Eq. (4.8) at the minimum

- *penelty* – the value of panelty (Eq. (4.4)) at the minimum
- *dist_hd* – the value of Eq. (4.5) at the minimum

```
P = fit_extended_images(lenses,sources,points_x_ei/xscale,
    points_y_ei/yscale, reform(arcs_data, dimx*dimy),
    points_x_ei, points_y_ei, dimx, dimy, [var0,var1,var2,
    var3,var4,var5,var6,var7,var8, var9, var10, var11,
    var_s0, var_s1], [[bound0],[bound1],[bound2],[bound3],
    [bound4],[bound5],[bound6],[bound7],[bound8], [bound9],
    [bound10], [bound11], [bound_s0], [bound_s1]],/violate)
```

A.7.2 C++ example

In this section a program of similar functionality to the one presented above, written in C++ however, will be presented.

First we include needed header files and define a global graphic module.

```
#include <png.h>
#include <iostream>
#include <cassert>
#include "Nie.h"
#include "Nis.h"
#include "FixedScaleRatioGroup.h"
#include "LensesSystem.h"
#include "LensesContainer.h"
#include "ToolBox.h"
#include "MinimizerPointsSourcePlane.h"
#include "MinimizerExtendedImagePlane.h"
#include "SourceCircular.h"
#include "SourcesContainer.h"
#include "DataContainer.h"
#include "Constants.h"
```

```
#include "./graphic/SDLGraphicsModule.h"
#include "mcmc.h"

SDLGraphicsModule* graphic;
```

In the main function, we initiate the graphics module

```
//init the graphics
graphic = new SDLGraphicsModule();
graphic->SetScreenDim(401, 402);
graphic->Init();
```

and then we define the lenses components, be creating appropriate lenses objects, and suppling a starting parameters, variability for each parameter and the bound for the variable parameters.

```
//****** lens configuration start **********
//free big clump
std::vector<double> Nie1_param;
Nie1_param.push_back(195);
Nie1_param.push_back(204);
Nie1_param.push_back(100);
Nie1_param.push_back(0.8);
Nie1_param.push_back(170*M_PI/180.);
Nie1_param.push_back(1);
std::vector<int> Nie1_var;
Nie1_var.push_back(1);
Nie1_var.push_back(1);
Nie1_var.push_back(1);
Nie1_var.push_back(1);
Nie1_var.push_back(1);
Nie1_var.push_back(1);
std::vector< std::pair<double, double> > Nie1_bound;
Nie1_bound.push_back(std::make_pair(150.0, 250.0));
```

```
Nie1_bound.push_back(std::make_pair(150.0, 250.0));
Nie1_bound.push_back(std::make_pair(0.0, 1000000.0));
Nie1_bound.push_back(std::make_pair(0.3, 0.99));
Nie1_bound.push_back(std::make_pair(0.0, 3.1415926));
Nie1_bound.push_back(std::make_pair(0.0, 1000000.0));



Nie* nie1 = new Nie();
nie1->SetId(1);
nie1->SetParameters(Nie1_param);
nie1->SetParametersVariable(Nie1_var);
nie1->SetParametersBound(Nie1_bound);

//fixed position big clump
std::vector<double> Nie2_param;
Nie2_param.push_back(195);
Nie2_param.push_back(204);
Nie2_param.push_back(50);
Nie2_param.push_back(0.7);
Nie2_param.push_back(1.52668);
Nie2_param.push_back(1);
std::vector<int> Nie2_var;
Nie2_var.push_back(0);
Nie2_var.push_back(0);
Nie2_var.push_back(1);
Nie2_var.push_back(1);
Nie2_var.push_back(1);
Nie2_var.push_back(1);

std::vector< std::pair<double, double> > Nie2_bound;
Nie2_bound.push_back(std::make_pair(150.0, 250.0));
```

```
Nie2_bound.push_back(std::make_pair(150.0, 250.0));
Nie2_bound.push_back(std::make_pair(0.0, 1000000.0));
Nie2_bound.push_back(std::make_pair(0.3, 0.99));
Nie2_bound.push_back(std::make_pair(0.0, 3.1415926));
Nie2_bound.push_back(std::make_pair(0.0, 1000000.0));

Nie* nie2 = new Nie();
nie2->SetId(2);
nie2->SetParameters(Nie2_param);
nie2->SetParametersVariable(Nie2_var);
nie2->SetParametersBound(Nie2_bound);

//the eliptical feature... D
std::vector<double> Nie3_param;
Nie3_param.push_back(177);
Nie3_param.push_back(108);
Nie3_param.push_back(10);
Nie3_param.push_back(0.3);
Nie3_param.push_back(50.0*M_PI/180.);
Nie3_param.push_back(1);
std::vector<int> Nie3_var;
Nie3_var.push_back(0);
Nie3_var.push_back(0);
Nie3_var.push_back(1);
Nie3_var.push_back(0);
Nie3_var.push_back(0);
Nie3_var.push_back(0);

std::vector< std::pair<double, double> > Nie3_bound;
Nie3_bound.push_back(std::make_pair(0.0, 100000.0));
Nie3_bound.push_back(std::make_pair(0.0, 100000.0));
```

```
Nie3_bound.push_back(std::make_pair(0.0, 1000000.0));
Nie3_bound.push_back(std::make_pair(0.0, 0.99));
Nie3_bound.push_back(std::make_pair(0.0, 3.1415926));
Nie3_bound.push_back(std::make_pair(0.0, 1000000.0));

Nie* nie3 = new Nie();
nie3->SetId(3);
nie3->SetParameters(Nie3_param);
nie3->SetParametersVariable(Nie3_var);
nie3->SetParametersBound(Nie3_bound);

//the substructure...
Nis* sub_nis1 = new Nis();
sub_nis1->SetId(11);
std::vector<double> sub_nis1_param;
sub_nis1_param.push_back(181.2);
sub_nis1_param.push_back(209.3);
sub_nis1_param.push_back(1.0);
sub_nis1_param.push_back(1.0);
sub_nis1->SetParameters(sub_nis1_param);

Nis* sub_nis2 = new Nis();
sub_nis2->SetId(12);
std::vector<double> sub_nis2_param;
sub_nis2_param.push_back(206.7);
sub_nis2_param.push_back(202.4);
sub_nis2_param.push_back(1.0);
sub_nis2_param.push_back(1.0);
sub_nis2->SetParameters(sub_nis2_param);

Nis* sub_nis3 = new Nis();
```

```
sub_nis3->SetId(13);
std::vector<double> sub_nis3_param;
sub_nis3_param.push_back(249.1);
sub_nis3_param.push_back(73.7);
sub_nis3_param.push_back(1.0);
sub_nis3_param.push_back(1.0);
sub_nis3->SetParameters(sub_nis3_param);

Nis* sub_nis4 = new Nis();
sub_nis4->SetId(14);
std::vector<double> sub_nis4_param;
sub_nis4_param.push_back(236.0);
sub_nis4_param.push_back(154.0);
sub_nis4_param.push_back(1.0);
sub_nis4_param.push_back(1.0);
sub_nis4->SetParameters(sub_nis4_param);

Nis* sub_nis5 = new Nis();
sub_nis5->SetId(15);
std::vector<double> sub_nis5_param;
sub_nis5_param.push_back(224.9);
sub_nis5_param.push_back(126.7);
sub_nis5_param.push_back(1.0);
sub_nis5_param.push_back(1.0);
sub_nis5->SetParameters(sub_nis5_param);

Nis* sub_nis6 = new Nis();
sub_nis6->SetId(16);
std::vector<double> sub_nis6_param;
sub_nis6_param.push_back(237.2);
sub_nis6_param.push_back(120.6);
```

```
sub_nis6_param.push_back(1.0);
sub_nis6_param.push_back(1.0);
sub_nis6->SetParameters(sub_nis6_param);

Nis* sub_nis7 = new Nis();
sub_nis7->SetId(17);
std::vector<double> sub_nis7_param;
sub_nis7_param.push_back(254.1);
sub_nis7_param.push_back(331.2);
sub_nis7_param.push_back(1.0);
sub_nis7_param.push_back(1.0);
sub_nis7->SetParameters(sub_nis7_param);

Nis* sub_nis8 = new Nis();
sub_nis8->SetId(18);
std::vector<double> sub_nis8_param;
sub_nis8_param.push_back(141.6);
sub_nis8_param.push_back(161.5);
sub_nis8_param.push_back(1.0);
sub_nis8_param.push_back(1.0);
sub_nis8->SetParameters(sub_nis8_param);

FixedScaleRatioGroup* substructure =
            new FixedScaleRatioGroup();
substructure->AddMember(sub_nis1, 0.605063);
substructure->AddMember(sub_nis2, 1.000000);
substructure->AddMember(sub_nis3, 0.138944);
substructure->AddMember(sub_nis4, 0.169590);
substructure->AddMember(sub_nis5, 0.434551);
substructure->AddMember(sub_nis6, 0.394022);
substructure->AddMember(sub_nis7, 0.454695);
```

```
substructure->AddMember(sub_nis8, 0.191267);

std::vector<double> substructure_param;
substructure_param.push_back(2.5);
substructure->SetParameters(substructure_param);
std::vector<int> substructure_var;
substructure_var.push_back(1);
substructure->SetParametersVariable(substructure_var);
std::vector< std::pair<double, double> >
            substructure_bound;
substructure_bound.push_back(std::make_pair(0.0,
            10000.0));
substructure->SetParametersBound(substructure_bound);
```

We add the newly created lenses to a *LensesSystem* and then we add this system to the *LensesContainer*

```
LensesSystem* lSystem1 = new LensesSystem();
lSystem1->AddLens(nie1);
lSystem1->AddLens(nie2);
lSystem1->AddLens(nie3);
lSystem1->AddLens(substructure);
lSystem1->SetRedshift(0.782);
lSystem1->SetId(1);
LensesContainer* lContainer = new LensesContainer();

lContainer->AddLensesSystem(lSystem1);
//********* lens configuration end **************
```

In a similar fashion we create two sources and then add them to the *SourcesContainer*

```
//********* source configuration begin **************
//create a source #1
SourceCircular* src1 = new SourceCircular;
```

```
std::vector<double> src1_param;
src1_param.push_back(190.781);
src1_param.push_back(195.253);
src1_param.push_back(2.35084);
src1_param.push_back(1.0);
src1->SetParameters(src1_param);

std::vector<int> src1_var;
src1_var.push_back(0);
src1_var.push_back(0);
src1_var.push_back(1);
src1_var.push_back(0);
src1->SetParametersVariable(src1_var);
std::vector< std::pair<double, double> > src1_bound;
src1_bound.push_back(std::make_pair(0.0, 400.0));
src1_bound.push_back(std::make_pair(0.0, 400.0));
src1_bound.push_back(std::make_pair(1.0, 4.0));
src1_bound.push_back(std::make_pair(0.0, 100.));
src1->SetParametersBound(src1_bound);


//create a source #2
SourceCircular* src2 = new SourceCircular;
std::vector<double> src2_param;
src2_param.push_back(163.027);
src2_param.push_back(193.914);
src2_param.push_back(4.87860);
src2_param.push_back(1.5);
src2->SetParameters(src2_param);

std::vector<int> src2_var;
```

```
src2_var.push_back(0);
src2_var.push_back(0);
src2_var.push_back(0);
src2_var.push_back(0);
src2->SetParametersVariable(src2_var);
std::vector< std::pair<double, double> > src2_bound;
src2_bound.push_back(std::make_pair(0.0, 400.0));
src2_bound.push_back(std::make_pair(0.0, 400.0));
src2_bound.push_back(std::make_pair(1.0, 10.0));
src2_bound.push_back(std::make_pair(0.0, 100.));
src2->SetParametersBound(src2_bound);



SourcesContainer* sContainer = new SourcesContainer;
sContainer->AddSource(src1);
sContainer->AddSource(src2);
//********** source configuration end **********
```

After having defined the sources and lenses objects we need to introduce the data. First we input the positions of point-like images.

```
//*********** the data ******************
//create the data
DataContainer* dContainer = new DataContainer();
//first arcs system
ArcsSystem* arcsystem1 = new ArcsSystem(src2);
//second arcs system
ArcsSystem* arcsystem2 = new ArcsSystem(src1);

//first arcs system
std::vector<double> arcsystem1_x;
arcsystem1_x.push_back(74.0);
```

```
arcsystem1_x.push_back(66.0);
arcsystem1_x.push_back(59.0);
arcsystem1_x.push_back(54.0);
arcsystem1_x.push_back(53.0);
arcsystem1_x.push_back(53.0);
arcsystem1_x.push_back(57.0);
arcsystem1_x.push_back(58.0);


arcsystem1_x.push_back(242.0);
arcsystem1_x.push_back(246.0);
arcsystem1_x.push_back(252.0);


std::vector<double> arcsystem1_y;
arcsystem1_y.push_back(273.0);
arcsystem1_y.push_back(258.0);
arcsystem1_y.push_back(240.0);
arcsystem1_y.push_back(220.0);
arcsystem1_y.push_back(210.0);
arcsystem1_y.push_back(200.0);
arcsystem1_y.push_back(155.0);
arcsystem1_y.push_back(145.0);

arcsystem1_y.push_back(194.0);
arcsystem1_y.push_back(194.0);
arcsystem1_y.push_back(192.0);


arcsystem1->SetImagesX(arcsystem1_x);
arcsystem1->SetImagesY(arcsystem1_y);
```

```
arcsystem1->SetSigma(0.5);
//second arcs system

std::vector<double> arcsystem2_x;
arcsystem2_x.push_back(145.0);
arcsystem2_x.push_back(157.0);
arcsystem2_x.push_back(166.0);
arcsystem2_x.push_back(176.0);
arcsystem2_x.push_back(181.0);
arcsystem2_x.push_back(206.0);
arcsystem2_x.push_back(222.0);
arcsystem2_x.push_back(236.0);
arcsystem2_x.push_back(247.0);
arcsystem2_x.push_back(253.0);
arcsystem2_x.push_back(261.0);
arcsystem2_x.push_back(265.0);
arcsystem2_x.push_back(266.0);
arcsystem2_x.push_back(261.0);
arcsystem2_x.push_back(176.0);
arcsystem2_x.push_back(187.0);
arcsystem2_x.push_back(197.0);
arcsystem2_x.push_back(210.0);
arcsystem2_x.push_back(117.0);
arcsystem2_x.push_back(121.0);
arcsystem2_x.push_back(189.5);
arcsystem2_x.push_back(195.5);
arcsystem2_x.push_back(201.0);


std::vector<double> arcsystem2_y;
arcsystem2_y.push_back(118.0);
```

```
arcsystem2_y.push_back(110.0);
arcsystem2_y.push_back(107.0);
arcsystem2_y.push_back(106.0);
arcsystem2_y.push_back(101.0);
arcsystem2_y.push_back(102.0);
arcsystem2_y.push_back(105.0);
arcsystem2_y.push_back(110.0);
arcsystem2_y.push_back(127.0);
arcsystem2_y.push_back(135.0);
arcsystem2_y.push_back(147.0);
arcsystem2_y.push_back(155.0);
arcsystem2_y.push_back(162.0);
arcsystem2_y.push_back(165.0);
arcsystem2_y.push_back(296.0);
arcsystem2_y.push_back(297.0);
arcsystem2_y.push_back(298.0);
arcsystem2_y.push_back(295.0);
arcsystem2_y.push_back(169.0);
arcsystem2_y.push_back(160.0);
arcsystem2_y.push_back(199.5);
arcsystem2_y.push_back(199.0);
arcsystem2_y.push_back(199.5);

arcsystem2->SetImagesX(arcsystem2_x);
arcsystem2->SetImagesY(arcsystem2_y);
arcsystem2->SetSigma(0.5);

//add arcs system for point sources minimi
dContainer->AddArcsSystem(arcsystem1);
dContainer->AddArcsSystem(arcsystem2);
//************ data end ********************
```

and add them to the *DataContainer*. Next we read in the shapes of the arcs from the png file.

```
//add arcs systems for extended image minimi;
dContainer->AddArcsSystemExtended(new ArcsSystem(src2));
dContainer->AddArcsSystemExtended(new ArcsSystem(src1));

//read arcs data
dContainer->LoadExtendedArcs("Data/rcs0224-with-central-B.png");
```

Now, we can start the minimization process. First we use only the point like images to find a first approximation to our lensing model.

```
//*************** the minimization ****************
//POINT LIKE IMAGES
MinimizerPointsSourcePlane* minimizer =
     new MinimizerPointsSourcePlane;
minimizer->SetLensesContainer(lContainer);
minimizer->SetSourcesContainer(sContainer);
minimizer->SetDataContainer(dContainer);
//initialize parameters to minimization
minimizer->InitializeVariableParameters();
minimizer->SetMinimizationMethodName("Powell");

minimizer->Minimize();
```

Then we use the previously found model as a starting point for the routine that searches for the bast fit model, based on the full shapes of arcs.

```
//EXTENDED IMAGES MINIMIZATION
//now the source positions are variable
src1->SetParameterVariableAt(0,1);
src1->SetParameterVariableAt(1,1);
src2->SetParameterVariableAt(0,1);
src2->SetParameterVariableAt(1,1);
```

```
//exteded images minimization test
MinimizerExtendedImagePlane* minimizer_ex =
    new MinimizerExtendedImagePlane;
minimizer_ex->SetLensesContainer(lContainer);
minimizer_ex->SetSourcesContainer(sContainer);
minimizer_ex->SetDataContainer(dContainer);
//initialize parameters to minimization
minimizer_ex->InitializeVariableParameters();
minimizer_ex->SetMinimizationMethodName("Powell");
minimizer_ex->Minimize();
```

Bibliography

Alcock, C., Akerloff, C. W., Allsman, R. A., Axelrod, T. S., Bennett, D. P., Chan, S., Cook, C. H., Freeman, K. C., Griest, K., Marshall, S. L., Park, H. S., Perlmutter, S., Peterson, B. A., Pratt, M. R., Quinn, P. J., Rodgers, A. W., Stubbs, C. W. and Sutherland, W.: 1993, Possible Gravitational Microlensing of a Star in the Large Magellanic Cloud, Nature **365**, 621–+.

Arnaud, K. A.: 1996, XSPEC: The First Ten Years, *in* G. H. Jacoby and J. Barnes (eds), *ASP Conf. Ser. 101: Astronomical Data Analysis Software and Systems V*, pp. 17–+.

Balestra, I., Tozzi, P., Ettori, S., Rosati, P., Borgani, S., Mainieri, V., Norman, C. and Viola, M.: 2007, Tracing the evolution in the iron content of the intra-cluster medium, A&A **462**, 429–442.

Bartelmann, M.: 1995, Arc statistics with realistic cluster potentials. III. A systematic effect on cluster mass estimates., A&A **299**, 11–+.

Bartelmann, M.: 1996, Arcs from a universal dark-matter halo profile., A&A **313**, 697–702.

Bartelmann, M., Narayan, R., Seitz, S. and Schneider, P.: 1996, Maximum-likelihood Cluster Reconstruction, ApJ **464**, L115+.

Bertin, E. and Arnouts, S.: 1996, SExtractor: Software for source extraction., A&AS **117**, 393–404.

Bertin, G. and Lombardi, M.: 2006, Looking at the Fundamental Plane through a Gravitational Lens, ApJ **648**, L17–L20.

Best, P. N., van Dokkum, P. G., Franx, M. and Röttgering, H. J. A.: 2002, μJy radio sources in the cluster MS1054-03, MNRAS **330**, 17–34.

Bradač, M., Clowe, D., Gonzalez, A. H., Marshall, P., Forman, W., Jones, C., Markevitch, M., Randall, S., Schrabback, T. and Zaritsky, D.: 2006, Strong and Weak Lensing United. III. Measuring the Mass Distribution of the Merging Galaxy Cluster 1ES 0657-558, ApJ **652**, 937–947.

Broadhurst, T., Benítez, N., Coe, D., Sharon, K., Zekser, K., White, R., Ford, H., Bouwens, R., Blakeslee, J., Clampin, M., Cross, N., Franx, M., Frye, B., Hartig, G., Illingworth, G., Infante, L., Menanteau, F., Meurer, G., Postman, M., Ardila, D. R., Bartko, F., Brown, R. A., Burrows, C. J., Cheng, E. S., Feldman, P. D., Golimowski, D. A., Goto, T., Gronwall, C., Herranz, D., Holden, B., Homeier, N., Krist, J. E., Lesser, M. P., Martel, A. R., Miley, G. K., Rosati, P., Sirianni, M., Sparks, W. B., Steindling, S., Tran, H. D., Tsvetanov, Z. I. and Zheng, W.: 2005, Strong-Lensing Analysis of A1689 from Deep Advanced Camera Images, ApJ **621**, 53–88.

Broadhurst, T. J., Taylor, A. N. and Peacock, J. A.: 1995, Mapping cluster mass distributions via gravitational lensing of background galaxies, ApJ **438**, 49–61.

Broadhurst, T., Takada, M., Umetsu, K., Kong, X., Arimoto, N., Chiba, M. and Futamase, T.: 2005, The Surprisingly Steep Mass Profile of A1689, from a Lensing Analysis of Subaru Images, ApJ **619**, L143–L146.

Cacciato, M., Bartelmann, M., Meneghetti, M. and Moscardini, L.: 2006, Combining weak and strong lensing in cluster potential reconstruction, A&A **458**, 349–356.

Chwolson, O.: 1924, Über eine mögliche Form fiktiver Doppelsterne, *Astronomische Nachrichten* **221**, 329–+.

Ciufolini, I., Lucchesi, D., Vespe, F. and Chieppa, F.: 2007, Detection of Lense-Thirring Effect Due to Earth's Spin, *gr-qc* .

Clowe, D., Bradač, M., Gonzalez, A. H., Markevitch, M., Randall, S. W., Jones, C. and Zaritsky, D.: 2006, A Direct Empirical Proof of the Existence of Dark Matter, ApJ **648**, L109–L113.

Dickey, J. M. and Lockman, F. J.: 1990, H I in the Galaxy, ARA&A **28**, 215–261.

Dubuisson, M. P. and Jain, A. K.: 1994, A modified Hausdorff distance for object matching., *ICPR* pp. A:566–568.

Eddington, A. S.: 1920, *Space, time and gravitation. an outline of the general relativity theory*, Cambridge Science Classics, Cambridge: University Press, 1920.

Einstein, A.: 1911, Über den Einfluß der Schwerkraft auf die Ausbreitung des Lichtes, *Annalen der Physik* **340**, 898–908.

Einstein, A.: 1915, Erklarung der Perihelionbewegung der Merkur aus der allgemeinen Relativitatstheorie, *Sitzungsber. preuss.Akad. Wiss., vol. 47, No.2, pp. 831-839, 1915* **47**, 831–839.

Einstein, A.: 1936, Lens-Like Action of a Star by the Deviation of Light in the Gravitational Field, *Science* **84**, 506–507.

Feller, W.: 1968, *An Introduction to Probability Theory and Its Applications*, New York: John Wiley, 1968.

Fischer P., Bernstein G., R. G. T. J. A.: 1997, The mass distribution of the cluster 0957+561 from gravitational lensing, *Astron. J.* **113**.

Gavazzi, R., Fort, B., Mellier, Y., Pelló, R. and Dantel-Fort, M.: 2003, A radial mass profile analysis of the lensing cluster MS 2137.3-2353, A&A **403**, 11–27.

Gladders, M. D., Yee, H. K. C. and Ellingson, E.: 2002, Discovery of a z =0.77 Galaxy Cluster with Multiple, Bright, Strong-Lensing Arcs, AJ **123**, 1–9.

Gradshteyn, I. S. and Ryzhik, I. M.: 1994, *Table of Itegrals, Series, and Producs, Fifth Edition, ed. A. Jeffrey*, San Diego: Academic Press, 1994.

Halkola, A., Seitz, S. and Pannella, M.: 2006, Parametric strong gravitational lensing analysis of Abell 1689, MNRAS **372**, 1425–1462.

Hicks, A. K., Ellingson, E., Bautz, M., Yee, H. K. C., Gladders, M. and Garmire, G.: 2005, Chandra X-ray observations of newly discovered, z $\sim$ 1 clusters from the red-sequence cluster survey, *Advances in Space Research* **36**, 706–709.

Hoekstra, H., Mellier, Y., van Waerbeke, L., Semboloni, E., Fu, L., Hudson, M. J., Parker, L. C., Tereno, I. and Benabed, K.: 2006, First Cosmic Shear Results from the Canada-France-Hawaii Telescope Wide Synoptic Legacy Survey, ApJ **647**, 116–127.

Holland, J. H.: 1975, *Adaptation in natural and artificial systems. an introductory analysis with applications to biology, control and artificial intelligence*, Ann Arbor: University of Michigan Press, 1975.

Jee, M. J., Ford, H. C., Illingworth, G. D., White, R. L., Broadhurst, T. J., Coe, D. A., Meurer, G. R., van der Wel, A., Benítez, N., Blakeslee, J. P., Bouwens, R. J., Bradley, L. D., Demarco, R., Homeier, N. L., Martel, A. R. and Mei, S.: 2007, Discovery of a Ringlike Dark Matter Structure in the Core of the Galaxy Cluster Cl 0024+17, ApJ **661**, 728–749.

Kaastra, J. S.: 1992, Soft and Hard X-Ray Variability in 3C382, *in* Y. Tanaka and K. Koyama (eds), *Frontiers Science Series*, pp. 533–+.

Kantowski, R.: 1969, Corrections in the Luminosity-Redshift Relations of the Homogeneous Fried-Mann Models, ApJ **155**, 89–+.

Keeton, C. R.: 2001a, A Catalog of Mass Models for Gravitational Lensing, *ArXiv Astrophysics e-prints* .

Keeton, C. R.: 2001b, Computational Methods for Gravitational Lensing, *ArXiv Astrophysics e-prints* .

Keeton, C. R. and Kochanek, C. S.: 1998, Gravitational Lensing by Spiral Galaxies, ApJ **495**, 157–+.

Kemeny, J. G. and Snell, J. L.: 1960, *Finite Markov Chains*, New York: Springer-Verlag, 1960.

Kinney, A. L., Calzetti, D., Bohlin, R. C., McQuade, K., Storchi-Bergmann, T. and Schmitt, H. R.: 1996, Template Ultraviolet to Near-Infrared Spectra of Star-forming Galaxies and Their Application to K-Corrections, ApJ **467**, 38–+.

Koopmans, L.: 2004, Dark-Matter and Baryons in Early-type Lens Galaxies, *in* R. Dettmar, U. Klein and P. Salucci (eds), *Baryons in Dark Matter Halos*.

Koopmans, L. V. E.: 2005, Gravitational imaging of cold dark matter substructures, MNRAS **363**, 1136–1144.

Koopmans, L. V. E. and Treu, T.: 2002, The Stellar Velocity Dispersion of the Lens Galaxy in MG 2016+112 at z=1.004, ApJ **568**, L5–L8.

Lebach, D., Corey, B., Shapiro, I., Ratner, M., Webber, J., Rogers, A., Davis, J. and Herring, T.: 1995, Measurement of the solar gravitational deflection of radio waves using very-long-baseline interferometry, *Phys. Rev. Lett.* **75**, 1439–1442.

Liebes, S.: 1964, Gravitational Lenses, *Physical Review* **133**, 835–844.

Liedahl, D. A., Osterheld, A. L. and Goldstein, W. H.: 1995, New calculations of Fe L-shell X-ray spectra in high-temperature plasmas, ApJ **438**, L115–L118.

Liesenborgs, J., De Rijcke, S. and Dejonghe, H.: 2006, A genetic algorithm for the non-parametric inversion of strong lensing systems, MNRAS **367**, 1209–1216.

Lombardi, M. and Bertin, G.: 1999, Weak lensing and cosmology, A&A **342**, 337–352.

Lynds, R. and Petrosian, V.: 1986, Giant Luminous Arcs in Galaxy Clusters, *Bulletin of the American Astronomical Society*, Vol. 18 of *Bulletin of the American Astronomical Society*, pp. 1014–+.

Meneghetti, M., Bartelmann, M. and Moscardini, L.: 2003, Cluster cross-sections for strong lensing: analytic and numerical lens models, MNRAS **340**, 105–114.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E.: 1953, Equations of State Calculations by Fast Computing Machines, *Journal of Chemical Physics* **21(6)**, 1087–1092.

Meylan, G., Jetzer, P., North, P., Schneider, P., Kochanek, C. S. and Wambsganss, J. (eds): 2006, *Gravitational Lensing: Strong, Weak and Micro*.

Moore, B., Governato, F., Quinn, T., Stadel, J. and Lake, G.: 1998, Resolving the Structure of Cold Dark Matter Halos, ApJ **499**, L5+.

Navarro, J. F., Frenk, C. S. and White, S. D. M.: 1996, The Structure of Cold Dark Matter Halos, ApJ **462**, 563–+.

Neal, M. R.: 1993, Probabilistic Inference Using Markov Chain Monte Carlo Methods, *Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto* .

Newton, I.: 1704, *Opticks: Or, A treatise of the Reflections, Refractions, Inflexions and Colours of Light. Also Two treatises of the Species and Magnitude of Curvilinear Figures*, London: 1704.

Peacock, J. A.: 1999, *Cosmological Physics*, Cambridge Science Classics, Cambridge: University Press, 1999.

Peacock, J. and Schneider, P.: 2006, The ESO-ESA Working Group on Fundamental Cosmology, *The Messenger* **125**, 48–+.

Powell, M. J. D.: 1964, An efficient method for finding the minimum of a function of several variables without calculating derivatives, *Computer Journal* **7**, 152–162.

Read, J.: 2007, PixeLens: Non-parametric lensing the Bayesian way, *ANGLES school on gravitational lens modelling, lecture notes* .

Refsdal, S.: 1964a, On the possibility of determining Hubble's parameter and the masses of galaxies from the gravitational lens effect, MNRAS **128**, 307–+.

Refsdal, S.: 1964b, The gravitational lens effect, MNRAS **128**, 295–+.

Rosati, P., Borgani, S. and Norman, C.: 2002, The Evolution of X-ray Clusters of Galaxies, ARA&A **40**, 539–577.

Sachs, R.: 1961, Gravitational Waves in General Relativity. VI. The Outgoing Radiation Condition, *Royal Society of London Proceedings Series A* **264**, 309–338.

Saha, P., Coles, J., Macciò, A. V. and Williams, L. L. R.: 2006, The Hubble Time Inferred from 10 Time Delay Lenses, ApJ **650**, L17–L20.

Saha, P. and Williams, L. L. R.: 1997, Non-parametric reconstruction of the galaxy lens in PG 1115+080, MNRAS **292**, 148–+.

Sand, D. J., Treu, T. and Ellis, R. S.: 2002, The Matter Distribution in Galaxy Clusters, *Bulletin of the American Astronomical Society*, pp. 1208–+.

Sand, D. J., Treu, T., Smith, G. P. and Ellis, R. S.: 2004, The Dark Matter Distribution in the Central Regions of Galaxy Clusters: Implications for Cold Dark Matter, ApJ **604**, 88–107.

Sarazin, C.: 1988, *X-Ray Emission from Clusters of Galaxies*, Cambridge: Cambridge Univ. Press.

Schmidt, M.: 1963, 3C 273 : A Star-Like Object with Large Red-Shift, *Nature* **197**, 1040–+.

Schneider, P. and Ehlers J. and Falco E.: 1992, *Gravitational Lenses*.

Schramm, T.: 1990, Realistic elliptical potential wells for gravitational lens models, A&A **231**, 19–24.

Shapiro, I. I.: 1964, Fourth Test of General Relativity, *Physical Review Letters* **13**, 789–791.

Sharon, K., Ofek, E. O., Smith, G. P., Broadhurst, T., Maoz, D., Kochanek, C. S., Oguri, M., Suto, Y., Inada, N. and Falco, E. E.: 2005, Discovery of Multiply Imaged Galaxies behind the Cluster and Lensed Quasar SDSS J1004+4112, ApJ **629**, L73–L76.

Soldner, J.: 1804, Über die Ablenkung eines Lichtstrals von seiner geradlinigen Bewegung, durch die Attraktion eines Weltkrpers, an welchem er nahe vorbeigeht, *Berliner Astron. Jahrb.* p. p. 161.

Spergel, D. N., Bean, R., Dore, O., Nolta, M. R., Bennett, C. L., Dunkley, J., Hinshaw, G., Jarosik, N., Komatsu, E., Page, L., Peiris, H. V., Verde, L., Halpern, M., Hill, R. S., Kogut, A., Limon, M., Meyer, S. S., Odegard, N., Tucker, G. S., Weiland, J. L., Wollack, E. and Wright, E. L.: 2007, Wilkinson microwave anisotropy probe (wmap) three year results: Implications for cosmology, *APJS* **170**, 377.
URL: *http://www.citebase.org/abstract?id=oai:arXiv.org:astro-ph/0603449*

Swinbank, A. M., Bower, R. G., Smith, G. P., Wilman, R. J., Smail, I., Ellis, R. S., Morris, S. L. and Kneib, J.-P.: 2007, Resolved spectroscopy of a gravitationally lensed L* Lyman-break galaxy at z ˜ 5, MNRAS **376**, 479–491.

Trotter, C. S., Winn, J. N. and Hewitt, J. N.: 2000, A Multipole-Taylor Expansion for the Potential of the Gravitational Lens MG J0414+0534, ApJ **535**, 671–691.

Tyson, J. A., Wenk, R. A. and Valdes, F.: 1990, Detection of systematic gravitational lens galaxy image alignments - Mapping dark matter in galaxy clusters, ApJ **349**, L1–L4.

Walsh, D., Carswell, R. F. and Weymann, R. J.: 1979, 0957 + 561 A, B - Twin quasistellar objects or gravitational lens, Nature **279**, 381–384.

Webb, T. M. A., Yee, H. K. C., Ivison, R. J., Hoekstra, H., Gladders, M. D., Barrientos, L. F. and Hsieh, B. C.: 2005, Submillimeter Imaging of RCS J022434-0002.5: Intense Activity in a High-Redshift Cluster?, ApJ **631**, 187–196.

Williams, L. L. R. and Saha, P.: 2004, Models of the Giant Quadruple Quasar SDSS J1004+4112, AJ **128**, 2631–2641.

Wilms, J., Allen, A. and McCray, R.: 2000, On the Absorption of X-Rays in the Interstellar Medium, ApJ **542**, 914–924.

Wright, A. H.: 1991, Genetic algorithms for real parameter optimization, *in* G. J. Rawlins (ed.), *Foundations of genetic algorithms*, Morgan Kaufmann, San Mateo, CA, pp. 205–218.
URL: *citeseer.ist.psu.edu/wright91genetic.html*

Zwicky, F.: 1937a, Nebulae as gravitational lenses, *Phys. Rev. Lett.* **51**, 290.

Zwicky, F.: 1937b, On the probability of detecting nebulae which act as gravitational lenses, *Phys. Rev. Lett.* **51**, 679.